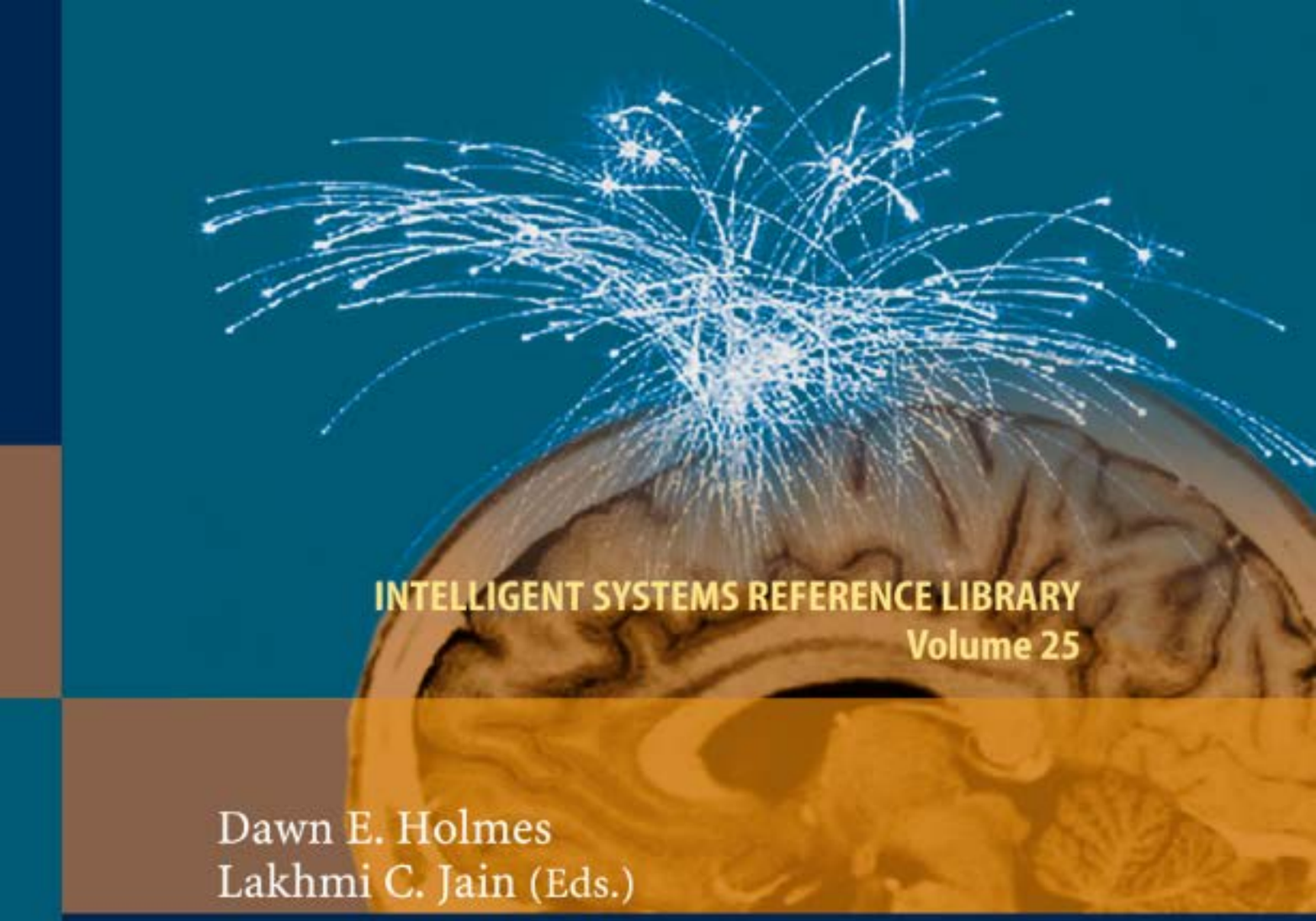


INTELLIGENT SYSTEMS REFERENCE LIBRARY
Volume 25

Dawn E. Holmes
Lakhmi C. Jain (Eds.)

# Data Mining: Foundations and Intelligent Paradigms

## Volume 3: Medical, Health, Social, Biological and other Applications

Springer

Intelligent Systems Reference Library, Volume 25



---

Further volumes of this series can be found on our homepage:
springer.com

Vol. 1. Christine L. Mumford and Lakhmi C. Jain (Eds.)
*Computational Intelligence: Collaboration, Fusion and Emergence,* 2009
ISBN 978-3-642-01798-8

Vol. 2. Yuehui Chen and Ajith Abraham
*Tree-Structure Based Hybrid Computational Intelligence,* 2009
ISBN 978-3-642-04738-1

Vol. 3. Anthony Finn and Steve Scheding
*Developments and Challenges for Autonomous Unmanned Vehicles,* 2010
ISBN 978-3-642-10703-0

Vol. 4. Lakhmi C. Jain and Chee Peng Lim (Eds.)
*Handbook on Decision Making: Techniques and Applications,* 2010
ISBN 978-3-642-13638-2

Vol. 5. George A. Anastassiou
*Intelligent Mathematics: Computational Analysis,* 2010
ISBN 978-3-642-17097-3

Vol. 6. Ludmila Dymowa
*Soft Computing in Economics and Finance,* 2011
ISBN 978-3-642-17718-7

Vol. 7. Gerasimos G. Rigatos
*Modelling and Control for Intelligent Industrial Systems,* 2011
ISBN 978-3-642-17874-0

Vol. 8. Edward H.Y. Lim, James N.K. Liu, and Raymond S.T. Lee
*Knowledge Seeker – Ontology Modelling for Information Search and Management,* 2011
ISBN 978-3-642-17915-0

Vol. 9. Menahem Friedman and Abraham Kandel
*Calculus Light,* 2011
ISBN 978-3-642-17847-4

Vol. 10. Andreas Tolk and Lakhmi C. Jain
*Intelligence-Based Systems Engineering,* 2011
ISBN 978-3-642-17930-3

Vol. 11. Samuli Niiranen and Andre Ribeiro (Eds.)
*Information Processing and Biological Systems,* 2011
ISBN 978-3-642-19620-1

Vol. 12. Florin Gorunescu
*Data Mining,* 2011
ISBN 978-3-642-19720-8

Vol. 13. Witold Pedrycz and Shyi-Ming Chen (Eds.)
*Granular Computing and Intelligent Systems,* 2011
ISBN 978-3-642-19819-9

Vol. 14. George A. Anastassiou and Oktay Duman
*Towards Intelligent Modeling: Statistical Approximation Theory,* 2011
ISBN 978-3-642-19825-0

Vol. 15. Antonino Freno and Edmondo Trentin
*Hybrid Random Fields,* 2011
ISBN 978-3-642-20307-7

Vol. 16. Alexiei Dingli
*Knowledge Annotation: Making Implicit Knowledge Explicit,* 2011
ISBN 978-3-642-20322-0

Vol. 17. Crina Grosan and Ajith Abraham
*Intelligent Systems,* 2011
ISBN 978-3-642-21003-7

Vol. 18. Achim Zielesny
*From Curve Fitting to Machine Learning,* 2011
ISBN 978-3-642-21279-6

Vol. 19. George A. Anastassiou
*Intelligent Systems: Approximation by Artificial Neural Networks,* 2011
ISBN 978-3-642-21430-1

Vol. 20. Lech Polkowski
*Approximate Reasoning by Parts,* 2011
ISBN 978-3-642-22278-8

Vol. 21. Igor Chikalov
*Average Time Complexity of Decision Trees,* 2011
ISBN 978-3-642-22660-1

Vol. 22. Przemysław Różewski, Emma Kusztina, Ryszard Tadeusiewicz, and Oleg Zaikin
*Intelligent Open Learning Systems,* 2011
ISBN 978-3-642-22666-3

Vol. 23. Dawn E. Holmes and Lakhmi C. Jain (Eds.)
*Data Mining: Foundations and Intelligent Paradigms,* 2012
ISBN 978-3-642-23165-0

Vol. 24. Dawn E. Holmes and Lakhmi C. Jain (Eds.)
*Data Mining: Foundations and Intelligent Paradigms,* 2012
ISBN 978-3-642-23240-4

Vol. 25. Dawn E. Holmes and Lakhmi C. Jain (Eds.)
*Data Mining: Foundations and Intelligent Paradigms,* 2012
ISBN 978-3-642-23150-6

Dawn E. Holmes and Lakhmi C. Jain (Eds.)

# Data Mining: Foundations and Intelligent Paradigms

## Volume 3: Medical, Health, Social, Biological and other Applications

Springer

Prof. Dawn E. Holmes
Department of Statistics and Applied Probability
University of California,
Santa Barbara,
CA 93106
USA
E-mail: holmes@pstat.ucsb.edu

Prof. Lakhmi C. Jain
Professor of Knowledge-Based Engineering
University of South Australia
Adelaide
Mawson Lakes, SA 5095
Australia
E-mail: Lakhmi.jain@unisa.edu.au

ISBN 978-3-642-23150-6 e-ISBN 978-3-642-23151-3

DOI 10.1007/978-3-642-23151-3

Intelligent Systems Reference Library ISSN 1868-4394

Library of Congress Control Number: 2011936705



*Typeset* & *Cover Design:* Scientific Publishing Services Pvt. Ltd., Chennai, India.

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

# Preface

There are many invaluable books available on data mining theory and applications. However, in compiling a volume titled "DATA MINING: Foundations and Intelligent Paradigms: Volume 3: Medical, Health, Social, Biological and other Applications" we wish to introduce some of the latest developments to a broad audience of both specialists and non-specialists in this field.

The term 'data mining' was introduced in the 1990's to describe an emerging field based on classical statistics, artificial intelligence and machine learning. By combining techniques from these areas, and developing new ones researchers are able to innovatively analyze large datasets productively. Patterns found in these datasets are subsequently analyzed with a view to acquiring new knowledge. These techniques have been applied in a broad range of medical, health, social and biological areas.

In compiling this volume we have sought to present innovative research from prestigious contributors in the field of data mining. Each chapter is self-contained and is described briefly in Chapter 1.

This book will prove valuable to theoreticians as well as application scientists/engineers in the area of Data Mining. Postgraduate students will also find this a useful sourcebook since it shows the direction of current research.

We have been fortunate in attracting top class researchers as contributors and wish to offer our thanks for their support in this project. We also acknowledge the expertise and time of the reviewers. Finally, we also wish to thank Springer for their support.

Dr. Dawn E. Holmes
University of California
Santa Barbara, USA

Dr. Lakhmi C. Jain
University of South Australia
Adelaide, Australia

# Contents

# Editors

Dr. Dawn E. Holmes serves as Senior Lecturer in the Department of Statistics and Applied Probability and Senior Associate Dean in the Division of Undergraduate Education at UCSB. Her main research area, Bayesian Networks with Maximum Entropy, has resulted in numerous journal articles and conference presentations. Her other research interests include Machine Learning, Data Mining, Foundations of Bayesianism and Intuitionistic Mathematics. Dr. Holmes has co-edited, with Professor Lakhmi C. Jain, volumes 'Innovations in Bayesian Networks' and 'Innovations in Machine Learning'. Dr. Holmes teaches a broad range of courses, including SAS programming, Bayesian Networks and Data Mining. She was awarded the Distinguished Teaching Award by Academic Senate, UCSB in 2008.

As well as being Associate Editor of the International Journal of Knowledge-Based and Intelligent Information Systems, Dr. Holmes reviews extensively and is on the editorial board of several journals, including the Journal of Neurocomputing. She serves as Program Scientific Committee Member for numerous conferences; including the International Conference on Artificial Intelligence and the International Conference on Machine Learning. In 2009 Dr. Holmes accepted an invitation to join Center for Research in Financial Mathematics and Statistics (CRFMS), UCSB. She was made a Senior Member of the IEEE in 2011.

Professor Lakhmi C. Jain is a Director/Founder of the Knowledge-Based Intelligent Engineering Systems (KES) Centre, located in the University of South Australia. He is a fellow of the Institution of Engineers Australia.

His interests focus on the artificial intelligence paradigms and their applications in complex systems, art-science fusion, e-education, e-healthcare, unmanned air vehicles and intelligent agents.

# Chapter 1
# Advances in Intelligent Data Mining

Dawn E. Holmes, Jeffrey W. Tweedale, and Lakhmi C. Jain

[1] Department of Statistics and Applied Probability,
University of California Santa Barbara,
Santa Barbara, CA 93106-3110, USA
holmes@pstat.ucsb.edu
[2] Defence Science and Technology Organisation,
PO. Box. 1500, Edinburgh
South Australia SA 5111, Australia
jeffrey.tweedale@dsto.defence.gov.au
[3] School of Electrical and Information Engineering,
University of South Australia, Adelaide,
Mawson Lakes Campus, South Australia SA 5095,
Australia
lakhmi.jain@unisa.edu.au

## 1 Introduction

The human body is composed of eleven sub-systems. These include the: respiratory, digestive, muscular, immune, circulatory, digestive, skeletal, endocrine, urinary, integumentary and reproductive systems [1]. Science shows how complex systems interoperate and have even mapped the human genome. This knowledge resulted through the exploitation of significant volumes of empirical data. The size of medical databases are many orders of magnitude those of text and transactional repositories. Acquisition, storage and exploitation of this data requires a disparate approach due to the modes and methods of representing what is being captured. This is significantly important in the medical field. As we transition from paper or film capture across to digital repositories, the challenges grow exponentially. The technological challenges compel the industry to undergo a paradigm shift that has resulted from the volume and bandwidth demanded of radiological imaging. Again, society is demanding instant access and analysis of diagnostic equipment to enable timely management of medical conditions or treatment. Such treatment also requires access to patient records, regardless of their source or location. This book examines recent developments in Medical, Health, Social and Biological applications.

More healthcare related data is being collected than ever before; for example, medication records for individual patients, epidemic statistics for public health specialists, data on new surgical techniques and drug interactions. Data mining techniques, such as clustering, k-means and neural networks, now hold a key role in knowledge discovery from medical data. Using prediction and classification techniques has enabled researchers to investigate the success of surgical procedures and the efficacy of medication. Much useful work has already been accomplished in the field of medical data

D.E. Holmes, L.C. Jain (Eds.): Data Mining: Found. & Intell. Paradigms, ISRL 25, pp. 1–7.
springerlink.com 

mining; for example, heart attack prediction using neural network technology and classification of breast tumors. Mining EEG and EKG images is also a growth area. As well as technical issues, ethical, social and legal questions, such as data ownership and confidentiality arise from this area of research, making it truly interdisciplinary. The development of clinical databases has led to the burgeoning field of healthcare data mining, which we now explore.

## 2 Medical Influences

Science and technology has stimulated medicine to evolve diagnostic and treatment regimes. Information technology has enabled humanity to capture and store virtually every aspect of its existing and influence on life. Photography demonstrated the effectiveness of capture and storage in diagnosing or recording changes in our bodies. This method of diagnosis provides snapshots and enables comparative analysis. Medical practitioners have embraced this concept by embracing equipment with digital capture. The transition from paper to screen has forced the data mining community to adapt text based paradigms to data mining and data fusion technology in order to extract meaningful knowledge form diagnostic images and complex data structures [2].

## 3 Health Influences

eHealth has emerged as a significant benefit to society, allowing doctors to monitor, diagnose and treat medical condition more effectively than ever before. There are obvious benefits to hospitals and cost significant savings that flow on to patients. Medical information has unique characteristics, with significant duplicity and redundancy [3,4]. It is plagued by multi-attribution, incomplete information and time sensitive data which influences diagnostic techniques. Significant research continues to evolve methodology to expose viable and verifiable knowledge about specific conditions. Obviously the intent is to isolate each condition for individuals patients and concentrate on creating a healthy and productive society.

## 4 Social Influences

Data mining has influenced preventative health regimes by enabling improved coordination of immunisation, pandemic control and public or social policy. Technology enables improved access to societal data with respect to public health trends, attitudes and perception. The community can monitor and engage in effective public health campaigns or treatment. The recent swine flu pandemic is a prime example.

### 4.1 Information Discovery

In the field of Information Discovery, a family of automated techniques are used to sift through volume of data available in order to find knowledge. These *nuggets* can be deployed directly by applications to help focus a more appropriate outcome. Examples

include: recommendations provided online by bookstores or even the ranking results provided from a search engines [5]. Other techniques are used to support management. These techniques focus on the use of analysis. Here industry could identify high-value customers or even detect credit-card fraud [6]. Governments also use data mining to identify suspicious individuals[1]. The reliance on discovery is increasing and ongoing research continues. We provide an example of current research in the chapter on 'Data Mining for Information Literacy'.

### 4.2 On-Line Communities

The use of *Social Networking* has become unbiquitous around the world. The ability to access friends and colleagues on-line has become mandatory for many people to be accepted into any perr group or social relationship. Facebook, Twitter, Forums and the Web provide connectivity to virtual communities regardless of the geographical dispersion of the participants. We are no longer tethered to the desktop and the use of mobile devices is enabling the real-time engagement in many activities previously restricted to the situation, circumstance or even physical presence. Data mining is increasingly used by the providers of these tools, in particular, the analysis of egocentric graphs of Internet forum users, may help in understanding social role attribution between users. An example is the Web 2.0[2]. This topic is discussed further in the chapter on 'Analysis and mining of online communities of Internet forum users'. Examples discussed include both *push* and *pull* architectures, bulletin boards, social data, various **API!** (**API!**) and even *AJAX*.

## 5 Biological Influences

The have been a number of significant projects where data mining has influenced the biological field of research. The *Human Genome Project* was jointly founded in 1990 by the U.S. Department of Energy and the U.S. National Institutes of Health. The Genome project was completed in 2003 [7][3]. A whole suite of genome projects and tools continue to evolve. Specialist Data Mining techniques are required to classify, cluster, view and analyse biological data [8,9]. The complexity of relationships between genomes and phenotypes make this data highly-dimensional which is typically resolved based on a series of assumption which are repeatedly tested using inference. We provide several chapters on *Biological Networks* and *Genome Expressions*.

### 5.1 Biological Networks

Metabolic networks are composed of different entities which are enzymes, compounds and reactions. The degree of similarity between pairs of entities of two networks is, usually, a good indicator of the similarity between the networks [10,**?**]. The algorithm considers both the pairwise similarities of entities (homology) and the organization of networks (topology) for the final alignment. In order to account for topological similarity, this section describes the notion of neighborhood for each compatibility class.

[1] See `http://www.wired.com/threatlevel/2009/09/fbi-nsac/`
[2] See `http://radar.oreilly.com/archives/2006/12/web20\compact.html`
[3] See `http://www.ornl.gov/sci/techresources/Human_Genome/home.shtml`

After that, the method creates support matrices which allow the use of neighborhood information [12]. Both the pairwise similarities of entities and the organization of these entities together with their interactions provide us great information about the functional correspondence and evolutionary similarity of metabolic networks. Hence, an accurate alignment strategy needs to combine these factors cautiously. In this subsection, we describe a strategy to achieve this combination.

### 5.2 Estimations in Gene Expression

Biology and bioinformatics problems can be formulated as optimization problems either single or multi-objective [13,14]. Heuristic search techniques are required, hopwever; Population-based search algorithms are being used to improve performance by finding the global optimum. Unlike classical optimization methods, population based optimization methods do not limit their exploration to a small region of the solution space and are able to explore a vast search space. In this book we provide a chapter, titled 'Estimation of Distribution Algorithms in Gene Expression Data Analysis' that discusses the a class of evolutionary optimization algorithms with different probabilistic model building methods used in order to explore the search space. Other techniques include: variance, mixture-model and bayesian missing value estimation models. Next we outline other chapters in the book.

## 6 Chapters Included in the Book

This book includes fifteen chapters. Each chapter is self-contained and is briefly described below. Chapter 1 by Holmes et al. provides an introduction to data mining and presents an overview of each successive chapters in this book.

Chapter 2 by Giulia Bruno and Paolo Garza explores temporal data mining in the medical context. In this paper the most popular temporal data mining algorithms are presented and an overview of their applications in different medical domains is given.

Chapter 3 by Min Song and Prat Tanapaisankit tackles extracting key-phrases from full-text documents using a novel unsupervised key phrase extraction system, called BioKeySpotter.

Chapter 4 by Ian Duncan presents his work on mining health claims data for assessing patient risk. A review of the likelihood of individuals of experiencing higher health resource utilization (either because of the use of more services, or more expensive services, or both) than other similarly-situated individuals is presented.

In Chapter 5 Ferhat Ay, Gunhan Gulsoy, and Tamer Kahveci discuss metabolic networks, an important type of biological network. Two algorithms and the results of their application are presented. The Further Reading section of this paper provides an invaluable resource for advanced study.

Chapter 6 by Elham Salehi, Robin Gras discusses the Estimation of Distribution Algorithm (EDA), a relatively new optimization method in the field of evolutionary algorithms. In this chapter, they first provide an overview of different existing EDAs and then review some applications in bioinformatics and finally discuss a specific problem that has been solved with this method.

Chapter 7 by Erliang Zeng, Chris Ding, Kalai Mathee, Lisa Schneper, and Giri Narasimhan focuses on investigating the semantic similarity between genes and its applications. A novel method to evaluate the support for PPI data based on gene ontology information is proposed.

In Chapter 8 Thanh-Phuong Nguyen and Tu-Bao Ho present a study of mining multiple data to reconstruct construct signal transduction networks STN. The experimental results demonstrate that the proposed method can construct STN effectively. Their method has promising applications in several areas, including constructing signal transduction networks from protein-protein interaction networks.

In Chapter 9 Xia Jiang, Shyam Visweswaran and Richard E. Neapolitan develop a specialized Bayesian network model for representing the relationship between features and disease, and a Bayesian network scoring criterion tailored for this model. They also develop an enhancement of Chickerings Greedy Equivalent Search, called Multiple Beam Search.

With D.B. Skillicorns work in Chapter 10, we move on to a different area of application. Adversarial settings are those in which the interests of those who are analyzing the data and those whose data is being analyzed are not aligned; for example, law enforcement, fraud detection and counterterrorism. Skillicorn investigates knowledge discovery techniques within these settings.

In Chapter 11 Mikolaj Morzy takes us into the realms of online communities of Internet forum users. An overview of Internet forums, their architecture, and characteristics of social-driven data is given and some of the challenges that need to be addressed are aired. A framework for analysis and mining of Internet forum data for social role discovery is given.

Chapter 12 explores another application of data mining techniques as Bettina Berendt looks at data mining for information literacy, arguing that data mining can be a means to help people better understand, reflect and influence information and information-producing and -consuming activities that they are surrounded by in todays knowledge societies.

Chapter 13 sees Rudy Setiono, Bart Baesens and David Martens presenting methods for extracting rules from neural networks and support vector machines. They are particularly interested in the business intelligence application domain of credit scoring, and effective tools for distinguishing between good credit risks and bad credit risks.

Chapter 14 by Andriy Andreev and Argyris Argyrou uses Self-organizing maps for data mining. They present a synthesis of the pertinent literature as well as demonstrate, via a case study, how SOM can be applied in clustering and visualizing accounting databases.

In Chapter 15 we see Khan Muhammad Badruddin, Isao Yagi and Takao Terano applying data mining techniques to assess steel plant operation conditions . Their paper describes methodology (rather than results) on the basis of which good classifiers are expected to be discovered and shows that their technique on flattened time series data has potential for good classifier discovery. It concludes with a discussion of some of the commercially-available grouper models used for assessing health risk.

## 7 Conclusion

This chapter presents a collection of selected contribution of leading subject matter experts in the field of data mining. This book is intended for students, professionals and academics from all disciplines to enable them the opportunity to engage in the state of art developments in:

- Temporal Pattern Mining for Medical Applications;
- BioKeySpotter: an Unsupervised Keyphrase Extraction Technique in the Biomedical Full-text Collection;
- Mining Biological Networks for Similar Patterns;
- Estimation of Distribution Algorithms in Gene Expression Data Analysis;
- Gene function prediction and functional network: the role of gene ontology;
- Mining Multiple Biological Data for Reconstructing Signal Transduction Networks;
- Mining Epistatic Interactions from High-Dimensional Data Sets;
- Knowledge Discovery in Adversarial Settings;
- Analysis and Mining of Online Communities of Internet Forum Users;
- Data Mining for Information Literacy;
- Rule extraction from Neural Networks and Support Vector Machines for Credit Scoring;
- Using Self-Organizing Map for Data Mining: A Synthesis with Accounting Applications; and
- Applying Data Mining Techniques to Assess Steel Plant Operation Conditions.

Readers are invited to contact individual authors to engage with further discussion or dialog on each topic.

## References

1. Lauralee, S.: Human Physiology - From cells to systems, 7th edn. Brooks - Cole, Belmont (2007)
2. Vaidya, S., Jain, L.C., Yoshida, H.: Advanced Computational Intelligence Paradigms in Healthcare - 2, 1st edn. Springer Publishing Company (incorporated), Heidelberg (2007)
3. Brahnam, S., Jain, L.C. (eds.): Advanced Computational Intelligence Paradigms in Healthcare 5: Intelligent Decision Support Systems. Springer, Heidelberg (2011)
4. Brahnam, S., Jain, L.C. (eds.): Advanced Computational Intelligence Paradigms in Healthcare 6: Virtual Reality in Psychotherapy, Rehabilitation, and Assessment. Springer, Heidelberg (2011)
5. Liu, B.: Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data. Springer, Heidelberg (2007)
6. Berry, M., Linoff, G.: Data mining techniques: for marketing, sales, and customer relationship management. Wiley, New York (2004)
7. McElheny, V.K.: Drawing the Map of Life: Inside the Human Genome Project. Merloyd Lawrence, New York (2010)
8. Sargent, R., Fuhrman, D., Critchlow, T., Di Sera, T., Mecklenburg, R., Lindstrom, G., Cartwright, P.: The design and implementation of a database for human genome research. In: Svenson, P., French, J. (eds.) Eighth International Conference on Scientific and Statistical Database Management, pp. 220–225. IEEE Computer Society Press, Los Alamitos (1996)

9. Seiffert, U., Jain, L., Schweizer, P. (eds.): Bioinformatics Using Computational Intelligence Paradigms. Studies in Fuzziness and Soft Computing. Springer, Heidelberg (2011)
10. Pinter, R.Y., Rokhlenko, O., Yeger-Lotem, E., Ziv-Ukelson, M.: Alignment of metabolic pathways. Bioinformatics 21(16), 3401–3408 (2005)
11. Tohsato, Y., Matsuda, H., Hashimoto, A.: A Multiple Alignment Algorithm for Metabolic Pathway Analysis Using Enzyme Hierarchy. In: Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology, pp. 376–383. AAAI Press, Menlo Park (2000)
12. Sridhar, P., Song, B., Kahveci, T., Ranka, S.: Mining metabolic networks for optimal drug targets. In: Pacific Symposium on Biocomputing, Stanford, CA, vol. 13, pp. 291–302 (2008)
13. Cohen, J.: Bioinformatics - an introduction for computer scientists. ACM Computing Surveys 36, 122–158 (2004)
14. Handl, J., Kell, D.B., Knowles, J.D.: Multiobjective optimization in bioinformatics and computational biology. IEEE/ACM Transactions on Computational Biology and Bioinformatics 4, 279–292 (2007)

# Chapter 2
# Temporal Pattern Mining for Medical Applications

Giulia Bruno and Paolo Garza

Politecnico di Torino,
Corso Duca degli Abruzzi 24, 10129 Torino, Italy
{giulia.bruno,paolo.garza}@polito.it

**Abstract.** Due to the increased availability of information systems in hospitals and health care institutions, there has been a huge production of electronic medical data, which often contains time information. Temporal data mining aims at finding interesting correlations or sequential patterns in sets of temporal data and has recently been applied in the medical context. The purpose of this paper is to describe the most popular temporal data mining algorithms and to present an overview of their applications in different medical domains.

## 1 Introduction

With recent proliferation of information systems in modern hospitals and health care institutions, there is an increasing volume of medical-related information being collected. Important information is often contained in the relationships between the values and timestamps of sequences of medical data. The analysis of medical time sequence data across patient populations may reveal data patterns that enable a more precise understanding of disease manifestation, progression, and response to therapy, and thus could be of great value for clinical and translational research [16]. Appropriate tools are needed to extract relevant and potentially fruitful knowledge from these previously problematic general medical data.

Temporal data mining can be defined as the activity of looking for interesting correlations or patterns in large sets of temporal data. Particularly, the temporal data mining is concerned with the algorithmic means by which temporal patterns are extracted and enumerated from temporal data. It has the capability of mining activity, inferring associations of contextual and temporal proximity, some of which may also indicate a cause-effect association.

In the literature a large quantity of temporal data mining techniques exist, e.g. sequential pattern mining, episodes in event sequences, temporal association rules mining, discovering calendar-based temporal association rules, patterns with multiple granularities mining, cyclic association rules mining, and inter-transaction associations. Some of them have been applied to temporal medical data to extract temporal patterns.

The aim of this paper is to present an overview of the techniques that deal with temporal pattern extraction and their usage in different medical application domains, to provide a guide to help practitioners looking for solutions to concrete problems. Other overviews of temporal pattern extraction have appeared in the literature [3], but they are not specific for the medical context.

D.E. Holmes, L.C. Jain (Eds.): Data Mining: Found. & Intell. Paradigms, ISRL 25, pp. 9–18.
springerlink.com 

## 2 Types of Temporal Data in Medical Domain

Medical temporal data can be analyzed by considering three dimensions: (i) the type of temporal information available, (ii) the number of sequences to analyze, and (iii) the number of allowed items for event. Another distinction can be made between continuous and categorical values of items. A sequence composed by a series of nominal symbols from a particular alphabet is usually called a temporal sequence and a sequence of continuous, real-valued elements, is known as a time series. Depending on the type of value, the approaches to solve the problem may be quite different.

However, since continuous values are usually transformed into categorical values before the extraction process, we do not consider techniques specific for time series analysis (e.g., signal processing methods, dynamic time wrapping).

The type of temporal data, according to the type of available temporal information, can be grouped in the following three types.

**Ordered Data.** This category includes ordered, but not timestamped, collections of events. Hence, a sorted list of events is available, but not the information about the timestamps of the events.

**Timed Data.** A timed sequence of data. For each event a timestamp is associated. Hence, it possible to sort the events but also to compute the time distance between two events.

**Multi-timed Data.** Each event may have one or more time dimensions, such as either or both a transaction-time or valid-time history. In this case it also possible to compute overlapping between events.

Each of the considered type is a specialization of the data type above it and possesses all its properties. Hence, timed data type is a specialization of ordered data, and Multi-timed data is a specialization of timed data.

Temporal data sets can also be organized in two groups depending on the number of sequences.

**Single Sequence Data Set.** In this case the data set is composed of one single sequence.

**Multiple Sequences Data Set.** In this case a set of sequences composes the data set.

Example of medical applications which involve the different sequence types and data set types described above is reported in Table 1.

**Table 1.** Example of medical applications for each data type

| | | **Sequence types** | | |
|---|---|---|---|---|
| | | **Ordered data** | **Timed data** | **Multi-timed data** |
| **Dataset types** | **Single sequence** | DNA (one item for each event)<br><br>Microarray (a set of items for each event) | Single patient monitoring | Patient therapy |
| | **Multiple sequences** | Set of DNA sequences | Patients' exam logs<br>Disease trends | Patients' activities |

## 3 Definitions

In this section, some background knowledge is introduced, including the definitions of item, sequence, and data set.

**Definition 1.** Let $I = \{i_1, i_2, .., i_k\}$ be a set of items comprising the alphabet. An *event* (or an *itemset*) is a not-empty set of items.

**Definition 2.** A *sequence* $<s_1 s_2 .. s_l>$ is an ordered list of events, where $s_j$ is an event.

**Definition 3.** A *data set* is a not-empty set of sequences.

**Definition 4.** A sequence $a = <a_1 a_2 .. a_n>$ is a *subsequence* of $b = <b_1 b_2 .. b_m>$ (i.e., a is *contained* in b) if there exist integers $1 \leq j_1 < j_2 < .. < j_n \leq m$ such that $a_1 \subseteq b_{j1}$, $a_2 \subseteq b_{j2}$, .. , $a_n \subseteq b_{jn}$.

**Definition 5.** The *support* of a sequence is the frequency of the sequence in the data set.

The definition of frequency of a sequence in a data set depends on the data set type (one sequence, or multiple-sequences) and on the type of analysis. For each type of analysis and data set, we will define more formally the concept of frequency in Section 4.

**Definition 6.** A sequence is *frequent* if it holds in the data set with a support at least equal to *minsup* in the data set, where *minsup* is a parameter of the mining process.

**Definition 7.** A *sequence rule* is a rule in the form $X \Rightarrow Y$ where both X and Y are frequent sequences.

Sequence rules are usually characterized by the confidence and the support quality measures.

**Definition 8.** The *support* of a sequence rule is given by the frequency of the sequence X followed by Y (i.e., the support of the sequence <XY>).

**Definition 9.** The *confidence* of a sequence rule is given by the support of X divided by the support of the rule.

The confidence of a rule represents the likelihood of Y occurring after X (i.e., the conditional probability of Y given X).

The support of a rule sequence is usually used to identify statistically significant rules, while the confidence of a rule represents the strength of the rule (strength of X implies Y).

## 4 Temporal Pattern Mining Algorithms

Two main classes of temporal pattern mining algorithms have been proposed in the previous work. Algorithms of the first class mine frequent temporal patterns from a set of input sequences (i.e., the input data set is composed of multiple sequences). Algorithms of the second class mine frequent temporal patterns from one single sequence (i.e., the input data set is composed of one single sequence).

We will address both classes of algorithms in this paper. However, we investigate more deeply the first class of algorithms since the problem of mining temporal patterns from a set of sequences is considered more interesting and it is useful in many context domains.

### 4.1 Temporal Pattern Mining from a Set of Sequences

The problem of sequence mining from a set of sequences consists of mining those sequences of events which are included in at least a minimum number of sequences of the input data set. More formally, given a data set of sequences $D$ and a minimum support threshold *minsup*, the frequent sequence mining problem consists of extracting all those sequences with a support at least equal to *minsup*. Given an arbitrary sequence *seq*, the support of *seq* in $D$ is given by the number of sequences $s$ in $D$ such that $seq \subseteq s$ (i.e., *seq* is a subsequence of $s$).

Different approaches have been proposed to efficiently solve this problem. The proposed algorithms can be categorized into three main classes: (i) level-wise algorithms based on an horizontal data set representation (e.g., [2]), (ii) level-wise algorithms based on a vertical data set representation (e.g., [17]), and (iii) projection-based pattern growth algorithms (e.g., [14][15]). In this section, one representative algorithm for each main class is described.

#### 4.1.1 Level-Wise Algorithms and Horizontal Data Representation

Similarly to itemset mining algorithms based on a level-wise approach [1], also sequence mining level-wise approaches use an iterative approach to mine sequences [2]. During the first step of the mining process, the level-wise algorithm mines the frequent sequences of length one (i.e., composed of one event), and then, step by step, longer sequences are generated by considering the already mined frequent sequences. In particular, by using an iterative approach, frequent sequences of length $k$ are used to generate candidate sequences of length $k$+1. At each step, the support of the candidate sequences is computed by scanning the input data set and the frequent sequences are selected. The extraction process ends when the generated candidate set is empty.

Figure 1 reports a pseudo-code of the mining process. The set of frequent events is generated by performing a scan of the data set (line 1). Each frequent event corresponds to one sequence of length one and it is used to initialize $L_1$ (i.e., the set of frequent sequences of length 1). Candidate sequences of length 2 are generated by combining frequent sequence of length 1 (line 3). The algorithm loops until candidates are available (lines 4-9). Each iteration analyzes sequences of a specific length. In particular, at iteration $k$ sequences of length $k$ are mined and candidate sequences of length $k$+1 are generated. At each iteration, a scan of the data set is performed to compute the support of each candidate sequence (line 5). Then, frequent sequences of length $k$ are inserted in $L_k$ (i.e., the set composed of frequent $k$-sequences) (line 6). Finally, candidate sequences of length $k$+1 are generated by combining frequent sequences of length $k$ (line 7). The whole set of frequent sequences $L$ is obtained by joining the $L_k$ sets (line 10).

The proposed approach is based on the initial computation of frequent events. Since each event is a set of items (i.e., an itemset), mining the set of frequent events corresponds to mine frequent itemsets. To efficiently perform the generation of $L_1$, Agrawal et al. [2] use an itemset mining algorithm [1]. Only frequent itemsets are needed as building blocks to extract frequent sequences.

Traditional level-wise approaches (e.g., AprioriAll [2]) are based on a horizontal representation of the input data set. Each row of the dataset represents a sequence and it is identified by a sequence id. Every time the support of an itemset has to be

computed, the dataset is entirely scanned and the number of matching sequences is counted. This operation can be computational expensive, in particular when large data set are considered. To address this problem, a set of level-wise approaches based on a vertical data representation have been proposed. In the following section, an approach based on a vertical data representation is discussed.

**Input**: a data set of sequences (*D*), a minimum support threshold (*minsup*)
**Output**: the set of frequent sequences (*L*) mined from *D*

1. $L_1$ = {frequent sequences of length 1}
2. $k = 2$
3. $C_k$ = generate_candidate_sequences ($L_{k-1}$)
4. WHILE ($C_k$ is not empty) {
5. scan *D* and count the support of each candidate-sequence in $C_k$
6. $L_k$ = set of candidates in $C_k$ with a support at least equal to *minsup*
7. $C_{k+1}$ = generate_candidate_sequences ($L_k$)
8. $k = k+1$
9. }
10. *L* = union of sequences in $L_k$
11. return *L*

**Fig. 1.** Sequence mining: pseudo-code of a level-wise approach

#### 4.1.2 Level-Wise Algorithms and Vertical Data Representation

To avoid multiple scans of the data set, or at least to limit the number of scans, a different representation of the input data set can be exploited. Some approaches (e.g., SPADE [17]) use a vertical data representation to improve the frequent sequence mining process. The vertical data representation stores for each item the identifiers of the set of sequences including the item and its "position" in each sequence. In particular, for each item a set of pairs (*sid,eid*) is stored, where *sid* is a sequence-identifier and *eid* the unique identifier of the item in the sequence. The time stamp is usually used to uniquely identify items in the sequences. Given a sequence, the value of *eid* is usually used to represent the order of the items in the sequence, but it can also be used to compute the distance between two events.

By exploiting the vertical representation, and some closure properties, more efficient level-wise sequence mining algorithms have been proposed (e.g., SPADE).

#### 4.1.3 Projection-Based Pattern Growth Algorithms

To improve the efficiency of the itemset mining algorithms, pattern growth approaches [10] have been proposed. These approaches allow significantly reducing the execution time by bounding the search space. An analogous approach has been proposed in the sequence mining context. Pei et al. [15] proposed PrefixSpan. PrefixSpan is based on a pattern growth approach. In particular, it recursively mines the set of patterns. At each recursion a projected data set (a projected set of sequences) is generated by considering the sequences mined so far. Differently from the level-wise approaches, PrefixSpan does not generate candidate sequences. Hence, PrefixSpan is significantly faster than level-wise approaches.

### 4.1.4 Sequence Mining with Constraints

The previously described algorithms mine all frequent sequences by exclusively enforcing a minimum support threshold. However, in some contexts, different constraints are essential. For instance, in many cases, it is useful to enforce a constraint on the maximum gap between two events of the mined sequences. A naïve approach to solve this problem could be the enforcing of the constraints as a post-processing step. However, this approach is not efficient. In [14] a set of sequence constraints (e.g., presence or absence of an item, sequence length or duration) are defined and some of them are pushed into a sequence mining algorithm based on PrefixSpan. Differently from [14], Orlando et al. [13] focused their attention exclusively on the gap constraint. The proposed algorithm efficiently mines sequences by enforcing a constraint on the minimum and maximum gap (timestamp difference) between two consecutive events of the mined sequences. The authors of the paper [13] focused their attention on the gap constraint because it is probably the most frequent and useful constraint in many real applications.

## 4.2 Temporal Pattern Mining from a Single Sequence

In some biological and medical applications (e.g., DNA analysis), the identification of frequent recurring patterns in a sequence allows highlighting interesting situations. The analysis of the mined recurring subsequences by domain experts allows learning new information about some biological mechanisms or confirming some hypothesis.

The extraction of recurrent subsequences from a single input sequence can be formalized as follows.

Given a sequence *seq*, a window size *winsize*, and a minimum frequency threshold *minfreq*, the frequent subsequence mining problem consists of extracting all those subsequences of *seq* such that each extracted subsequence holds in *seq* with a frequency at least equal to *minfreq*.

The window size parameter is used to "partition" the input sequence. A set of windows of events of size *winsize* is generated by applying a sliding window of size *winsize* on the input sequence. The definition of this set of windows of events allows defining the support (frequency) of a subsequence. The support of a subsequence *s* is given by the number of windows including *s*.

Mannila et al. [11] proposed a set of efficient algorithms for discovering frequently occurring episodes (i.e., subsequence of events) in a sequence. The proposed algorithms were used in telecommunication alarm systems, but they can also be used in medical applications.

The proposed algorithms are based on a level-wise approach. During the first step the frequent single events are extracted. Then, longer subsequences are mined by extending shorter frequent subsequences. In particular, frequent subsequences of length $k$ are used to generate candidate subsequences of length $k$+1. The support of the candidate subsequences is computed by performing a complete scan of the input sequence.

This algorithm is based on the same idea of AprioriAll [2] but it works on a different type of data (in this case the input dataset is composed of only one sequence) and hence it mines a different type of frequent sequences.

## 5 Medical Applications

In the following subsections, some applications based on temporal pattern mining algorithms in medical context are described.

**Time-Annotated Sequences for Medical Data Mining**
One of the typical structures of medical data is a sequence of observations of clinical parameters taken at different time moments. In [6] authors applied the time-annotated sequences (TAS) to discover frequent patterns describing trends of biochemical variables along the time dimension. TAS are sequential patterns where each transition between two events is annotated with a typical transition time that is found frequent in the data. Extracted patterns are in the form *A - B [t1, t2]*, where *A* and *B* are itemsets (i.e., events) and *t1* and *t2* are the minimum and the maximum time delay allowed to consider A and B forming a sequential pattern. The approach has received a positive feedback from physicians and biologists as the extracted patterns represented additional evidences of the effectiveness of the applied therapy for liver transplantation (the study involved 50 patients, 38 variables, and 40 time instances). The algorithm adopts a prefix-projection approach to mine candidate sequences, and it is integrated with an annotation mining process that associates sequences with temporal annotations. This work represents a way of considering the time information in the pattern extraction process. However, the patterns are limited to length two.

**Sequential Data Mining in Development of Atherosclerosis Risk Factors**
In [9] authors compare three sequential based approaches applied to observations recording the development of risk factors and associated conditions, to identify frequent relations between patterns and the presence of cardiovascular diseases. The original continuous values are discretized in the form of trends (e.g., increasing, decreasing, etc.) for the windowing method, in both trend and fixed intervals (e.g., low, medium, and high) for the inductive logic and episode rule methods. Then, classification rules are extracted by all the three methods in the form *IF* (a set of conditions appears) *THEN* (there is/there is not cardiovascular disease risk). In the conditions, the time information regards the position of events in the patient sequence (e.g., beginning, middle, end) and the relative position among events (e.g., before, after). Thus, these methods require a lot of pre-processing before the extraction and it seems that there none of them clearly outperforms the others.

**Temporal Rules from Antenatal Care Data**
A dataset of clinical variables of pregnant mothers is analyzed to predict the presence of risk factors [12]. The dataset is in the form of transactional data, i.e. each transaction row contains a patient id, a time point t, and the set of signs observed at time t. Only the frequent events are considered. The authors, first generalized the events by specifying the first and the last point time in which the event is present, and then extract rules from the resulting temporal relation. The rules are extracted by first constructing a tree in which each node is a sign type and each edge shows the temporal relation between two signs (e.g., before, overlap, during), and then forming a graph by using the relations which satisfy the minimum support. In the resulting graph, each edge represents a frequent temporal relation between two events. A first critical state is in the event generalization because the discontinuity of events is not considered and

only the first and the last time point are reported. Furthermore, only rules involving two signs are extracted, and longer frequent sequences are not considered.

**Temporal Gene Networks**

A gene regulatory network aims at representing relationships that govern the rates at which genes in the network are transcribed into mRNA. Genes can be viewed as nodes whose expression levels are controlled by other nodes. In the analysis of gene expression data, an expression profile can represent a single transaction, and each gene a single item. In an expression profile each gene is assigned a real value which specifies the relative abundance of that gene in the sample. Hence, in applying association rules to gene expression data, there is the need to discretize the domain by defining appropriate value intervals. Binning the values helps to alleviate problems with noise, by focusing the analysis on the more general effects of genes. Usually gene expression data are binned into two levels, up-regulated (i.e. highly expressed) or down-regulated (i.e. inhibited) according to specific fixed thresholds. Sometimes there is also a third state (neither up nor down regulated) which is not considered in the analysis.

In [5], authors propose a method to infer association rules from gene expression time series data sets which represent interactions among genes. The temporal rules are in the form *G1(a)->G2(b)+k*. It means that if gene *G1* is equal to *a*, then after *k* units of time gene *G2* will be equal to *b*. The gene expression data are associated to an expression level by using a suitable function. Different techniques (i.e., clustering techniques or fixed thresholds) are exploited. The discretized data are organized in a time-delay matrix and analyzed by the Apriori algorithm, which extracts the association rules. Rules are reduced and evaluated by means of an appropriate quality index.

**Unexpected Temporal Associations in Detecting Adverse Drug Reactions**

In medical application, it can be useful mining unanticipated episodes, where certain event patterns unexpectedly lead to outcomes. In [8], authors address the problem of detecting unexpected temporal association rules to detect adverse drug reaction signals from healthcare administrative databases. The extracted rules are in the form *A -C (T)*, i.e., an event *A* unexpectedly occurs in a *T*-sized period prior to another event *C*. The basic idea is to first exclude expected patterns in individual *T*-sized subsequences, and then aggregate unexpectedness. Also in this case, the extracted rules are of lengths 2 and are of type Drug implies Condition.

**Frequent Closed Sequences of Clinical Exams**

Standard medical pathways have been defined as care guideline for a variety of chronic clinical conditions. They specify the sequence and timing of actions necessary to provide treatments to patients with optimal effectiveness and efficiency. The adoption of these pathways allows health care organizations to control both their processes and costs.

Real pathways (i.e., the pathways actually followed by patients) deviate from predefined guidelines when they include different or additional exams, or some exams are missing. When not justified by specific patient conditions, these non compliant pathways may cause additional costs without improving the effectiveness of the treatment. Non compliant pathways may be due to patient negligence in following the prescribed treatments, or medical ignorance of the predefined guidelines, or incorrect

procedures for data collection. For example, to speed up the registration process, the operator may only record a subset of the exams actually performed by the patient. On the other hand, available guidelines may not provide procedures to cover some particular (e.g., rare or new) diseases. By analyzing the electronic records of patients, the medical pathways commonly followed by patients are identified. This information may be exploited to improve the current treatment process in the organization and to assess new guidelines.

In [4] the analysis of patients' exam log data is addressed to rebuild from operational data an image of the steps of the medical treatment process. The analysis is performed on the medical treatment of diabetic patients provided by a Local Sanitary Agency in Italy. The extracted knowledge allows highlighting medical pathways typically adopted for specific diseases. Detected medical pathways include both the sets of exams which are frequently done together, and the sequences of exam sets frequently followed by patients. The proposed approach is based on the extraction of the frequent closed sequences by exploiting a modified version of PrefixSpan, which provide, in a compact form, the medical pathways of interest.

**Motif Extraction**

In the bioinformatics field, an important task is to extract important motifs from sequences. A motif is a nucleotide or amino-acid sequence pattern that is widespread and has a biological significance. The challenges for motif extraction problem are two-fold: one is to design an efficient algorithm to enumerate the frequent motifs, and the other is to statistically validate the extracted motifs and report the significant ones. In [18], authors propose an algorithm, called EXMOTIF, that given a sequence and a structured motif template, extracts all frequent structured motifs. An example of extracted motif is *MT[115,136]MTNTAYGG[121,151]GTNGAYGAY*. Here *MT*, *MTNTAYGG* and *GTNGAYGAY* are three simple motifs, while *[115,136]* and *[121,151]* are variable gap constraints ([minimum gap, maximum gap]) allowed between the adjacent simple motifs.

**Patterns of Temporal Relations**

Temporal relations can be more complex than the simple chaining of events. However, the more temporal relations are used, the more the complexity of the process is increased. In [7], authors propose an approach to extract patterns of temporal relations between events. They use the Fuzzy Temporal Constraint Networks formalism, a temporal abstraction algorithm to work at a higher knowledge level and to reduce the volume of data, and then they exploit a temporal data mining algorithm inspired on Apriori to discover more informative temporal patterns, which applies temporal constraint propagation for pruning non-frequent patterns. The temporal relations are in the form <event, relation, event>, where the relation stands for one of the Allen's temporal relation such as before, starts, during, etc.

## 6 Conclusions

Temporal pattern mining has been successfully exploited in medical domains. In this paper, we presented a description of some of the most popular temporal data mining algorithms used in literature. Furthermore we illustrated the exploitation of temporal data mining algorithms in several real medical applications.

## References

[1] Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules in Large Databases. In: Proceedings of 20th International Conference on Very Large Data Bases, Santiago de Chile, Chile, pp. 487–499 (1994)
[2] Agrawal, R., Srikant, R.: Mining Sequential Patterns. In: Proceedings of the Eleventh International Conference on Data Engineering, Taipei, Taiwan, pp. 3–14 (1995)
[3] Antunes, C., Oliveira, A.: Temporal data mining: an overview. In: Proc. Workshop on Temporal Data Mining KDD 2001 (2001)
[4] Baralis, E., Bruno, G., Chiusano, S., Domenici, V.C., Mahoto, N.A., Petrigni, C.: Analysis of Medical Pathways by Means of Frequent Closed Sequences. In: Setchi, R., Jordanov, I., Howlett, R.J., Jain, L.C. (eds.) KES 2010. LNCS, vol. 6278, pp. 418–425. Springer, Heidelberg (2010)
[5] Baralis, E., Bruno, G., Ficarra, E.: Temporal Association Rules for Gene Regulatory Networks. In: 4th International IEEE Conference of Intelligent Systems, Varna, Bulgaria (2008)
[6] Berlingerio, M., Bonchi, F., Giannotti, F., Turini, F.: Time-annotated Sequences for Medical Data Mining. In: 7th IEEE International Conference on Data Mining, pp. 133–138 (2007)
[7] Campos, M., Palma, J., Marin, R.: Temporal Data Mining with Temporal Constraints. In: Proceedings of the 11th Conference on Artificial Intelligence in Medicine, Amsterdam, The Netherlands, pp. 67–76 (2007)
[8] Jin, H., Chen, J., He, H., Williams, G.J., Kelman, C., O'Keefe, C.M.: Mining Unexpected Temporal Associations: Applications in Detecting Adverse Drug Reactions. IEEE Transactions on Information Technology in Biomedicine 12(4), 488–500 (2008)
[9] Klema, J., Novakova, L., Karel, F., Stepankova, O., Zelezny, F.: Sequential Data Mining: A Comparative Case Study in Development of Atherosclerosis Risk Factors. IEEE Transactions on Systems, Man, and Cybernetics 38(1), 3–15 (2008)
[10] Han, J., Pei, J., Yin, Y.: Mining Frequent Patterns without Candidate Generation. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, Dallas, Texas, USA, pp. 1–12 (2000)
[11] Mannila, H., Toivonen, H., Verkamo, A.I.: Discovery of Frequent Episodes in Event Sequences. Data Mining and Knowledge Discovery 1(3), 259–289 (1997)
[12] Meamarzadeh, H., Khayyambashi, M.R., Saraee, M.H.: Extracting Temporal Rules from Medical data. In: International Conference on Computer Technology and Development (2009)
[13] Orlando, S., Perego, R., Silvestri, C.: A new algorithm for gap constrained sequence mining. In: Proceedings of the 2004 ACM Symposium on Applied Computing, Nicosia, Cyprus, pp. 540–547 (2004)
[14] Pei, J., Han, J., Wang, W.: Mining Sequential Patterns with Constraints in Large Databases. In: Proceedings of the 2002 ACM CIKM International Conference on Information and Knowledge Management, VA, USA, pp. 18–25 (2002)
[15] Pei, J., Han, J., Mortazavi-Asl, B., Wang, J., Pinto, H., Chen, Q., Dayal, U., Hsu, M.-C.: Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach. IEEE Transactions on Knowledge and Data Engineering 16(10), 1–17 (2004)
[16] Post, A.R., Harrison, J.H.: Temporal Data Mining. Clinics in Laboratory Medicine 28, 83–100 (2008)
[17] Zaki, M.J.: SPADE: An Efficient Algorithm for Mining Frequent Sequences. Machine Learning 42, 31–60 (2001)
[18] Zhang, Y., Zaki, M.J.: EXMOTIF: efficient structured motif extraction. Algorithms for Molecular Biology 1(21) (2006)

# Chapter 3
# BioKeySpotter: An Unsupervised Keyphrase Extraction Technique in the Biomedical Full-Text Collection

Min Song and Prat Tanapaisankit

Information Systems, New Jersey Institute of Technology
{min.song,pt26}@njit.edu

**Abstract.** Extracting keyphrases from full-text is a daunting task in that many different concepts and themes are intertwined and extensive term variations exist in full-text. In this chapter, we proposes a novel unsupervised keyphrase extraction system, BioKeySpotter, which incorporates lexical syntactic features to weigh candidate keyphrases. The main contribution of our study is that BioKeySpotter is an innovative approach for combining Natural Language Processing (NLP), information extraction, and integration techniques into extracting keyphrases from full-text. The results of the experiment demonstrate that BioKeySpotter generates a higher performance, in terms of accuracy, compared to other supervised learning algorithms.

## 1 Introduction

In recent years, there has been a tremendous increase in the number of biomedical documents in the digital libraries that provide users with access to the scientific literature. For example, the PubMed digital library currently contains over 18 million citations from various types of biomedical documents published in the past several decades (www.pubmed.gov). With the rapid expansion of the number of biomedical documents, the ability to effectively determine the relevant documents from a large dataset has become increasingly difficult for users. Examining the complete contents of a document to determine whether the document would be useful is a challenging and often time consuming task. A more efficient approach would allow a reader to understand the concept of the document by using short semantic metadata like keyphrases. It is even more useful if concise semantic metadata is available to users in digesting full-text documents.

It is much harder to extract keyphrases from biomedical full-text documents than from abstract corpus since, 1) many different concepts and themes are intertwined in full-text and 2) extensive lexical variations of terminology reside in full-text.

To address and resolve this challenging research problem, an unsupervised lexical-syntactic approach called BioKeySpotter is proposed to extract keyphrases from biomedical full-text. In this approach, it is hypothesized that keyphrases in biomedical domain are either biological entities or are co-located with biological entities. Detailed description of the algorithm is reported in Section 3.

D.E. Holmes, L.C. Jain (Eds.): Data Mining: Found. & Intell. Paradigms, ISRL 25, pp. 19–27.
springerlink.com 

The major contribution of our study is three-fold: 1) BioKeySpotter is an unsupervised learning technique, which can be easily adapted to other domains, 2) the performance of BioKeySpotter is superior to, or at least equivalent to, classification-based keyphrase extraction algorithms, and 3) BioKeySpotter addresses and resolves the problem with extracting keyphrases from full-text.

The rest of the chapter is organized as follows: Section 2 provides backgrounds in keyphrase extraction and briefly describes related work. Section 3 details the algorithm and architecture of BioKeySpotter. Section 4 reports the experiments and analyzes the results. Section 5 concludes the paper.

## 2 Backgrounds and Related Work

Keyphrase extraction is the task that takes a document as input and automatically generates a list (in no particular order) of keyphrases as output (Turney 1997). Given a document, keyphrases is a list of phrases that represents the document. From machine learning perspective, keyphrase extraction techniques can be roughly categorized into supervised and unsupervised learning approaches. Supervised learning techniques treated keyphrase extraction as a classification problem and the task is to find whether a particular phrase is a keyphrase or not. In supervised learning techniques, once the learning method has generated the model given the training data, it can be applied to unlabeled data. The training documents are used to adjust the keyphrase extraction algorithms to attempt to maximize system performance (Turney 2000).

Many researchers have taken the supervised learning approach. Turney has presented GenEx system which has two components: Genitor and Extractor. Extractor performs extraction task using twelve parameters that determine how it processes the input text (Turney 1997). These parameters are tuned by the Genitor genetic algorithm to maximize performance (fitness) on training data. Witten et al. have implemented Kea which builds on Turney's work but their system uses a Naïve Bayes learning model instead of a genetic algorithm (Witten, Paynter et al. 1999). The learning model employs two features: the TF-IDF weight of the phrases and their relative position within the document. The performance of KEA is comparable to that of Extractor. Song et al. proposed KPSpotter, which employs a new technique combining data mining techniques called Information Gain and several Natural Language Processing techniques such as stemming and case-folding (Song 2003). A disadvantage of supervised approaches is that they require a lot of training data and they are constrained by any specific training documents. Unsupervised approaches could be a possible alternative in this regard.

The unsupervised approaches eliminate the need of training data. They rely solely on a collection of plain non-annotated textual data. These approaches usually select quite general sets of candidates, and use a ranking function to limit the selection to the most important candidates. For example, Barker and Cornacchia restrict candidates to noun phrases, and rank them using heuristics based the number of words and the frequency of a noun phrase, as well as the frequency of the head noun (Barker and Cornacchia 2000). Also, Bracewell et al. restrict candidates to noun phrases (Bracewell, Ren et al. 2005). The noun phrases are extracted from a document, and

then clustered if they share a term. The clusters are ranked based on term and noun phrase frequencies. Finally, the centroids of the top-n ranked clusters are selected as keyphrases of the document. More recently, some researchers have used graph-based ranking methods by having an assumption that a sentence should be important if it is heavily linked with other important sentences, and a word should be important if it is heavily linked with other important words. Mihalcea and Tarau propose a graph-based unsupervised approach, called TextRank, where the graph vertices are tokens and the edges reflect co-occurrence relations between tokens in the document (Mihalcea and Tarau 2004). Candidates are ranked using PageRank and adjacent keywords are merged into keyphrases in a post-processing step. Using the same approach, Wan and Xiao propose an iterative reinforcement approach to do simultaneously keyword extraction and summarization.

In calculating the weight of candidate phrases, the weight is calculated to enable ranking by applying linguistic and/or statistical techniques on domain text such as TFIDF, C/NC-value. TFIDF weighting is the most common statistical method of measuring the weight of a candidate phrase in a document (Zhang, Zincir-Heywood et al. 2005). TFIDF value has been used in the candidate phrase extraction step (Song et al., 2003; El-Beltagy, 2006). C/NC-value is the other method for calculating the weight of candidate phrases used in the biomedical domain, introduced by Frantzi et al (1998). The C/NC-value method combines linguistic (linguistic filter, part-of-speech tagging and stop-list) and statistical analysis (frequency analysis, C/NC-value) to enhance the common statistical techniques (e.g. TFIDF) by making sensitive to a particular type of multi-word term. C-value enhances the common statistical measure of frequency of occurrence for phrase extraction. NC-value gives a method for the extraction of phrase context words and the incorporation of information from phrase context words to the extraction of terms (Frantzi, Ananiadou et al. 1998).

In recent years, more effective systems have been developed to improve the performance of a key phrase extraction by integrating data mining techniques (decision tree, Naïve Bayes, SVMs, etc). For instance, KP-Miner (El-Beltagy 2006), improves TFIDF by introducing two factors (provide higher weights for terms whose length is greater than one and for terms that appear somewhere in the beginning of the document); LAKE (D'Avanzo, Magnini et al. 2004) relies on the linguistic features of a document in order to perform keyphrase extraction through the use of a Naïve Bayes classifier as the learning algorithm and TFIDF term weighting with the position of a phrase as a feature; KPSpotter (Song et al., 2003), uses the information gain measure to rank the candidate phrases based on a TFIDF and distance feature.

## 3 The Proposed Approach

In this section, we describe the architecture and the algorithm of BioKeySpotter. As illustrated in Figure 1, lexical and syntactic features are identified for candidate keyphrases through several modules of BioKeySpotter. In the core extraction engine, candidate keyphrases are ranked by mutual information and similar ones are merged by Markov Random Field (MRF)-based string similarity algorithm.

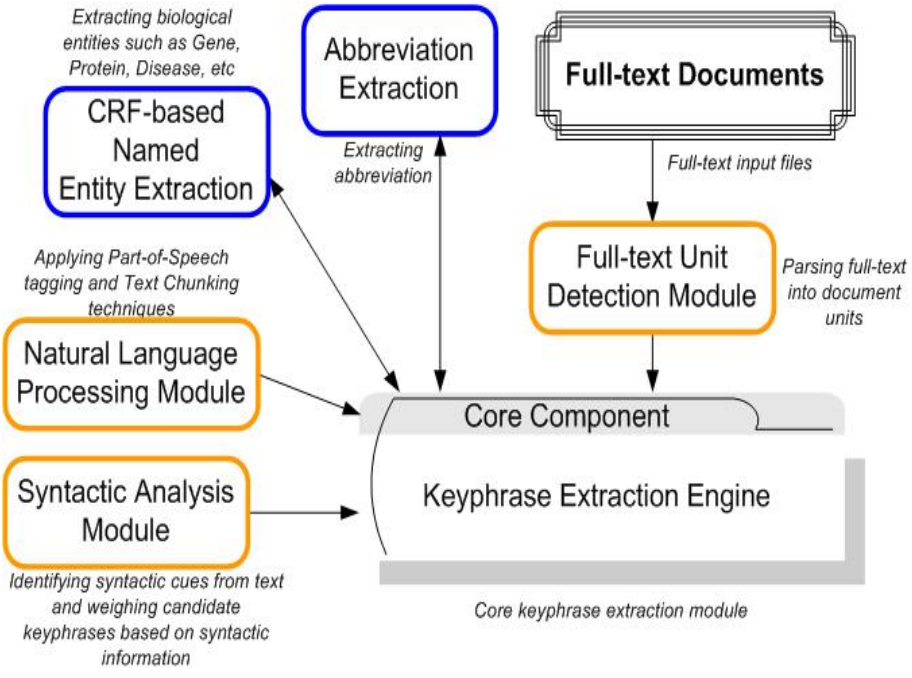



**Fig. 1.** Architecture of BioKeySpotter

The procedure of BioKeySpottor is as follows:

**Step 1:** Parses full text articles. Two main sources of the full text are HTML and PDF files. In this step, original full-text is converted into plain text and identifies full-text units such as section and paragraph. To parse full-text, we employ the template-based parsing technique that is applied to each site that we downloaded full-text from.

**Step 2:** Extracts biological named entities by Conditional Random Field (CRF)-based named entity extraction from sentences. CRF was reported as the best algorithm for Named Entity Recognition (NER) tasks (Settles, 2004).

**Step 3:** Detects abbreviation and its long forms based on the unsupervised abbreviation extraction technique that combines lexical analysis and statistical learning techniques (Song and Liu, 2008).

**Step 4:** Employs a shallow parsing technique to create a dependency tree for sentences. We need this step to identify the shortest path between biological entities and neighbor noun groups. To identify noun groups, we employ the CRF-based text chunking technique.

After step 4 is finished, we assign each candidate keyphrase to one of the six categories illustrated in Figure 2. The main goal is to assign higher weight values to promising candidate keyphrases that belong to the first four categories (category 1 to 4). Category 1 is for candidate keyphrases that are biological entity but not abbreviation. Category 2 is for candidate keyphrases that are biological entity in the form of abbreviation. Category 3 and category 4 are for candidate keyphrases that are co-located with biological entities in a sentence. Category 5 and 6 are for candidate keyphrases that are not co-located with biological entities.


**Fig. 2.** Six Categories of Candidate Keyphrases

**Step 5:** Calculate weight for each candidate keyphrase by pair-wise mutual information formula. The mutual information between two candidate keyphrases xi and xj in a given category is defined as follows:

$$I(x_i, x_j) = \sum_x P(x_i, x_j) \log\left(\frac{P(x_i, x_j)}{P(x_i)P(x_j)}\right)$$

In addition to mutual information weight, we further apply heuristic rules to weight keyphrases from a list of candidate keyphrases. The following two more syntactical rules are applied:

- Rule 1: font size of the candidate phrases. Important concepts tend to be highlighted by different font style or size. We utilize this syntactical information to weigh candidate keyphrases.
- Rule 2: location of candidate phrases. Important concepts are often introduced in abstract and introduction. These concepts also are re-introduced in other sections of full-text. We utilize location information of candidate keyphrases.

The final weight of a candidate keyphrase (CW) is calculated as follows:

$$CW = \alpha * (F + L + I(x_i, x_j))$$

Where α is constant assigned based on the category ranking, F is font information, and L is location information. In our experiment, we assigned 1 for category 2, 0.7 for category 1 and 4, 0.5 for category 3, and 0.3 for category 5 and 6.

**Step 6:** Merge similar candidate keyphrases based on the MRFED string similarity technique (Song et al., 2008). MRFED is a graph model, Markov Random Field, based string similarity. It was originally proposed to detect the duplicate records. For our system, MRFED is tuned to calculate similarity distance among candidate keyphrases.

## 4 Evaluation

In this section, we report our evaluation method, data collection, and experimental results.

### 4.1 Dataset

The data set for the experiments consists of 1000 full-text documents, collected from PubMed (www.pubmedcentral.nih.gov). All of these full-text documents have abstract and keywords assigned by the authors. A sample system output is given in Table 1 which shows 5 keyphrases assigned by the author and extracted by the five techniques.

**Table 1.** Five extracted keyphrases by all five systems

| **PubMed ID:** PMC7601436 | | |
|---|---|---|
| **Author assigned keyphrases** | Cdt1<br>Licensing<br>prereplication | DNA<br>replication<br>geminin |
| **Naïve Bayes** | geminin<br>chromatinal | Cdt1<br>licenses |
| **Linear Regression** | Genetics<br>prevents<br>Institute | Synthesis<br>Xenopus |
| **SVM** | Lutzmann<br>prevents<br>activity | sufficient<br>Xenopus |
| **BKS-W** | Geminin<br>DNA<br>replication<br>complex | Cdt1<br>license |
| **BKS-WC** | Geminin<br>DNA<br>replication<br>synthesis | Cdt1<br>license |

### 4.2 Comparison Algorithms

**Naïve Bayes Classifier:** is popular due to its simplicity, and computational efficiency, and has been widely used for text classification (D'Avanzo, Frixione et al. 2004). The algorithm uses the joint probabilities of words and categories to estimate the probabilities of categories given in a text document. It computes the posterior probability that the text document belongs to different classes and assigns it to the class with the highest posterior probability. The posterior probability of class is computed using the Bayes rule, and the testing sample is assigned to the class with the highest posterior probability. The advantage of Naïve Bayes can be explained in a way which is fast to train and fast to evaluate; the classifier can also handle missing values by ignoring the examples during model building and classification.

**Linear Regression:** is another important classifier algorithm used to analyze biological datasets. It is from regression analysis in which the relationship between one or more independent variables and dependent variables (Arshadi and Jurisica 2005). The goal of

linear regression is to find the line that best predicts the dependent variable from independent variables. To achieve this goal, linear regression finds the line that minimizes the sum of the squares of the vertical distances of the points from the line.

**Support Vector Machines (SVMs):** is relatively new machine learning process, influenced by advances in statistical learning theory. The algorithm is a linear learning system that builds two spate classes (suitable for binary classification tasks), which are generated from training examples. The overall aim is to generalize test-data well. Utilized as binary categorical classifiers, the SVM method performs classification more accurately than most other supervised learning algorithms in biological data analysis applications, especially those applications involving high dimensional datasets (Brown 2000; Liu 2007). The SVM method uses kernel function (the similarity function, which is computed in the original attribute space). The Support Vector Machine can be considered complex, slow and takes a lot of memory (limitation of the kernel), but is a very effective classifier in a wide range of bioinformatic problems, and in particular performs well in analyzing biological datasets (Brown 2000).

### 4.3 Experimental Results

In this section, we will report the experimental results with average (μ) and standard deviation (σ). In our experiments, we compared our method, BKS-W with four other algorithms such as Naïve Bayesian, Linear Regression, SVM, and BKS without categorization (BKS-WC).

We randomly choose a training set of 100 documents. The others are used as a test set. For each document, there is a set of keyphrases, assigned by the authors of the articles. We have made a comparative study of all techniques.

Each of the performances of the five algorithms is tested. Table 2 and Table 3 show the results of the performance of Naïve-Bayes, linear regression, SVM classifiers, BKS-W and BKS-WC on full text documents and abstracts based on the holdout evaluation method. AVG denotes the mean of the number of keyphrases, and STDEV denotes the standard deviation.

The comparative study results show that (1) BKS-C outperforms the other four algorithms when we use the full text documents to extract 5, 10, and 15 keyphrases; (2) BKS-WC performs the second best with marginal difference from BKS-C; (3) SVM outperforms the other supervised learning algorithms, Naïve Bayesian, Linear Regression, but it's performance was not competitive to BKS-C and BKS-WC.

**Table 2.** Performance comparison of five algorithms using full text

| Algorithm | 5 Key phrases | 10 Keyphrases | 15 Keyphrases |
|---|---|---|---|
| | **μ/ σ** | **μ/ σ** | **μ/ σ** |
| Naïve Bayes | 0.95 +/- 0.91 | 1.31 +/- 1.09 | 1.57 +/- 1.19 |
| Linear Regression | 0.98 +/- 0.97 | 1.41 +/- 1.19 | 1.66 +/- 1.30 |
| SVM | 1.00 +/- 0.94 | 1.44 +/- 1.16 | 1.76 +/- 1.28 |
| BKS-C | 1.72+/-1.78 | 1.77+/-1.5 | 1.91+/1.62 |
| BKS-WC | 1.61+/-1.57 | 1.64+/-1.41 | 1.85+/-1.57 |

In the second experiment, our data set consists of 1000 abstracts of the same documents, which were extracted from the full text documents used in the first experiment. In Table 3, the comparative study results show that (1) the performance of BKS-C, BKS-WC, and Naïve Bayesian is similar when we use the abstracts only to extract 5, 10, 15 keyphrases. (2) The performance of SVM is inferior to BKS-C, BKS-WC, and Naïve Bayesian but better than Linear Regression when we use the abstracts to extract 5, 10, 15 keyphrases. (3) Linear regression performs worst compared to others in most of the cases. The performance difference among BKS-W and BKS-WC, and Naïve Bayesian is not statistically significant. Linear regression was the worst performer compared to the best performer in 5, 10 keyphrases by 161.29%, 59.42% respectively. Also when it comes to 15 keyphrases, SVMreg-1 is the worst performer with 50.62%.

**Table 3.** Performance comparison of five algorithms using abstracts

| Algorithm | 5 Key phrases | 10 Keyphrases | 15 Keyphrases |
|---|---|---|---|
| | **μ/ σ** | **μ/ σ** | **μ/ σ** |
| Naïve Bayes | 0.81 +/- 0.84 | 1.10 +/- 1.02 | 1.22 +/- 1.07 |
| Linear Regression | 0.31 +/- 0.56 | 0.69 +/- 0.84 | 0.88 +/- 0.92 |
| SVM | 0.38 +/- 0.61 | 0.69 +/- 0.81 | 0.81 +/- 0.90 |
| BKS-C | 0.79 +/- 0.86 | 1.16 +/- 1.07 | 1.27 +/- 1.12 |
| BKS-WC | 0.80 +/- 0.86 | 1.13 +/- 1.0 | 1.23 +/- 1.05 |

Overall, BioKeySpotter outperforms the other supervised keyphrase extraction techniques both on full-text and abstract. In particular, the performance of BioKeySpotter is outstanding on full-text. With abstracts, however, the performance of BKS is competitive but not appealing. The reason for the inferior performance of BKS on abstracts is because lexical and syntactic features required by BKS are less present in abstracts than in full-text. In addition, based on the experimental results, we are inclined to reject our hypothesis.

## 5 Conclusion

In this chapter, we proposed BioKeySpotter, a novel unsupervised keyphrase extraction technique for biomedical full-text collections. We utilized lexical syntactic features to extract candidate keyphrases. We categorized candidate keyphrases into six different groups and assign different weight to a keyphrase according to the category where each keyphrase belongs to. We chose pair-wise mutual information to calculate weight of each keyphrase. We applied the MRF-based string similarity technique to merge similar phrases. Our experimental results show that BioKeySpotter outperforms the other three supervised keyphrase extraction techniques. We are currently conducting a new set of experiments with larger full-text collections. We are also exploring how to improve the performance of BioKeySpotter with abstract.

**Acknowledgements.** Partial support for this research was provided by the National Science Foundation under grants DUE- 0937629, by the Institute for Museum and Library Services under grant LG-02-04-0002-04, and by the New Jersey Institute of Technology.

## References

Arshadi, N., Jurisica, I.: Feature Selection for Improving Case Based Classifiers on High Dimentional Data Sets. AAAI, Menlo Park (2005)

Barker, K., Cornacchia, N.: Using Noun Phrase Heads to Extract Document Keyphrases. In: Hamilton, H.J. (ed.) Canadian AI 2000. LNCS (LNAI), vol. 1822, pp. 40–52. Springer, Heidelberg (2000)

Bracewell, D.B., Ren, F., et al.: Multilingual single document keyword extraction for information retrieval. In: NLP-KE 2005. IEEE, Los Alamitos (2005)

Brown, J.: Growing up digital: How the web changes work, education, and the ways people learn. Change, 10–20 (2000)

D'Avanzo, E., Magnini, B., et al.: Keyphrase Extraction for Summarization Purposes: The LAKE System at DUC-2004. In: Document Understanding Workshop. HLT/NAACL, Boston, USA (2004)

El-Beltagy, S.: KP-Miner: A Simple System for Effective Keyphrase Extraction. In: Innovation in Information Technology. IEEE Xplore (2006)

Frantzi, K.T., Ananiadou, S., Tsujii, J.: The $C-value/NC-value$ Method of Automatic Recognition for Multi-word Terms. In: Nikolaou, C., Stephanidis, C. (eds.) ECDL 1998. LNCS, vol. 1513, pp. 585–604. Springer, Heidelberg (1998)

Liu, X.: Intelligent Data Analysis. Intelligent Information Technologies: Concepts, Methodologies, Tools and Applications, 308 (2007)

Settles, B.: Biomedical Named Entity Recognition Using Conditional Random Fields and Rich Feature Sets. In: Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and Its Applications (NLPBA), pp. 104–107 (2004)

Mihalcea, R., Tarau, P.: TextRank: Bringing Order into Texts. In: EMNLP-2004, Barcelona, Spain (2004)

Song, M., Song, I.-Y., et al.: KPSpotter: a flexible information gain-based keyphrase extraction system. In: International Workshop on Web Information and Data Management ACM (2003)

Song, M., Rudniy, A.: Markov Random Field-based Edit Distance for Entity Matching, Biomedical Literature. In: International Conference on Bioinformatics and Biomedicine, pp. 457–460 (2008)

Turney, P.D.: Extraction of Keyphrases from Text: Evaluation of Four Algorithms, pp. 1–29. National Research Council Canada, Institute for Information Technology (1997)

Turney, P.D.: Learning Algorithms for Keyphrase Extraction. Information Retrieval 2(4), 303–336 (2000)

Wan, X., Yang, J., et al.: Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. ACL, Prague (2007)

Witten, I.H., Paynter, G.W., et al.: KEA: Practical Automatic Keyphrase Extraction. In: The Fourth on Digital Libraries 1999. ACM CNF, New York (1999)

Zhang, Y., Zincir-Heywood, N., et al.: Narrative text classification for automatic key phrase extraction in web document corpora. In: 7th Annual ACM International Workshop on Web Information and Data Management. ACM SIGIR, Bremen (2005)

# Chapter 4
# Mining Health Claims Data for Assessing Patient Risk

Ian Duncan

Visiting Assoc. Professor,
Dept. of Statistics & Applied Probability,
University of California Santa Barbara,
Santa Barbara, CA 93106

**Abstract.** As all countries struggle with rising medical costs and increased demand for services, there is enormous need and opportunity for mining claims and encounter data to predict risk. This chapter discusses the important topic, to health systems and other payers, of the identification and modeling of health risk. We begin with a definition of health risk that focuses on the frequency and severity of the events that cause patients to use healthcare services. The distribution of risk among members of a population is highly skewed, with a few members using disproportionate amounts of resources, and the large majority using more moderate resources. An important modeling challenge to health analysts and actuaries is the prediction of those members of the population whose experience will place them in the tail of the distribution with low frequency but high severity. Actuaries have traditionally modeled risk using age and sex, and other factors (such as geography and employer industry) to predict resource use. We review typical actuarial models and then evaluate the potential for increasing the relevance and accuracy of risk prediction using medical condition-based models. We discuss the types of data frequently available to analysts in health systems which generate medical and drug claims, and their interpretation. We also develop a simple grouper model to illustrate the principle of "grouping" of diagnosis codes for analysis. We examine in more depth the process of developing algorithms to identify the medical condition(s) present in a population as the basis for predicting risk, and conclude with a discussion of some of the commercially-available grouper models used for this purpose in the U.S. and other countries.

## 1 What Is Health Risk?

Health risk can take different forms, depending on one's perspective. Systems in which care is insured are greatly concerned with health risk because of the potential that it represents for claims to exceed available resources. Even single payer systems in which care is centrally directed have begun to realize the importance of health risk assessment because of what it can teach planners about opportunities to provide more efficient care. Clinicians are concerned with *clinical* risk, or the deterioration of body systems due to the progress of disease or injury (or treatment of disease). Although we will touch on clinical risk in this chapter, our focus is largely on the identification and management of financial risk.

D.E. Holmes, L.C. Jain (Eds.): Data Mining: Found. & Intell. Paradigms, ISRL 25, pp. 29–62.
springerlink.com 

At its most fundamental, risk is a combination of two factors: **loss** and **probability**. We define a loss as having occurred when an individual's post-occurrence state is less favorable than the pre-occurrence state. Financial Risk is a function of Loss Amount and Probability. Symbolically,

**Risk = *F*(Loss Amount; Probability)**

The measurement of risk requires the estimation and quantification of losses and the probability of their occurrence. Health risk management requires the identification of ways that either the amount of losses or the probability of their occurrence may be mitigated. These components of risk are often referred to as "Severity" (amount of loss), and "Frequency" (probability of a loss).

Some risks may be assessed simply: the loss resulting from being hit by an automobile while crossing a street is rather high and encompasses medical expenses, loss of future function and loss of income; the probability of such an event occurring is, however, slight, so the overall financial risk is low. In addition, the risk may be mitigated almost entirely by crossing at a designated intersection and obeying the crossing signal.

Health risk is a more complicated concept because it exists on many levels in many contexts. In the United States, in part because health care is financed through an *insurance* system, we are particularly focused on financial risk, or the likelihood that a payer will incur costs as a result of the clinical risks (and their treatments) to which to the insured life is subject. In the U.S. we are concerned with concepts that do not occur in other systems, such as Pricing Risk and Underwriting Risk. We may loosely distinguish between underwriting risk and pricing risk by thinking of the former as resulting from the cost of unknown risks while the latter is more related to the cost of known risks. Understanding of these concepts is useful irrespective of the system, because the forces that give rise to risk are at work universally, irrespective of the way healthcare is financed.

1. The most obvious health risk confronted by a payer[1] is **pricing risk**. A health plan accepts pricing risk by agreeing to reimburse health-related services (limited only by medical necessity and any other restrictions defined in the insurance contract) in exchange for a fixed monthly premium. This pricing risk can take one (or both) of two forms that coincide with the two risk concepts defined earlier – loss amount and probability. A health insurer's actual experience can be equal to the expected *number* of claims, but at the same time include more large claims (severity). Alternatively, individual claims may be of modest size, but a payer could simply experience a higher-than-expected *volume* of them (frequency).
2. Closely linked to, but not identical to, pricing risk is **underwriting risk**. In setting a price for an insurance product, a health plan is expecting that the overall risk pool will perform, on average, at the estimated total claim level. Some participants in the pool will cost more and others less than the projected average. Sound financial management of an insurance pool requires underwriting standards to be set at such a level that the actual distribution of member-risks in the pool approximates the distribution of members expected in the pricing. Too

[1] In the U.S. a payer could be an employer, a health plan, an individual or a government.

many high-risk members or too few low-risk members will result in the overall claims exceeding those expected in pricing. Controls on access could include a requirement that the potential member demonstrate good health at the time of application, be subject to exclusion of claims for pre-existing conditions, be subject to a limitation on high cost procedures, or be required to seek care from a network of physicians who are known to practice conservatively or efficiently. The underwriting process identifies members who could potentially have high claims; those with risk factors such as a high cost condition, family history of certain illnesses, or potentially risky lifestyles (such as smoking). Depending on the type of insurance being offered, higher-risk members may pay a higher premium, be excluded completely from insurance or be subject to exclusion of known conditions likely to generate high costs (a process known as "lasering" or pre-existing condition exclusion) The management of large claims may include reinsurance of high amounts and care management programs targeting certain high-risk conditions. Some states prohibit underwriting, (and the concept is of course foreign to countries with a single payer, universal coverage system) exclusion of pre-existing conditions or variations in pricing; New York, for example, requires community rating and prohibits underwriting.

If, as we discussed, potentially high claiming individuals drive underwriting risk, what risk factors imply that an individual may generate high claims? This is a crucial risk management issue for healthcare, with numerous implications, including:

1. Surplus Requirements: how much free capital does a health plan need to hold in order to withstand extreme losses due to claim frequency or severity (or both)? In the United States, the amount of free capital (surplus) is regulated. While a health plan may perform its own analysis of free capital requirements, it is ultimately bound by regulatory requirements. Whether capital is regulated or subject to actuarial modeling, it is important to understand frequency and severity of claims in order to manage capital requirements for an insurer to remain a going concern.
2. Care Management: who are the potential high utilizers of health care resources? Can these individuals be identified prior to their episodes of high utilization in order to engage them in a process of care management that will moderate their cost?
3. Pricing and underwriting: can utilization and cost of members be predicted with sufficient accuracy over the rating period that this information may be incorporated into a plan's pricing methodology?

These three examples of the use of risk factor identification and prediction indicate the potential value of data analysis and risk prediction to a health payer.
We can, for the purpose of this discussion, categorize risk factors as follows:

1. Inherent risk factors such as age, sex, or race. These are factors that are immutable, as compared (for example) to geographic risk, over which the individual has some control. Some readers may find it difficult to accept these characteristics as “risk factors.” Objectively, however, they are associated with

differential risk levels (losses and probabilities of losses). Health plan actuaries can and do assess these differential risks; it is for the market and the regulators to determine (as they do) which risk factors may be applied in practice to price or underwrite products. As the Society of Actuaries studies have shown[2], age, sex and prior cost are associated with future claim levels.

2. Medical condition-related risk factors such as diabetes or cancer. Individuals with these types of conditions clearly will generate higher claims than members who do not have medical conditions.
3. Family history: some risk factors and medical conditions, such as hemophilia or certain cancers are inheritable. Information about family history can therefore be predictive in certain circumstances.
4. Lifestyle related risk factors, such as smoking, stress, seat-belt use, lack of exercise and poor nutrition contribute to higher cost. Some of these factors will have a short-term impact on member cost; other factors, such as obesity or smoking, will take years to have an effect on member health, leading to the emergence of medical conditions later.
5. Exogenous risk factors, such as: the industry in which an individual works, the location of his home, his education level, and cultural or religious beliefs.

In addition to the risks discussed above which operate at the individual level, a payer must be concerned with population risk. This will be different to that of an individual, because of the "spread of risks" that is inherent when a number of lives are pooled in a population. An individual may be highly risky because of his condition-based risk or lifestyle risk factors, yet the population of which he is part may not represent an elevated risk. Conversely, in a small population, a single catastrophic event could result in a large loss to the population. Furthermore, it is not only catastrophic events that represent significant population risk: a relatively small increase in the frequency or cost of some basic (low-cost) services delivered to the larger population can result in the overall losses in the population being much larger than anticipated by the pricing. Population risk is not only a function of large claims amounts or higher frequency of claims. The behavior of low claiming members of the population, whose participation is required to support the cost of the population, also affects the experience of the pool, which could be adversely affected if the participation of low-claiming members does not materialize in sufficient numbers. Discussion of population risk often focuses on the small percentage of high claiming individuals, but should not overlook the anti-selection risk posed by the participation of insufficient low-risk members of a population, or their withdrawal from participation as rates increase. Stability of the pool is dependent on continued funding by low-risk members and the withdrawal of these members poses a risk to the financial stability of the pool[3].

In the U.S., financial risk tends to be evaluated in terms of monthly or annual time periods, consistent with the way healthcare is financed. Clinicians, however, tend to think in terms of events (admissions to the hospital; visits to an outpatient facility; loss of physical or mental function) rather than financial losses. So we should recognize that risk may mean different things to different audiences.

---

[2] Society of Actuaries risk adjuster studies, for example Dunn et all[1]; Cumming et al, [2]; Winkelman and Mehmud, [3].

[3] The withdrawal of low-risk members from a pool following rate increases is known as an "assessment spiral" because the termination of these members often forces increases in rates, leading to further withdrawals.

All of these risks have a place in the concerns of those managing an insurance company, pool or healthcare budget. At its heart, health risk is a combination of the amount and probability of a loss. Management of an insurer's risk requires that both the frequency and the severity of claims be predicted as accurately as possible, and that techniques to manage both of these factors be applied.

There are two major applications, related but different, of data mining in health care data:

1. Predictive modeling is a technique that helps companies more accurately predict both frequency *and* severity of claims. Predictive modeling is most frequently used for high-risk member case finding, for example to find candidates for a care management program. It is also used to predict member cost and utilization for underwriting and pricing purposes, or for program evaluation when a predicted cost level is required for comparison with an actual level of cost.
2. Risk Adjustment is one of many techniques used to normalize populations for comparative purposes, for example to classify members by potential risk level for the purpose of provider reimbursement, (e.g. the DRG system in the U.S.) or for payment to a provider or insurer (e.g. the systems of payments used to reimburse insurers in the Netherlands and in the U.S. Medicare Advantage system).[4]

## 2 Traditional Models for Assessing Health Risk

Actuaries have been using models for many years to predict health risk and cost. Traditional models, based on age, sex and prior cost have been supplanted in recent years by models that incorporate additional knowledge about individual health conditions. In this section we shall examine actuarial techniques for assessing health risk.

The treatment here is theoretical, to illustrate the principle. In the United States, regulators in many states prohibit the application of many of these techniques without further adjustment.

Figure 1 shows a typical claims frequency distribution. Claims in this case are "allowed" amounts[5], that is, the amount of claims recognized by a health plan for reimbursement, but before the cost-sharing imposed by a plan design between insured and payer. The mean allowed amount per member per year (PMPY) is $4,459. Cost-sharing typically reduces this amount by 15% to 20%. The distribution of allowed amount is highly-skewed, with the highest percentage of members having relatively low claims and a small percentage having very high claims, often in the hundreds of thousands or millions of dollars per member. The skewed distribution illustrates some of the risk management techniques employed by payers: a payer wants to maximize the number of low-claiming members while simultaneously identifying and managing the costs of those who are in the tail of the distribution.

[4] Other methods exist to classify members into risk levels, for example case-mix adjustment. The advantage of predictive modeling methods is that they impose a common numerical system of risk on a heterogeneous system of risks.

[5] "Allowed charges" refers to the amount of a claim after the deduction of any applicable provider discounts but before the deduction of cost-sharing (co-pays, deductibles, etc.). We use allowed charges as the basis of comparison because they are not confounded by different cost-sharing arrangements.

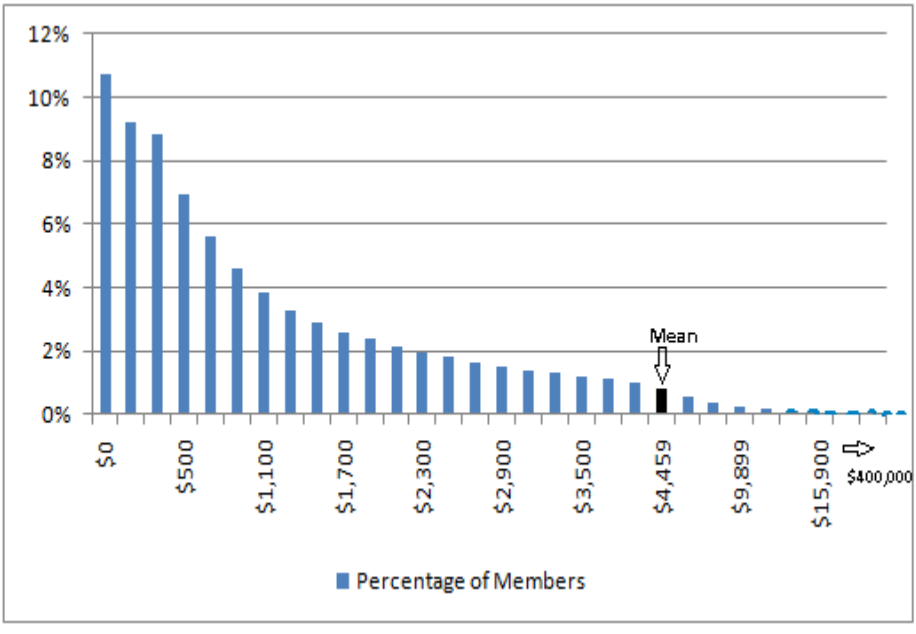



**Fig. 1.** Distribution of Members by Claim Amount

Traditionally, actuaries have predicted the likely future cost of an individual or of the group of which the individual is a member using one (or a combination) of the following techniques:

1. **Age/Sex:** although *individuals* of the same age and sex represent a range of risk profiles and costs, *groups* of individuals of the same age and sex categories follow more predictable patterns of cost. Table 1 takes the member data underlying Figure 1 and shows the distribution of relative costs (allowed charges) of different age and sex categories.

**Table 1.**

| Relative Cost PMPY by Age/Sex | | | |
|---|---|---|---|
| | Male | Female | Total |
| < 19 | $2,064 | $1,951 | $2,008 |
| 20-29 | $1,894 | $3,949 | $2,921 |
| 30-39 | $2,509 | $4,863 | $3,686 |
| 40-49 | $3,679 | $5,259 | $4,469 |
| 50-59 | $6,309 | $6,994 | $6,652 |
| 60-64 | $9,266 | $9,166 | $9,216 |
| Total | $3,978 | $4,940 | $4,459 |

For underwriting and pricing purposes, such relative claims are often converted into factors that may be applied to a particular population's demographics to derive its overall risk "score." Assuming that the insurer's entire book of business has a score of 1.00, a rate may then be established for a specific group (and individuals within a group) based on the group's risk relative to that of the book-of-business of which it will be a member. This is illustrated in Table 2. Male and Female factors may be used where this is permitted; when "unisex" factors are required, the "Total" risk factor column applies, based on age only.

As these factors illustrate, younger males and females cost about half the average for the group, while older males and females cost about twice the average. Men generally represent a lower-cost group in the age range 20 to 60.

**Table 2.**

| Relative Age/Sex Risk Factors | | | |
|---|---|---|---|
| | Male | Female | Total |
| < 19 | 0.47 | 0.44 | 0.45 |
| 20-29 | 0.41 | 0.91 | 0.66 |
| 30-39 | 0.57 | 1.09 | 0.83 |
| 40-49 | 0.83 | 1.19 | 1.01 |
| 50-59 | 1.42 | 1.58 | 1.50 |
| 60-64 | 2.09 | 2.07 | 2.08 |
| Total | 0.88 | 1.12 | 1.00 |

Age/sex factors are frequently applied to groups to develop a manual rate. An example of such a calculation is provided in Table 3.

**Table 3.**

| | Male Risk Factor | Male Number | Female Risk Factor | Female Number | Weighted Number |
|---|---|---|---|---|---|
| < 19 | 0.47 | 4 | 0.44 | 12 | 14.3 |
| 20-29 | 0.41 | 12 | 0.91 | 19 | 24.9 |
| 30-39 | 0.57 | 24 | 1.09 | 21 | 35.7 |
| 40-49 | 0.83 | 30 | 1.19 | 24 | 50.0 |
| 50-59 | 1.42 | 10 | 1.58 | 10 | 25.8 |
| 60-64 | 2.09 | 3 | 2.07 | 1 | 9.4 |
| **Total** | 0.88 | 88 | 1.12 | 89 | 160.1 |
| Relative age/sex factor | | | | | 0.94 |

Assuming that our underlying manual (book-of-business) claims are expected to be $4,459 per year, this group is predicted to have costs 0.94 * $4,459 or $4,191[6]. This simple method is universally applied, but its principal drawback is that it ignores information that may exist about the population's *clinical* risk burden and the likely future development of costs from those risks, and is thus not as accurate as methods that consider clinical risk. Age/sex is a poor technique in some cases, but better in others. For relatively large groups and groups where the underlying demographics are similar to those of the book of business (age/sex factors closer to 1.00), the estimate is reasonably close. Where the group is smaller or significantly old age/sex factors are a poor estimator of future cost. Since older members are more likely to develop health conditions, the divergence at older ages is not unexpected.

2. **Prior Cost** (prior year's claims)**:** prior cost is one the most frequently used risk predictors for pricing and underwriting for groups, and is also often used for selecting candidates for care management programs. The Prior high cost is not a particularly accurate predictor of future high cost at the individual level (see Duncan (4) Chapter 1). In a large population, only one-third of previously high cost members remain high cost from one year to the next; about one-fifth of all high cost members in a subsequent year were in the low cost group in the prior year. However, this observation of *individual* migration between cost statuses is mitigated at the group level because, particularly in larger groups, member costs are constantly changing and tend to offset each other. For smaller groups, variations are less likely to be offsetting and can result in significant deviations between actual and predicted costs, leading to the use of credibility-weighting[7] approaches for smaller employer pricing.

3. **Combination of age/sex and prior cost:** particularly for rating smaller groups, a combination of prior cost and age/sex rating is often used, with the proportions of each in the final calculation being driven by the credibility assigned by the underwriter to the size of the group (and sometimes to the validity of its data). The combination of both approaches produces more accurate predictions than either method used separately. A frequently-applied method for performing this calculation applies credibility as follows:

[6] These numbers may seem high to some readers. However, they represent 2011 costs, and are allowed charges, that is, before the effect of member deductibles and co-payments.

[7] Credibility-weighting is a standard actuarial technique that allows actual experience of a group to be taken into account in developing rates, while recognizing that a smaller group's experience is more subject to random fluctuations than a large group's experience. See Bluhm, Chapter 35 for more detail.

$$\text{Expected Cost} = \text{Prior Year Cost} \times \text{Trend} \times \text{Z} + \text{Book of Business Cost} \times (1 - \text{Z})$$

where $Z = \left(\frac{N}{2000}\right)^{0.5}$ and $N$ is the number of members in the group.

## 3 Risk Factor-Based Risk Models

The traditional models discussed in Section 2 have in common that they predict health cost and risk using limited information about individual member risk factors (demographic only). Age is clearly a proxy for medical condition-based risk, but as the Society of Actuaries studies show, not a particularly accurate one, even when member risks are grouped ($R^2$ rarely exceeds 0.1). However, it may be possible to make more accurate predictions if we incorporate additional risk factors into our modeling.

Typical predictive modeling techniques rely on incurred claims. The detail contained in claims makes the risk assessment and predictive modeling based on this data reasonably reliable. We will discuss risk factors that can be derived from three different sources of data:

1. Medical condition-related risk factors derivable from claims, such as diabetes or cancer. Members with chronic medical conditions clearly will generate higher claims than members who do not have those medical conditions, and the fact that these conditions persist makes them particularly useful for risk prediction.
2. Lifestyle related risk factors derivable from self-reported data, such as smoking, stress, lack of exercise, poor nutrition, poor seat-belt use, scuba diving, auto racing etc. Some of these risk factors may have a short-term impact on member cost such as an auto accident; other factors, such as obesity or smoking, may take years to have a pronounced effect on member health, generally leading to the emergence of chronic medical conditions.
3. Lifestyle-related risk factors that are derivable from external or exogenous risk factors, such as the industry in which an individual works, the location of his home, his education level, household income level, type of insurance coverage, etc.

The remainder of this section discusses the underlying theory of risk factors and their incorporation into predictive models.

Different medical conditions impose different healthcare risks and costs. We illustrate this principle in Table 4 with some sample lives from the Solucia Consulting test database[8], a national database of payer data. Standardized costs are those for an average individual of the same age and sex and year of payment as the sample lives.

The examples in this table illustrate a number of points about medical condition-based risk. First, we see clearly that the existence or absence of a health condition is correlated with deviation from the standardized (average) member cost in the last column. Members at any age that have no conditions cost less than the standardized cost for the age/sex group. Members with chronic or acute conditions cost considerably more.

[8] Solucia Consulting: www.soluciaconsulting.com

**Table 4.**

| Condition Based vs. Standardized Costs | | | | | | |
|---|---|---|---|---|---|---|
| Member | Age | Sex | Condition | Actual Cost (annual) | Standardized Cost (age/sex) | Condition cost/standardized cost |
| 1 | 25 | M | None | $1,247 | $1,894 | 66% |
| 2 | 55 | F | None | $4,137 | $6,994 | 59% |
| 3 | 45 | M | Diabetes | $7,257 | $3,679 | 197% |
| 4 | 55 | F | Diabetes | $10,098 | $6,994 | 144% |
| 5 | 40 | M | Diabetes and Heart conditions | $33,914 | $3,679 | 922% |
| 6 | 40 | M | Heart condition | $26,267 | $3,679 | 714% |
| 7 | 40 | F | Breast Cancer and other conditions | $38,605 | $4,863 | 794% |
| 8 | 60 | F | Breast Cancer and other conditions | $23,016 | $9,166 | 251% |
| 9 | 50 | M | Lung Cancer and other conditions | $60,243 | $6,309 | 955% |

This example also illustrates the principle of additivity:

- The 40 year old male whose only health risk is a heart condition experiences a cost of $26,267.
- The 45 year old male whose only health risk is a diagnosis of diabetes experiences a cost of $7,257.
- The 40 year old male with diagnoses of both a heart condition and diabetes experiences a cost of $33,914.

In this particular example, the effect of heart condition and diabetes diagnoses is additive. Experience shows, however, that some co-morbidities could also result in cost efficiencies (total cost being less than the sum of individual condition costs) while other co-morbidities could result in cost-reinforcement (total cost being greater than the sum of individual costs).

Finally, there is considerable variance in the cost of the same condition at different ages. In the case of the members with diabetes-only diagnoses, the older member has a higher cost than the younger member. In the example of diagnoses of breast cancer and other conditions, the older member's cost is lower. This variance could be due to a number of factors, including difference in severity of the diagnosis, treatment of the patient, or non-specific "other conditions."

One note of caution about the numbers in Table 4. The members in this table have been selected to illustrate the relative costs of members with certain conditions. For some of these members the costs relative to the standardized (age/sex) costs are high.

The relative frequency of these high cost members is low. We still see the principle of insurance at work, however, as the premiums for the low-cost members (25 year old male and 55 year old female) subsidize the claims of the other members.

As the above example shows, the member's diagnosis contains considerable information that is potentially useful for building more accurate models to predict an individual's or group's cost.

In a later section we will discuss ways in which to derive information from claims data for risk assessment. But first, we will discuss the types of data available to the analyst.

## 4 Data Sources

This section addresses data sources available to the analyst interested in predictive modeling or risk adjustment. There are many different types of data, with different uses and degrees of validity.

The discussion in this section is highly U.S.-centric. Readers from other countries may want to skim this section and consider equivalent data sources, if any, in their own countries.

Identification of medical conditions is the basis of all prediction and risk adjustment. The major source of raw material for identification is data, primarily claims. Availability of data is a central issue. Without data, there is no identification, targeting, or prediction. Unfortunately, there is no ideal source of data. Each source has its benefits and drawbacks, which must be weighed against each other. Predictive modeling, as any other data-driven endeavor, is subject to the "garbage in garbage out" principle. Keep in mind, the more data obtained from care provided in as many different settings as possible, the better a picture of overall healthcare services utilization can be assembled. Knowing your data, its limitations, and its potential is therefore crucial.

At least six types of data are commonly available to the health care analyst. We summarize these data, with a subjective evaluation (based on years of experience with different types of data) of their advantages and disadvantages, in the following table.

**Table 5.**

| Data Source | Reliability | Practicality |
|---|---|---|
| Enrollment | High | High |
| Claims | Medium | High |
| Pharmacy | Medium | High |
| Physician Referral/chart | High | Low |
| Laboratory Values | High | Low |
| Self-reported | Low/medium | Low |

The practicality of physician referrals, chart information, laboratory values, and self-reported data, while currently low, continues to improve.

### 4.1 Enrollment Data

Many key financial, clinical and operational statistics are expressed on a per member, per month or per 1,000 members basis. Health plan enrollment (often called "eligibility") information can be a valuable resource for denominator/population information to include in claims analysis. Enrollment Data allows plan participants to be identified, counted, and categorized, regardless of whether they had a healthcare service during the study period. Many health plans utilize the 834 Benefit Enrollment and Maintenance Form, a set of standards for HIPAA[9]-compliant electronic transfer of client enrollment and disenrollment information. By promoting industry standards for the transfer and storage of plan membership data, the 834 format has resulted in increased availability and more accurate historical enrollment data. Standardized fields for data transmission include coverage date spans (starting and ending), provider/plan information, and limited member demographics. Although a large number of plans use the 834 format, many of them modify the data elements to fit their particular data definitions, thereby lessening (but not eliminating) the reliability of this tool in standardized reporting.

Enrollment data can sometimes specify which services are covered. High-level benefit design information is particularly useful for understanding which members have prescription drug, vision, or behavioral health benefits. More detailed benefit information such as deductible, coinsurance, and copayment levels is also useful for understanding financial barriers to utilization, as numerous studies have demonstrated an inverse relationship between members' cost-sharing and service utilization. Information about benefit design is not generally as readily available, or in as easily interpreted form as enrollment information. If required, it may be inferred, with a certain amount of effort, from claims data, by analyzing differences between allowed and paid claims for different classes of service.

### 4.2 Claims and Coding Systems

Claims data contains a wealth of useful information for the analyst and modeler. In order to obtain reimbursement for any service in the United States, the provider of that service must submit a claim in one of several formats, depending on the type of service. Examples of claims forms are UB 04 for hospital claims (UB is an abbreviation for "universal benefit") and CMS 1500 for professional claims (CMS is the abbreviation for Centers for Medicare and Medicaid Services)[10]. Claims forms

[9] A very significant piece of legislation regulating healthcare data is the Health Insurance Portability and Accountability Act. It was signed into law by President Clinton in 1996 with full implementation required by early in 2003. There are three parts of HIPAA that have the greatest impact. These are the EDI Rule (42 CFR 162.1000), which defines the classification systems that the Health Care organizations should use, the Security Rule (42 CFR 164.306) which requires the Health Care organizations to create and implement security safeguards and The Privacy Rule (42 CFR 164.502) which requires protection of the data.

[10] See Duncan (2011) for examples of these forms, together with a discussion of the data therein.

contain detailed information about the patient and the provider, together with descriptions of the services rendered, each with its own coding scheme. Below, we discuss some of the major types of information that may be derived from claims.

Medical claims data have high ratings for availability. They are generally available in a health plan environment, except when **capitation**[11]agreements are in place. Claims data are often criticized (at least as compared with data from medical records) for the depth and accuracy of the medical information they contain, but because providers have an interest in submitting accurate information for reimbursement, medical claims data quality is relatively high. Key data elements contained in medical claims include the diagnosis, the procedure, the provider, the place and type of the service, date of service, date of adjudication, amount charged and amount paid. The quality of such data, however, varies greatly between health plans and there are many additional possibilities for error in the data-gathering, submission, warehousing and transmission stages. Data quality diagnostic reports should therefore be run to determine data quality before beginning any analysis, and control totals should be produced to ensure that data is being read accurately, and that successive data feeds are consistent in terms of volume and amount with prior feeds. Examples of the type of reports that are often run including the traditional actuarial completion[12] (claims triangle) report, as well as reports on key trends such as number of claims, services and prescriptions per member per month, and their associated costs.

Possibly the most useful information contained in claims data is the member's diagnosis and the procedure(s) (services) provided. The **International Statistical Classification of Diseases and Related Health Problems** (**ICD**) known as ICD codes, provides a framework for categorizing diseases and other conditions on a very detailed level. ICD codes classify diseases and a wide variety of signs, symptoms, abnormal findings, complaints, social circumstances, and external causes of injury or disease. Every health condition can be assigned to a unique category and is given a code, up to five digits long. The International Classification of Diseases was first published by the World Health Organization (WHO) in 1948. The original intent of ICD codes was to report mortality statistics only. The WHO published the ICD-8 codes in 1968 and ICD-9 codes in 1977. Although ICD-10 codes were published in 1994, official adoption for *clinical* coding purposes[13] of the ICD-10-CM (Clinical Modification) system in the Unites States has not yet occurred as of this writing. Adoption in the U.S. is currently projected by October 1, 2013. Although not yet

[11] Capitation arrangements are those arrangements in which providers accept the risk of service volumes in return for a fixed ("capitated") payment per health plan member. Under such an arrangement the capitated provider is responsible for providing all eligible and medically-necessary services to the covered member. (A stop-loss arrangement is sometimes negotiated with the payer to protect the provider against catastrophic events.)

[12] Claims Triangle: an incurred and paid claims triangle is a useful method for displaying claims data for a number of purposes, including data validation and reserving. The triangle arranges data by date of claims incurral (service) and payment. It is then possible to evaluate whether there is any change, month to month, in the reporting of initial and subsequent claims relative to historical patterns, as well as to project the amount of future claims that may be expected to be reported for any month of incurral.

[13] The ICD-10 codes were adopted for mortality reporting and have been used for this purpose in the U.S. since 1999.

used in the U.S., the 10th edition (ICD-10) is used broadly throughout many countries. An 11th edition is in preparation.

The ICD-10 coding system introduces several changes compared with the previous ICD-9 system. Although ICD-10 will not be implemented in the US before 2013 at the earliest, the changes inherent in this classification system (compared with the previous edition) will have significant implications for all predictive models and risk adjusters. For this reason, we note some of the most significant changes:

1. Almost twice the number of categories as ICD-9.
2. Use of alphanumeric categories instead of numeric only (except for E and V).
3. Changes in chapters, categories, titles, and regrouped conditions.
4. Expansion of injury codes, ambulatory and managed care encounters.
5. Addition of a 6th character.

Medical conditions are identified from diagnosis codes contained in member claims data. Because there are in excess of 20,000 diagnosis codes (within the ICD-9 code system), some form of grouping is usually essential for analysis. Grouping may also be performed without a commercial or internally-developed grouper. The ICD system generally classifies diagnosis by parts of the body. One example of a grouping that follows this approach is the Major Diagnostic Category, as shown in Table 6.

While the MDC level of grouping is sometimes useful for reporting purposes, the amount of variation in resource consumption within classes makes it too broad a classification for predictive or risk adjustment purposes.

In addition to codes within the series above, ICD-9 codes sometimes begin with a letter, either a V (which replaces the first digit of the three-digit number) or an E code (which consists of the letter E plus a 3-digit number). V codes indicate ongoing treatment not related to a particular condition (immunizations for example). E codes designate accident, injury, or poison-related diagnoses.

The number before the decimal point indicates the diagnostic category; the fourth digit (first position after the decimal point) provides further diagnostic information (such as body part or complication). The fifth digit can also denote further information, such as whether an episode of care is initial, subsequent, or unspecified, or the severity of the condition. For example, diabetes-related diseases are part of the classification group 240-279 (Endocrine, nutritional and metabolic disorders). Code 250 is Diabetes Mellitus; a fourth digit represents various complications of diabetes, from Ketoacidosis (250.1) through 250.9 (diabetes with unspecified complications). A fifth digit is added, specific to Diabetes, indicating the type of diabetes and degree to which it is controlled.

| | |
|---|---|
| Type II, not stated as uncontrolled: | 0 |
| Type I, not stated as uncontrolled: | 1 |
| Type II, uncontrolled: | 2 |
| Type I, uncontrolled: | 3 |

**Table 6.**

| Major Diagnostic Categories | |
|---|---|
| 001-139 | Infectious and parasitic diseases |
| 140-239 | Neoplasms |
| 240-279 | Endocrine, nutritional and metabolic diseases, and immunity disorders |
| 280-289 | Diseases of the blood and blood-forming organs |
| 290-319 | Mental disorders |
| 320-359 | Diseases of the nervous system |
| 360-389 | Diseases of the sense organs |
| 390-459 | Diseases of the circulatory system |
| 460-519 | Diseases of the respiratory system |
| 520-579 | Diseases of the digestive system |
| 580-629 | Diseases of the genitourinary system |
| 630-676 | Complications of pregnancy, childbirth, and the puerperium |
| 680-709 | Diseases of the skin and subcutaneous tissue |
| 710-739 | Diseases of the musculoskeletal system and connective tissue |
| 740-759 | Congenital anomalies |
| 760-779 | Certain conditions originating in the perinatal period |
| 780-799 | Symptoms, signs, and ill-defined conditions |
| 800-999 | Injury and poisoning |

These complications and their associated costs are discussed in more detail below.

**Diagnosis codes** are nearly always present on medical claims data. Sometimes there may be multiple diagnosis codes, although usually only the primary and secondary codes are populated with any regularity or reliability. Diagnosis codes nearly always follow the uniform ICD formats. Unfortunately, coding errors are sometimes present on a claim. First, claims are usually not coded by the diagnosing physician, or they may be coded vaguely – physicians may have a strong conjecture on a precise diagnosis but will often code a diagnosis in the absence of confirming tests because a diagnosis is required for claim reimbursement. This is often the case for rare diagnoses as well as for emergency treatments where precision is less important than swift treatment. Diagnosis codes may sometimes be selected to drive maximum reimbursement (a process also known as "upcoding"). Finally, coding may lack uniformity – different physicians may follow different coding practices. For all of their drawbacks, however, diagnosis codes are invaluable due to their availability, uniformity of format, and usefulness.

**Procedure Codes:** A procedure is a medical service rendered by a health care professional in a clinical, office, hospital or other setting, where the professional provides and bills for the service. These bills include procedure codes, CPT4 (“Current Procedural Terminology, Version 4”) codes, ICD Procedure codes or

HCPCS (Healthcare Common Procedure) codes, which are almost always found in claims data. Unlike diagnosis codes, however, procedure codes come in various formats, some of which are specific to particular health plans. Some of the most common formats will be discussed later in this chapter. Like diagnosis codes, procedure codes are useful when available, and often provide valuable insight into the actual services being performed. For this reason they are particularly useful in the case of physician services and when assessing provider performance with respect to efficiency and quality.

ICD procedure codes are found on the header of UB 04 facility claims ("UB" is an abbreviation for Universal Billing) and are generally less specific than CPT and HCPCS procedure codes. Facility claims may contain more specific CPT and HCPCS codes on the claim detail. Professional claims do not contain ICD procedure codes but should always contain CPT and/or HCPCS codes.

Service location and type are generally present on medical claims and can be used to determine basic types of service, for example emergency, hospital inpatient, etc. Formats are sometimes standard and sometimes health plan-specific. Data quality is generally good enough to use in modeling.

**Claims Costs:** Cost of services is both a highly necessary data element, and highly problematic. It is a necessary factor because our definition of risk is largely financial. It is problematic due to a lack of uniformity. First, there is always debate about whether it is appropriate to use billed, allowed, or reimbursed cost. Billed cost is more subject to non-uniformity and can be highly inflated (because it is generally subject to negotiated provider discounts). Allowed charges represent billed charges less disallowed costs and non-covered charges, (such as the cost of a television set in a hospital room) and contractual discounts. Reimbursed cost (allowed charges) is altered by deductibles and other cost-sharing, and is usually referred to as "paid claims" or "net paid claims." Both allowed charges and net paid claims are subject to variation in physician coding practices as well as regional differences in customary reimbursement rates. In discussing "Claims" or associated analysis (such as predictive modeling) it is important to be clear at which level in the claim process the "claims" under consideration have been extracted.

The date of service and dates of **adjudication** (evaluation of the submitted claims for reimbursement) or of any adjusting entries are essential to medical claims. Date of service is a primary need, because it defines whether a claim will be reimbursed, as well as the timing of a diagnosis or service. The date of adjudication is essential for verification of completeness of data. There is a lag (sometimes referred to as **"run-out"**) between dates of service and adjudication that can be several months or more. This lag is illustrated in Table 7, which shows a typical health claims run out triangle. The data are arranged by month of incurral and month of payment, resulting in increasing frequency of blank cells as we move through the year.

Completeness of paid claims data maybe evaluated using different methods such as the chain-ladder or the inventory method (using a report of claims costs cross-tabbed between months of service and adjudication). Time-based reports change as data is refreshed and missing claims are added. A rule-of-thumb often used historically was that a minimum of 90 days of claims run-out (number of days between the last date of service being considered and the latest available adjudication date) was sufficient for a claims data set to be considered "complete." More recently, claims lags have shortened and claims are likely to be more complete after 60 or 90 days than in past years.

**Table 7.** Example of a Claims Triangle

| Month of Payment | Month of Incurral: January | February | March | April | May | June |
|---|---|---|---|---|---|---|
| January | 10.0 | | | | | |
| February | 175.0 | 12.0 | | | | |
| March | 200.0 | 180.0 | 9.0 | | | |
| April | 25.0 | 80.0 | 160.0 | 11.0 | | |
| May | 33.0 | 55.0 | 80.0 | 180.0 | 7.0 | |
| June | 15.0 | 15.0 | 40.0 | 120.0 | 200.0 | 6.0 |
| July | 15.0 | 20.0 | 25.0 | 65.0 | 95.0 | 180.0 |
| August | 12.0 | 7.0 | 12.0 | 45.0 | 60.0 | 86.0 |
| September | 10.0 | 12.0 | 7.0 | 25.0 | 45.0 | 75.0 |
| October | 5.0 | 12.0 | 7.0 | 10.0 | 25.0 | 55.0 |
| November | 2.0 | 7.0 | 18.0 | 12.0 | 12.0 | 24.0 |
| December | 1.0 | 5.0 | 9.0 | 7.0 | 8.0 | 12.0 |
| TOTAL | 503.0 | 405.0 | 367.0 | 475.0 | 452.0 | 438.0 |

To validate claims payment data and to calculate reserves for outstanding incurred claims not yet paid, different methods may be used. In any particular year, claims are unlikely to be more than 95% to 98% complete at 90 days after the end of a year[14]. For the purposes of model development, the additional 2% to 5% of outstanding claim amounts is unlikely to have a material effect on the results. Pharmacy claims generally complete more quickly (most within the month incurred); outpatient and professional claims complete less quickly. Historically, inpatient claims completed least quickly, but as the percentage of claims that are adjudicated automatically continues to increase, the minimum period for "completeness" of inpatient claims has also tended to fall. More important than the *definition* of a standard run-out period, however, is the use of consistent incurred/paid periods for each period of a study (that is, claims completeness should be consistent between periods).

**Hospital claims**, often referred to as "facility" claims, can be categorized broadly as Inpatient and Outpatient. Because of their cost, considerable attention is directed at analyzing (and reducing) inpatient admissions, and the prediction of inpatient stays is a frequent target of predictive modelers. Inpatient claims are generated as a result of an admission to a facility involving an overnight stay. Because of the duration of inpatient admissions, there can be more than one inpatient claim record, containing

[14] This evaluation of completeness is often referred to as "Incurred in 12 (months) Paid in 15." Claims in the first month have had 15 months to mature; claims in the last month only 3. In total (for the year) the claims are reasonably mature, however.

multiple claims lines, for an admission. It is not unusual for a claim to be generated for an inpatient stay at the close of a month and, if the patient is still admitted, to have a new claim generated for the same admission but be "reset" to the first day of the new billing month. Complications also occur in billing for inpatient stays that involve a transfer to a different facility or, sometimes, an admission resulting from a visit to the Emergency Room. Re-admissions occurring within a brief period of discharge are another complicating factor when assessing whether an inpatient episode is a continuation of an earlier admission, or a new admission. Inpatient claims are most commonly identified by the Bill Type, but can be identified from room and board revenue codes. Inpatient claims can be classified as acute or sub-acute (such as skilled-nursing and rehabilitation stays).

Outpatient facility claims are generated for services such as emergency room visits, ambulatory surgeries, and other services provided in an institutional setting where there is a facility charge (usually in addition to a "professional" charge, which will be discussed in the next section).

Facility claims are billed using the UB form (Uniform Billing). In virtually all cases, payment for inpatient and outpatient claims is based on the Revenue Code listed on the UB04. The UB04 form, also known as a CMS-1450, is the latest iteration of a standard claim form used for billing Medicare services, but used more generally for other payers. The prior version, UB92, was discontinued in 2007. There are often multiple revenue codes per claim, each carrying a separate line item charge. When the contract with the payer requires reimbursement based on DRGs (Diagnosis Related Groups), payers put facility claims through a "DRG Grouper" in the adjudication process, and are reimbursed based on a DRG. The DRG grouping process maps ICD-9 diagnoses, procedures, patient age, sex, discharge status and the presence or absence of complicating conditions into one of approximately 500 DRGs. For providers that are reimbursed based on billed charges, the billed charges contained in the UB04 are the basis of reimbursement.

**Physician claims** often are classified into a larger category of "Professional" claims. This grouping encompasses all services rendered by a physician or other non-physician medical professional such as Nurse Practitioner, Respiratory Therapist, or Physical Therapist regardless of place of service.

It is important to note that with respect to hospital admissions, two claims may be submitted: the UB04 (or CMS 1450) for the facility services (room and board, ancillary services, in-hospital drugs, etc.) and the CMS 1500 for the services of a physician, assistant surgeon, anesthesiologist, etc. Claims of the facility and the professions are sometimes bundled together into a single bill, but this is unusual. More often, separate bills are issued for different types of inpatient services.

**Pharmacy Claims:** Of all data sources, pharmacy claims have the best overall quality. The nature of the pharmacy transaction, as compared to that of a medical encounter, is relatively straightforward, thus lending itself to routine and complete recording of the transaction. They are adjudicated quickly, if not immediately,

resulting in a faster completion rate than medical claims. The essential fields are nearly always populated. Unfortunately, because pharmacy claims do not contain diagnosis codes (diagnosis has to be inferred based on the therapeutic use of a particular drug) they are not as robust for predictive purposes. There are some exceptions, for example in the area of drug-drug interactions. Despite the best efforts of Pharmacy Benefit Managers (PBMs) and Pharmacies, conflicting prescriptions are sometimes filled, and the presence of conflicting drugs (drug-drug interactions) in the patient record is highly predictive of adverse outcomes. Moreover, there are enrollees with prescriptions for multiple drug classes, sometimes as many as four or more (**polypharmacy**). Such enrollees are statistically more risky, regardless of the specific therapeutic classes of their prescriptions.

All outpatient drugs are assigned an **NDC** (**National Drug Code**) identifier. This is a unique, three-segment number, which is a universal product identifier for human drugs. The NDC code is assigned by the U.S. Food and Drug Administration (FDA) which inputs the full NDC number and the information submitted as part of the listing process into a database known as the Drug Registration and Listing System (DRLS). On a monthly basis, the FDA extracts some of the information from the DRLS data base (currently, properly listed marketed prescription drug products and insulin) and publishes that information in the NDC Directory. There are a number of reasons why a drug product may not appear in the NDC Directory. For example:

- the product may not be a prescription drug or an insulin product;
- the manufacturer has notified the FDA that the product is no longer being marketed;
- the manufacturer has not complied fully with its listing obligations and therefore its product is not included until complete information is provided.

The FDA standard code contains 10 digits, while HIPAA requires an 11-digit number. To conform to a HIPAA compliant code, FDA-compliant codes are often transformed with the addition of an extra "0." This complete code, or NDC, identifies the labeler, product, and trade package size. The first segment, the labeler code, is assigned by the FDA. A labeler is any firm that manufactures (including repackers or relabelers) or distributes the drug under its own name. The second segment, the product code, identifies a specific strength, dosage form, and formulation for a particular firm. The third segment, the package code, identifies package sizes and types. Both the product and package codes are assigned by the firm. The NDC will have one of the following numbers of digits: 4-4-2, 5-3-2, or 5-4-1.

An example of an NDC code is 00087-6071-11.

| | |
|---|---|
| 00087: | Bristol-Myers Squibb. |
| 6071: | Glucophage 1000 mg tablets (Glucophage, or metformin, is an oral anti-diabetic medication). |
| 11: | 100 unit-size bottle. |

A typical drug claim file may also contain some descriptive information related to the drug itself. While it is most convenient to obtain all this information from one source, descriptive pharmacy drug information is available on various standardized databases. Access to supplemental databases can help to fill in the "blanks" regarding drug type/therapeutic class, brand vs. generic indicators, etc. In addition, drug codes are updated much more frequently than Medical codes, making maintenance of drug-based algorithms and analysis of drug claims more difficult. Supplemental databases can solve many of these challenges, but may be costly to obtain and can also be proprietary[15].

In addition to the information contained in the NDC code, a drug claim will contain other information. The key fields provided on pharmacy data include number of days supply of the drug, number of refills, prescription fill date, and the amounts paid and allowed on the claim.

Pharmacy claims have limitations. Prescriptions written by a physician but never filled will not generate claims. An April 8, 2009 report in the Wall Street Journal cited a study that found that cost was the reason that 6.8% of all brand-name prescriptions, and 4.1% of all generic prescriptions were unfilled. A significant limitation in pharmacy claims, compared with medical claims, is the absence of a diagnosis on the claim. Thus from pharmacy claims alone it is impossible to determine precisely the member's diagnosis. In addition, while a prescription is filled, it is not possible to determine whether the member actually took the medications. Fortunately, diagnoses may be extrapolated from the therapeutic class (or classes) of drug claims. There are several commercial grouper models available to the analyst who wishes to generate diagnoses from drug claims.

**Laboratory Values:** at present, it can be difficult to obtain laboratory values, and it requires effort to render them useful for modeling or analysis. The percentage of regular submissions of laboratory values is increasing, although even in Managed Care plans, a relatively low percentage of tests (often fewer than 25%) are processed by contracted reference laboratory vendors, due to out-of-network usage. It is important to recognize that, even if full reporting were available from reference laboratories, these values would still not represent all of the tests performed on patients, because of the large number of tests performed in-hospital, which are rarely available for analytical purposes. Few vendors code test results consistently using a standard protocol (such as the LOINC[16] code). Data from large national reference laboratory vendors are more likely to contain standardized formats than those from hospital laboratories. The potential for their use is very high and the availability of laboratory values is always worth investigating.

---

[15] See the section on Drug Groupers for more information about models that assign therapeutic classes, as well as grouper models that assign diagnoses from drug codes.

[16] **Logical Observation Identifiers Names and Codes (LOINC)** is a universal standard for identifying medical laboratory observations. It was developed and is maintained by the Regenstrief Institute, Inc. a U.S. non-profit medical research organization, in 1994.

**Patient Self-Reported Data:** Although not widely available at present, self-reported data is likely to be one of the new frontiers of healthcare analytics. Availability of self-reported data varies considerably. As PHR/EHR technology becomes more readily available and user-friendly, it will be easier to involve the patient in management of his/her own medical information. Member portals, where members have access to their own detailed medical records online, would not only allow for better care, but could be collected into databases for supplementing claims data in other forms. Additionally, Health Risk Assessments (HRA) are available for some populations; these generally provide information on behavioral risk factors (e.g., smoking, diet, exercise) and functional health status which are not available from claims. Self-reported data include member compliance with triggered activities (such as participation in smoking cessation or on-line education courses) as well as information about non-reimbursed services. As more emphasis is placed on member cost-sharing and reduction in expenses through healthier lifestyle and reduction of health risks, self-reported data will become an important method of identifying risk and risk reduction opportunities.

### 4.3 Interpretation of Claims Codes

In order to perform modeling, it is important to interpret the actual values of codes contained in the common claims forms. We cover a number of these types of codes and their interpretation.

#### PROCEDURE (CPT AND HCPCS) CODES

**Procedure codes** are those codes assigned by professionals to describe the services performed. They are a sub-set of codes under the **Healthcare Common Procedure Coding System (or HCPCS).** HCPCS is a two-tiered system, established by Centers for Medicare and Medicaid Services (CMS) to simplify billing for medical procedures and services, and CPT is the first of the two levels of coding.

**CPT codes** (HCPCS level I codes) are five-digit numerical codes from the Current Procedural Terminology code-set. CPT codes were first published in 1966 and are maintained and licensed by the American Medical Association. They are used primarily by physicians and other healthcare professionals to bill for services. CPT codes are categorized according to type of service. Examples are physician evaluation and management ("E&M"), laboratory, medical/surgical and radiology. Most of the procedures and services performed by healthcare professionals are billed using CPT codes. One of the major deficiencies of CPT codes for analytic purposes is the lack of specificity on some codes related to ancillary services such as supplies, injections and other materials, as well as newly-emerging services. In addition to CPT coding of procedures, the ICD-9 system also has procedure codes (found in Volume 3 of the ICD-9 code books). CPT codes are more detailed, and are more commonly encountered, but some payers require that both be coded for reimbursement. Table 8 shows the types of services by CPT code range.

**Table 8.**

| CPT Code Ranges | |
|---|---|
| Anesthesiology | 00100-01999; 99100 – 99140 |
| Surgery | 10040 – 69979 |
| Radiology | 70010 – 79999 |
| Pathology and Laboratory | 80002 – 89399 |
| Medicine (excl. Anesthesiology) | 90700 – 99199 (excl. 99100-99140) |
| Evaluation and Management | 99201 – 99499 |

As we shall see later, Evaluation and Management (E&M) codes are very important for confirming the validity of the diagnosis on an outpatient claim. Table 9 provides more detail on E&M codes:

**Table 9.**

| E&M Code Ranges | |
|---|---|
| Office and other outpatient services | 99201 – 99220 |
| Hospital inpatient services | 99221 – 99239 |
| Consultations | 99241 – 99275 |
| Emergency Department Services | 99281 – 99288 |
| Critical and Neonatal intensive care | 99291 – 99297 |
| Nursing facility/custodial | 99301 – 99333 |

**HCPCS:** There are over 2,400 Level II codes. These codes are alpha-numeric and include physician medical services such as durable medical equipment ("DME"), prosthetics, orthotics, injections, and supplies. Most of these codes begin with a single letter, followed by four numbers, ranging from A0000 – V0000. The Level II HCPCS codes are uniform in description across the United States.

**Place of Service:** Place of Service ("POS") codes are used in healthcare professional services claims to specify the setting in which a service is provided. POS codes range from 00-99, and they are maintained by CMS. These include codes for categories such as Professional Office, Home, Emergency Room, and Birthing Center.

**Type of Service:** Although a standardized code set for Type of Service ("TOS") is available in the public domain, it is complicated to assign TOS consistently and accurately. As a result, many payers assign their own TOS codes, based on other variables such as Place of Service and Bill Type. As a result, Type of Service is usually plan/payer specific, and analysis must be tailored to match the specific plan variables.

**Bill Type:** Bill type codes contain 3 digits and are defined by CMS. Each digit has significance. The first digit describes the type of facility (hospital, skilled nursing facility, home health, for example). The second digit gives information on the bill

classifications and differs for clinics and special facilities, inpatient and outpatient. The third digit is linked to frequency of billing for a claim (for example, a code of 2 is used for "interim-first claim;" code 3 is used for "interim- continuing claims," etc.).

**Revenue Code:** Revenue codes are a required part of billing for hospital facility services, as they are the chief piece of information linked to claims reimbursement for facility-based services. CMS 1500 and UB-04 revenue codes must be used to bill outpatient hospital facility services, and in some instances, a HCPCS procedure code is required in addition to the revenue code for accurate claims processing. There may be more than one revenue code per claim, as each line item can be assigned a separate charge for a separate service. Revenue codes are often used to distinguish between facility and professional claims. Inpatient facility claims will usually include a DRG; if no DRG is present, but revenue codes are provided then the claim is most likely "facility outpatient." Specific ranges are used to help define service type. Revenue codes 450-459, for example, denote "Emergency Room."

## 5 Clinical Identification Algorithms

As we have seen above claims data contains considerable information regarding a member's diagnosis, its severity, procedures, therapy and providers. As we have seen the presence of medical conditions is highly correlated with claims costs, so knowing about an individual's or population's conditions could help to predict that individual's or population's costs and utilization of medical services. Diagnosis codes alone number around 20,000, so that trying to understand a member's conditions and history is challenging, without considering other data sources such as drugs. Therefore, predictive modelers and other data miners frequently use **clinical identification algorithms,** a set of rules that, when applied to a claims data set, identifies the conditions present in a population, to reduce the information in claims to a more manageable volume. For the analyst who does not wish to build and maintain his own algorithms, there are several commercially-available models, referred to as "Grouper Models" or simply "Groupers." We shall review some of these later in the chapter.

There are several factors that need to be taken into account when building a clinical identification algorithm. The most important component of an algorithm is the diagnosis, but it is not the only component. An analyst building an algorithm must usually decide how to address the following components, in addition to diagnosis:

1. The source of the diagnosis (claims; laboratory values; medical charts, etc.);
2. If the source is claims, what claims should be considered? Which services will be scanned for diagnoses? Diagnoses can be derived from the following services, all of which may be found in medical claims:
    - Inpatient;
    - Outpatient;
    - Evaluation and Management;
    - Medical;
    - Surgical;
    - Laboratory;
    - Radiology; and
    - Ancillary services (e.g., durable medical equipment).

In addition, diagnoses may be obtained from other data sources, including self-reported (survey) data and inferred from drug claims.
3. If the claim contains more than one diagnosis, how many diagnoses will be considered for identification? Diagnoses beyond the first one or two recorded in a claim submission may not be reliable.
4. Over what time span, and with what frequency, will a diagnosis have to appear in claims for that diagnosis to be incorporated in the algorithm?
5. What procedures may be useful for determining the level of severity of a member's diagnosis?
6. What prescriptions drugs may be used to identify conditions? Because prescription drug claims do not contain diagnostic information, diagnoses may be inferred. As we discuss in Section 4.1.2, this is not always possible to do unambiguously.

Ultimately the analyst requires a systematic way for the information contained in the member diagnoses to be captured in a series of independent variables that will be used to construct a model. What diagnoses, frequencies and sources are used may depend on the proposed use of the algorithm. There are several challenges that face the analyst constructing a condition-based model.

1. The large number of different types of codes, as well as large numbers of procedure and drug codes. The numbers of codes and their redundancy (the same code will often be repeated numerous times in a member record) makes it essential to develop an aggregation or summarization scheme.
2. The level at which to recognize the condition. How many different levels of severity should be included in the model?
3. The impact of co-morbidities. Some conditions are often found together (for example heart disease with diabetes). The analyst will need to decide whether to maintain separate conditions and then combine where appropriate, or to create combinations of conditions.
4. The degree of certainty with which the diagnosis has been identified (confirmatory information).
5. The extent of "coverage" of the data. All members of a population with health coverage will be covered by claims data, but this will not be true, for example, for self-reported data.
6. The type of benefit design that underlies the data: for example, if the employee is part of a high deductible plan, certain low cost, high-frequency services may not be reimbursed through the health plan and therefore not generate the necessary claims-based diagnoses.

**Source Data.** Data for identification may come from different sources with a range of reliability and acquisition cost. A diagnosis in a medical record, assigned by a physician, will generally be highly reliable. Unfortunately, medical record information is seldom available on a large scale. For actuarial and other analytical work, the two sources that are most common are medical and drug claims. Laboratory values and self-reported data may be added to the algorithm, if available in sufficient quantity and accuracy ("coverage"). Medical and drug data contain considerable identification information, but

with different degrees of certainty. This leads to a second key element of the algorithm: frequency and source of identifying information.

F**requency and Source of codes.** The accuracy of a diagnosis may differ based on who codes the diagnosis, for what purpose and how frequently a diagnosis code appears in the member record. The more frequently a diagnosis code appears, the more reliable the interpretation of the diagnosis. Similarly, the source of the code (hospital, physician, laboratory) will also affect the reliability of the diagnostic interpretation.

A diagnosis that appears in the first position on a hospital claim is generally reliable, but the reliability of additional (co-morbid) diagnoses may be more questionable, as these diagnoses are often added to drive higher reimbursement. Similarly for physician-coded claims: if a patient visits the physician several times a year for diabetes check-ups, it is reasonably certain that the patient has diabetes. If the patient has only one diabetes diagnosis from a physician in 12 months, it is possible that diabetes was coded because the physician suspected this as a possible diagnosis but was conducting tests for confirmation. The absence of confirmatory diagnoses elsewhere (follow-up visits, prescription drug fills) makes the diagnosis suspect. It is usual to limit identification from physician codes to diagnoses from "E&M" ("Evaluation and Management") codes only. These codes indicate that the physician was actually treating the member, rather than conducting exploratory testing. Diagnoses on ancillary services such as laboratory claims often have less diagnostic precision and can include "rule-out" diagnoses: diagnosis codes for conditions that the patient is suspected as having but which have not been confirmed through interpretation of diagnostic tests.

Prescription drug data suffers from similar issues, together with issues that are unique such as multi-use drugs and non-approved ("off label") uses[17]. An example of a multi-use drug is a class of drugs called Beta Blockers. These drugs are normally prescribed for use in heart patients to reduce blood pressure and heart rate, and restore normal heart rhythm. We could therefore interpret the presence of a claim for a Beta Blocker prescription in a member's history as implying a possible heart condition. A regular history of repeat prescriptions would tend to confirm this interpretation. Beta Blockers are, however, also used for treatment of Anxiety disorders and Social Anxiety (being used on occasion by public speakers to overcome their fear of speaking in public). They have also been shown to be effective in the treatment of osteoporosis, alone or in combination with thiazide diuretics. The fact that a drug such as a Beta Blocker can be mapped to more than one class of disease makes the use of drug data alone a less reliable identifier of a diagnosis than a medical diagnosis, particularly if only one prescription is found.

**Severity and Co-morbid conditions.** Because of the multiplicity of codes, patient records may contain different codes for the same disease. The analyst who is building an algorithm needs to determine how many levels of severity he or she wants to recognize. We illustrate this with an example from diabetes.

[17] Multi-use drugs are drugs that may be used to treat more than one condition. An example of a multi-use drug is Thiazide, a diuretic, used to treat both the heart condition Hypertension and osteoporosis. The drug is tested and approved by the FDA for treatment of heart conditions, but not for osteoporosis. Its use for the latter condition is referred to as "Off Label" use.

**Table 10.**

| Codes for Identification of Diabetes | |
|---|---|
| ICD-9 Code | Code Description |
| 250.x | Diabetes Mellitus |
| 357.2 | Polyneuropathy in diabetes |
| 362.0 | Diabetic Retinopathy |
| 366.41 | Diabetic Cataract |
| 648.00 to 648.04 | Diabetes Mellitus (as other current condition in mother complicating pregnancy or childbirth)[18] |

Each of these codes identifies the possible presence of diabetes. The fourth digit within the 250.x class of codes, however, indicates complications and increased severity. Table 11 shows an expansion of the 250.x diabetes class of codes. It also provides information on the annual cost of patients who present with different diagnoses, relative both to all diabetes patients and to the average cost of all health plan members[19].

Depending on the use for which the algorithm is intended, the analyst may or may not want to capture the severity information contained in the fourth (or subsequent) digit of the diagnosis code.

Co-morbidities are a related issue that must also be addressed. Patients who have suffered from diabetes for some time, for example, frequently develop co-morbid heart conditions. The same heart conditions are found in patients without diabetes. From a health risk perspective, the question that must be asked is whether the heart risk in a patient with diabetes is the same as that in a patient without diabetes. If it is the same, an additive model by condition is appropriate. But the presence of additional co-morbid conditions could either raise or lower risk, compared with that posed by the sum of the individual risks. Ultimately we could leave to the data the task of determining the answer to this question. Nevertheless, how we handle this issue could affect our ultimate model. We can, for example, evaluate the relative cost (risk) of a patient with different diabetes diagnoses.

Table 12 shows a simple aggregation scheme grouping the detailed codes from Table 11 into a hierarchical condition system. In our case the mapping of conditions to severity levels results in groups of codes with approximately the same cost. Alternatively, we could map conditions based on clinical knowledge.

[18] Note that the pregnancy or childbirth codes are not without controversy as identifiers of diabetes. They are codes associated with gestational diabetes, a condition in which women with no previous history of diabetes exhibit elevated glucose levels during pregnancy. There is a specific code for gestational diabetes (648.8) but coding errors occur in this category and it is not always possible to distinguish between gestational and other forms of diabetes.

[19] Adapted from Duncan (2011).

**Table 11.**

| Relative Costs of Members with Different Diabetes Diagnoses* | | | | |
|---|---|---|---|---|
| **Diagnosis Code ICD-9-CM** | **Description** | **Average cost PMPY** | **Relative cost for all diabetics** | **Relative cost for all members** |
| 250 | A diabetes diagnosis without a fourth digit (i.e., 250 only). | $13,000 | 100% | 433% |
| 250.0 | Diabetes mellitus without mention of complication | $10,000 | 77% | 333% |
| 250.1 | Diabetes with ketoacidosis (complication resulting from severe insulin deficiency) | $17,000 | 131% | 567% |
| 250.2 | Diabetes with hyperosmolarity (hyperglycemia (high blood sugar levels) and dehydration) | $26,000 | 200% | 867% |
| 250.3 | Diabetes with other coma | $20,000 | 154% | 667% |
| 250.4 | Diabetes with renal manifestations (kidney disease and kidney function impairment) | $25,000 | 192% | 833% |
| 250.5 | Diabetes with ophthalmic manifestations | $12,000 | 92% | 400% |
| 250.6 | Diabetes with neurological manifestations (nerve damage as a result of hyperglycemia) | $17,000 | 131% | 567% |
| 250.7 | Diabetes with peripheral circulatory disorders | $20,000 | 154% | 667% |
| 250.8 | Diabetes with other specified manifestations | $30,000 | 231% | 1000% |
| 250.9 | Diabetes with unspecified complication | $13,000 | 100% | 433% |
| 357.2 | Polyneuropathy in Diabetes | $20,000 | 154% | 667% |
| 362 | Other retinal disorders | $13,000 | 100% | 433% |
| 366.41 | Diabetic Cataract | $14,000 | 108% | 467% |
| 648 | Diabetes mellitus of mother complicating pregnancy childbirth or the puerperium unspecified as to episode of care | $12,000 | 92% | 400% |
| **TOTAL** | | $13,000 | 100% | 433% |

**Table 12.**

| Example of a Grouping System for Diabetes Diagnoses | | | |
|---|---|---|---|
| **Severity Level** | **Diagnosis Codes Included** | **Average Cost** | **Relative Cost** |
| 1 | 250; 250.0 | $10,000 | 77% |
| 2 | 250.5; 250.9; 362; 366.41; 648 | $12,500 | 96% |
| 3 | 250.1; 250.3; 250.6; 250.7; 357.2 | $18,000 | 138% |
| 4 | 250.2; 250.4 | $25,000 | 192% |
| 5 | 250.8 | $30,000 | 238% |
| | **TOTAL** (All diabetes codes) | $13,000 | 100% |

* The average cost for all members of this population is $ 3,000 PMPY.

This simple grouping indicates that, while there is a multiplicity of codes available for identifying diabetes, the relative cost information contained in the codes may be conveyed by a relatively simple grouping (in this case, 5 groups).

## 6 Sensitivity-Specificity Trade-Off

When building algorithms, the analyst always has to make decisions regarding sensitivity and specificity. Sensitivity and Specificity are important concepts in predictive modeling. **Sensitivity** refers to the percentage of "true positives" that are correctly identified as such. **Specificity** refers to the percentage of "true negatives" correctly identified. Thus if the measure of interest is, for example, diabetes, the sensitivity of an identification algorithm would be calculated as the percentage of those members of the population who have diabetes who are correctly identified, while the specificity of the algorithm refers to the number of healthy individuals correctly identified (and not incorrectly identified as having diabetes).

In medical applications the diagnosis of diabetes is ultimately assigned by a clinician, often after a series of tests. As we have discussed previously, it is frequently impossible to obtain the necessary level of detailed clinical data on every member of the population to determine whether a member has diabetes, and the analyst must use secondary sources (such as claims) for this identification. Determining whether a member identified as a diabetic by an algorithm is a "false positive" (does not have the condition) cannot be determined directly and must be inferred from other evidence.

Use of a sensitive algorithm to identify diabetics will identify a larger population, but at the risk of including "false positives." In some applications, such as care management, where follow-up interventions are often performed, this may not be a significant drawback. In other applications, such as provider reimbursement or evaluation of programs, identification of members who do not truly have a condition can distort the analysis. In our example in Table 11, the average cost of all members was $3,000, compared with the average cost for a person with diabetes of about $13,000. Clearly, inclusion of an individual without diabetes in a diabetic population (as a "false positive") will distort the average cost of that population. For this reason, it is always important to determine the use which will be made of the algorithm before determining the degree of specificity to be included.

### 6.1 Constructing an Identification Algorithm

Construction of an Identification Algorithm requires that the analyst make decisions about the source of data, the codes to be used, and rules for frequency with which specific codes are identified in the member record. We illustrate the construction of an example below with an identification algorithm for diabetes:

**Table 13.**

| Example of a Definitional Algorithm | | | |
|---|---|---|---|
| **Disease** | **Type** | **Frequency** | **Codes** |
| **Diabetes Mellitus** | Hospital Admission or ER visit with diagnosis of diabetes in any position | At least one event in a 12-month period | ICD-9 codes 250, 357.2, 362.0, 366.41, 648.0 |
| | Professional visits with a primary or secondary diagnosis of diabetes | At least 2 visits in a twelve month period | CPT Codes in range of 99200-99499 series E & M codes or 92 series for eye visits |
| | Outpatient Drugs: dispensed insulin, hypoglycemic, or anti-hyperglycemic prescription drug | One or more prescriptions in a twelve month period | Diabetes drugs (see HEDIS or similar list of drug codes). |
| **EXCLUDE gestational diabetes[20].** | Any (as above) | As above | 648.8x |

## 6.2 Sources of Algorithms

For an analyst who does not wish to construct his or her own algorithms, algorithms may be obtained from other sources. The following are examples of commercially-available algorithms.

1. NCQA (NATIONAL COMMITTEE FOR QUALITY ASSURANCE): The HEDIS Data Set (HEALTHCARE EFFECTIVENESS DATA AND INFORMATION SET)

HEDIS is a tool used by health insurance plans to measure performance on important dimensions of care and service. It is offered by NCQA, which adapted and refined the first set of measures initially developed by a coalition of health plans and employer groups. The first widely-used and accepted version of HEDIS was version 2.0 released in 1993. It is a tool used by more than 90% of America's managed health care plans (and a growing number of PPO plans) to measure performance on important dimensions of care and service. HEDIS consists of 71 measures across 8 domains of care. Algorithms for identifying conditions (and appropriate services or medications) are provided for the following services (among others):

- Use of Appropriate Medications for People with Asthma
- Cholesterol Management for Patients with Cardiovascular Conditions
- Controlling High Blood Pressure
- Antidepressant Medication Management
- Breast, Cervical and Colorectal Cancers
- Comprehensive Diabetes Care

HEDIS measures are often calculated as the quotient of a numerator and denominator. The HEDIS algorithms provide definitions for both numerators (the numbers of health

[20] Gestational diabetes is excluded because it is non-recurring, that is, the diabetes is not likely to affect the member's costs and utilization prospectively.

plan members who meet the specific test) and denominators (all members who qualify during the measurement period for the required service) and are often used by analysts for population identification. HEDIS 2010 algorithms are found in NCQA's HEDIS 2010 Volume 2: Technical Specifications (currently at http://www.ncqa.org/tabid/78/Default.aspx). The electronic version is highly recommended. Volume 2 is updated each year and available shortly after the HEDIS season ends in June. In early October, critical updates and corrections to Volume 2 are available for download from the NCQA website. Support in the form of FAQs is available as is e-mail support. NDC codes lists (which are defined in Volume 2) are generally updated in November or December See Appendix 4 in Volume 2 for information on HEDIS terminology.

2. DMAA (Disease Management Association of America, now renamed The Care Continuum Alliance)

DMAA sponsors a regular series of workshops on outcomes evaluation. Part of the work performed in the workshops is the development of algorithms for identification of Chronic Diseases. DMAA's algorithms are published in "*Dictionary of Disease Management Terminology*" (2nd edition) (Duncan (2008)) and its "*Outcomes Guidelines Reports*." (Care Continuum Alliance (2010)). It should be noted that the purpose of these algorithms is the evaluation of programs, so they may tend to be more specific than sensitive in their identification. CCA's Outcomes Guidelines reports are available at no cost and may be found at the Care Continuum Alliance website, www.carecontinuum.org.

## 7 Construction and Use of Grouper Models

Above, we introduced claims codes and saw that there are different ways of identifying a disease, and that within "families" of diseases we encounter different levels of severity. We also discussed identification algorithms: rules that are applied to datasets to drive consistent identification of conditions and their severity within a member population. Such consistency is important, particularly for risk adjustment and predictive modeling applications where comparison of populations in different geographies or at different points in time is often required, a consistent and transparent algorithm is important.

Grouper models are commercially-available models that apply fixed, pre-defined algorithms to identify conditions present in the population. They provide a means of identifying member conditions, with the practical value (for modelers) that the algorithms are maintained by an external party. In addition to their value as algorithms, grouper models have achieved prominence because they are viewed as predictive models, providing a means of "scoring" patients for relative risk and cost. These models frequently generate a "relative risk score" that is used for underwriting, reimbursement and other applications. The relative risk score is simply a translation of the member's predicted cost to a numeric scale, based on a fixed relationship between the points scale and the predicted member cost. For example, in Table 12, members with risk level 3 have an expected cost of $18,000 per year. Assuming that the member average cost ($3,000) translates to a score of 1.0, a member in risk level 3 would score 6.0 on the relative risk scale.

Although algorithms may be developed by the analyst from first principles using available data, as discussed in Section 5, there are three primary reasons that may make use of commercially-available grouper models ("groupers") preferable.

1. There is a considerable amount of work involved in building algorithms, particularly when this has to be done for the entire spectrum of diseases. Adding drug or laboratory sources to the available data increases the complexity of development.
2. While the development of a model may be within the scope and resources of the analyst who is performing research, use of models for production purposes (for risk adjustment of payments to a health plan or provider groups for example) requires that a model be maintained to accommodate new codes. New medical codes are not published frequently, but new drug codes are released monthly, so a model that relies on drug codes will soon be out of date unless updated regularly.
3. Commercially-available clinical grouper models are used extensively for risk adjustment when a consistent model, accessible to many users, is required. Providers and plans, whose financial stability relies on payments from a payer, often require that payments be made according to a model that is available for review and validation. The predictive accuracy and usefulness of commercially available models has been studied extensively by the Society of Actuaries, which has published three comparative studies in the last 20 years[21].

The use and effectiveness of the models for risk adjustment or other purposes is outside the scope of this chapter. Rather, our interest is in the underlying clinical models that the grouper models offer, which simplify the analyst's task of identifying condition(s) present within a member record. If a grouper model is used, it is important to understand its construction, including (if possible) the clinical categories that the grouper recognizes, and the rules that the developer has applied to map diagnoses into these categories. Grouper models, while useful, are not without their challenges. Primary among these is the lack of transparency in their construction: the precise definition of code groupings is generally not available. Also, because the analyst cannot define either the codes, their frequency or source in the algorithm, grouper models essentially determine the sensitivity/specificity trade-off.

There are several grouper models available on the market, each with its own unique approach to algorithm construction. Models have also been developed for different uses, although over time they have come to be applied somewhat interchangeably. In order to use grouper models in projects, it is important to know what their advantages and disadvantages are.

The models that are most frequently encountered, and which are discussed in this chapter are summarized in Table 14. This table also indicates the data source used in the different models.

While most groupers contain common elements (grouping different diagnoses into successively higher-order groupings) there are some differences in approach and use of medical claims groupers. We note some of these here; a more detailed discussion is available in Duncan (2011).

[21] See Dunn et al (1996), Cumming et al (2003), Winkelman and Mehmud (2007).

**Table 14.**

| Commercially Available Grouper Models | | |
|---|---|---|
| **Company** | **Risk Grouper** | **Data Source** |
| CMS | Diagnostic Risk Groups (DRG) (There are a number of subsequent "refinements" to the original DRG model) | Hospital claims only |
| CMS | HCCs | Age/Sex, ICD -9 |
| 3M | Clinical Risk Groups (CRG) | All Claims (inpatient, ambulatory and drug) |
| IHCIS/Ingenix | Impact Pro | Age/Sex, ICD-9<br>NDC, Lab |
| UC San Diego | Chronic disability payment system<br>Medicaid Rx | Age/Sex, ICD -9<br>NDC |
| Verisk Sightlines™ | DCG<br>RxGroup | Age/Sex, ICD -9<br>Age/Sex, NDC |
| Symmetry/Ingenix | Episode Risk Groups (ERG)<br>Pharmacy Risk Groups (PRG) | ICD – 9, NDC<br>NDC |
| Symmetry/Ingenix | Episode Treatment Groups (ETG) | ICD – 9, NDC |
| Johns Hopkins | Adjusted Clinical Groups (ACG) | Age/Sex, ICD – 9 |

The earliest example of a grouper is the Diagnostic Risk Group system developed at Yale University in the 1980s and still in use (with successive modifications) by CMS for reimbursement of hospital services irrespective of the *actual* resources used to treat the patient. The DRG system projects relative resource use as a single number, providing a normative cost for the services appropriate to patients of the same diagnosis, age, sex, co-morbidities, severity of illness, risk of dying, prognosis, treatment difficulty, need for intervention and resource intensity.

There are several examples of medical and drug-based condition-based groupers: HCCs (used by CMS for reimbursement of Medicare Advantage plans), ACGs, DCGs, CRGs and CDPS. The Symmetry Episode Grouper (ETG) applies a different approach, combining services into clinically homogenous episodes of care, regardless of treatment location or duration. Unlike condition-based models, in which the unit of observation is generally the patient-year, ETGs group services over the episode, which may last less than a year. A member may experience multiple episodes of care for different diagnoses in the course of a year, or, in the case of chronic diseases, be subject to a year-long "episode."

While they are frequently used for the same applications (all models, for example, result in relative risk scores for patients) the design of different models makes them more suitable for different applications. All models are potentially useable for high-risk member case finding ("predictive modeling"), while the grouping of resources into episodes of care makes ETGs an obvious choice for provider evaluation.

### 7.1 Drug Grouper Models

In addition to models that group diagnosis codes, commercial models are also available to group drug codes, impute diagnoses and assign relative risk scores. We classify these models into two classes: Therapeutic Class Groupers and Drug-based risk adjusters.

As we discussed above, there are hundreds of thousands of drug (NDC) codes. Therapeutic groupers assign individual groupers to therapeutic classes, enabling analysis such as compliance or persistency of a member with appropriate drug therapy by searching for the presence of a class of drug within the population, rather than individual NDC codes. Two examples of commercially-available models that group drug codes into a hierarchy of therapeutic classes are the American Hospital Formulary Service classification system and the Medispan Drug database, which classifies drugs by Generic Product indicator (GPI).

While these two therapeutic grouper models perform a function similar to the grouping of ICD-9 Codes performed by Commercial grouper models, therapeutic classes, while useful for analysis, are not directly linked to risk adjustment or predictive modeling. An intermediate step is necessary, in which the member's diagnosis(es) must be inferred from the therapeutic class of drugs in the member record. Many of the vendors who provide risk adjustment/risk scoring models make available drug-only models that perform the clinical mapping from therapeutic class to diagnosis, and then assign relative risk scores to members based on their drug utilization experience alone. The risk scoring models discussed next do not, however, perform therapeutic class grouping, for which one of the therapeutic groupers will be necessary.

### 7.2 Drug-Based Risk Adjustment Models

Unlike the therapeutic class groupers discussed in Sections above, which map individual NDC codes to therapeutic classes, Drug-based Risk Adjustment models are a special case of the class of risk-adjuster models discussed previously. The medical grouper models provide a way of mapping the approximately 15,000 diagnosis codes to different condition categories, each of which represents a different relative risk. Drug-based risk-adjustment models serve a similar function, except that they are based on member *drug* utilization rather than member utilization of all medical services. Both types of models, in addition to categorizing member conditions, also generate a relative risk score. The model may be predictive of drug cost only or of drug plus medical costs. As Table 14 shows, most commercially-available grouper models also provide a drug-only model which relates the cost of either drugs only or all services (medical and drug) to the drug-only patient history.

## 8 Summary and Conclusions

In this chapter we have reviewed health risk, which we define as the likelihood of individuals of experiencing higher health resource utilization (either because of the use of more services, or more expensive services, or both) than other similarly-situated individuals. We have seen that increased utilization is often associated with patient diagnoses and treatments, and that information about commonly available risk factors such as age, sex, geography and medical condition is predictive of future utilization and cost. It is possible to derive information about diagnosis, treatment, provider of services, place of service and other potentially-predictive variables from health insurance claims data, as well as other data routinely collected by health insurance payers. The volume, ease of access and frequency of updating of health insurance claims makes this a valuable resource for the data miner who wants to test different hypotheses about health resource utilization.

In this chapter we have discussed the types of data most-frequently available to the data miner, as well as some of the techniques used to group data for interpretation. Once the data miner has assembled available data sources, the next challenge is to apply an appropriate model to begin to test hypotheses about the relationship between risk factors and health risk. Discussion of analytical models themselves is outside the scope of this chapter.

The size and scope of the health sector in every country and the urgency of the need to control healthcare costs makes healthcare data mining a crucial growth opportunity for the future.

## References

1. Dunn, D.L., Rosenblatt, A., Taira, D.A., et al.: A comparative Analysis of Methods of Health Risk Assessment. In: Society of Actuaries (SOA Monograph M-HB96-1), pp. 1–88 (October 1996)
2. Cumming, R.B., Cameron, B.A., Derrick, B., et al.: Comparative Analysis of Claims-Based Methods of Health Risk Assessment for Commercial Populations. Research Study Sponsored by Society of Actuaries (2002)
3. Winkelman, R., Mehmud, S.: A Comparative Analysis of Claims-Based Tools for Health Risk Assessment. In: Society of Actuaries, pp. 1–63 (April 2007), http://www.soa.org/files/pdf/risk-assessmentc.pdf
4. Duncan, I.: Healthcare Risk Adjustment and Predictive Modeling, pp. 1–341. Actex Publications (2011) ISBN 978-1-56698-769-1
5. Duncan, I. (ed.): Dictionary of Disease Management Terminology. Disease Management Association of America, Washington, D.C (2006) (now Care Continuum Alliance)
6. Duncan, I.: Managing and Evaluating Healthcare Intervention Programs, pp. 1–314. Actex Publications (2008) ISBN 978-1-56698-656-4
7. Bluhm, W. (ed.): Group Insurance, 5th edn., pp. 1–1056. Actex Publications (2005) ISBN 978-1-56698-613-7
8. Care Continuum Alliance. Outcomes Guidelines Report - Volume 5. Care Continuum Alliance (formerly DMAA) (5), 1–127 (May 2010), http://www.carecontinuum.org/OGR5_user_agreement.asp

# Chapter 5
# Mining Biological Networks for Similar Patterns

Ferhat Ay, Günhan Gülsoy, and Tamer Kahveci

Computer and Information Science and Engineering,
University of Florida, Gainesville, FL 32611
{fay,ggulsoy,tamer}@cise.ufl.edu

**Abstract.** In this chapter, we present efficient and accurate methods to analyze biological networks. Biological networks show how different biochemical entities interact with each other to perform vital functions for the survival of an organism. Three main types of biological networks are protein interaction networks, metabolic pathways and regulatory networks. In this work, we focus on alignment of metabolic networks.

We particularly focus on two algorithms which successfully tackle metabolic network alignment problem. The first algorithm uses a nonredundant graph model for representing networks. Using this model, it aligns reactions, compounds and enzymes of metabolic networks. The algorithm considers both the pairwise similarities of entities (homology) and the organization of networks (topology) for the final alignment. The second algorithm we describe allows mapping of entity sets to each other by relaxing the restriction of 1-to-1 mappings. This capturing biologically relevant alignments that cannot be identified by previous methods but it comes at an increasing computational cost and additional challenges. Finally, we discuss the significance of metabolic network alignment using the results of these algorithms on real data.

## 1 Introduction

Importance of the biological interaction data stems from the fact that the driving forces behind organisms' functions are described by these interactions. Unlike other types of biological data, such as DNA sequences, interaction data shows the roles of different entities and their organization to perform these processes. Analyzing these interactions can reveal significant information that is impossible to gather by analyzing individual entities.

With the recent advances in high throughput technology, in the last decades, significant amount of research has been done on identification and reconstruction of biological networks such as regulatory networks [1,2,3], protein interaction networks [4] and metabolic networks [5,6]. These networks are compiled in public databases, such as KEGG (Kyoto Encyclopedia of Genes and Genomes) [7], EcoCyc (Encyclopedia of Escherichia coli K-12 Genes and Metabolism) [8], PID (the Pathway Interaction Database.) [9] and DIP (Database of Interacting Proteins) [10]. These databases maintain information both in textual form and graphical form.

D.E. Holmes, L.C. Jain (Eds.): Data Mining: Found. & Intell. Paradigms, ISRL 25, pp. 63–99.
springerlink.com 

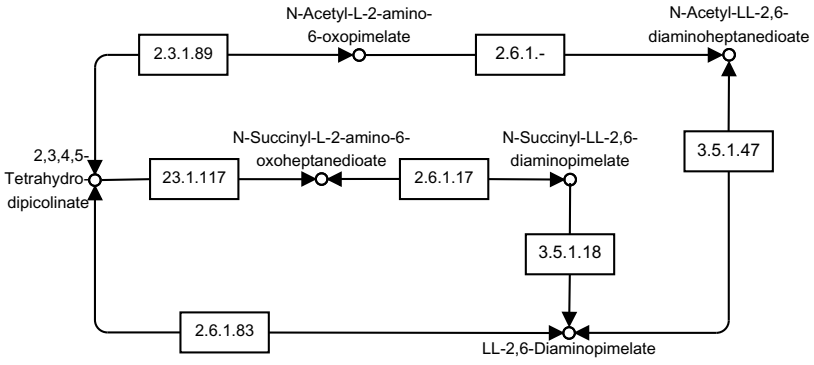



**Fig. 1.** A subnetwork of human lysine biosynthesis network. In this network, rectangles represents enzymes that catalyze reactions. Each enzyme is labeled with its Enzyme Commission (EC) number. The circles represent input and output compounds for the reactions.

An important type of biological network is *metabolic networks*. In metabolic networks relationships between different biochemical reactions together with their inputs, outputs and catalyzers are organized as networks. Analyzing these networks is necessary to capture the valuable information carried by them. An essential type of analysis is the comparative analysis which aims at identifying similarities between metabolisms of different organisms. Finding these similarities provides insights for drug target identification [11], metabolic reconstruction of newly sequenced genome [5] and phylogeny reconstruction [12,13]. A sample metabolic network can be seen in Figure 1.

To identify similarities between two networks it is necessary to find a mapping of their entities. Figure 2 shows an alignment of two networks with only one-to-one node mappings allowed. Every node need not be mapped in network alignments. Similar to sequence alignment, such nodes are called insertions or deletions (*indels*).

In the literature, alignment is often considered as finding one-to-one mappings between the molecules of two networks. In this case, the global/local network alignment problems are GI(Graph Isomorphism)/NP complete as the graph/subgraph isomorphism problems can be reduced to them in polynomial time [14]. Hence, even for the case described above, efficient methods are needed to solve the network alignment problem for large scale networks.

A number of studies have been done to systematically align different types of biological networks. For metabolic networks, Pinter *et al.* [15] devised an algorithm that aligns query networks with specific topologies by using a graph theoretic approach. Tohsato *et al.* proposed two algorithms for metabolic network alignment one relying on Enzyme Commission (EC [16]) numbers of enzymes and the other considering the chemical structures of compounds of the query networks [17,18]. Latterly, Cheng *et al.* developed a tool, *MetNetAligner*, for metabolic network alignment that allows a certain number of insertions and

deletions of enzymes [19]. These methods focus on a single type of molecule and the alignment is driven by the similarities of these molecules (e.g., enzyme similarity, compound similarity). Also, some of these methods limit the query networks to certain topologies, such as trees, non-branching paths or limited cycles. These limitations degrade the applicability of these methods to complex networks. One way to avoid this is to combine both topological features and homological similarity of pairwise molecules using a heuristic method. This approach has been successfully applied to find the alignments of protein interaction networks [20,21] and metabolic networks [22,23]. Another advantage of these two methods is that they improve the accuracy of the alignment algorithm without restricting the topologies of query networks.

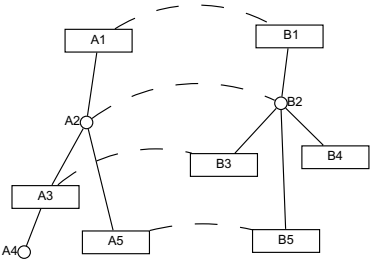



**Fig. 2.** An alignment of two hypothetical networks. The dashed lines represent the mapping between nodes of two networks. In this alignment, the nodes A1, A2, A3 and A5 are mapped to the nodes B1, B2, B3 and B5 respectively. Note that not all the nodes need to be mapped in an alignment. The nodes A4 and B4 are not mapped in this alignment.

In this chapter, we discuss two fundamental algorithms for aligning metabolic networks. We briefly describe a number of other computational methods aimed at solving this problem in Section 6. Before we describe the problem of network alignment or the solutions to the problem, we discuss major challenges inherent in biological networks in general and metabolic networks in particular.

– **Challenge 1.** A common delusion of a number of algorithms for metabolic network alignment is to use a model that focuses on only one type of entity and ignores the others. This simplification converts metabolic networks to graphs with only compatible nodes. The word *compatible* is used for the entities that are of the same type. For example, for metabolic networks two entities are compatible if they both are reactions or enzymes or compounds. The transformations that reduce the metabolic networks to graphs with only compatible entities are referred as *abstraction*. Reaction based [12], compound based [17] and enzyme based [15,24] abstractions are used for modeling metabolic networks. Figure 3 illustrates the problems with the enzyme based abstraction used by Pinter *et al.*[15] and Koyuturk *et al.*[24]. In the top portion of Figure 3(a), enzymes $E_1$ and $E_2$ interact on two different paths.

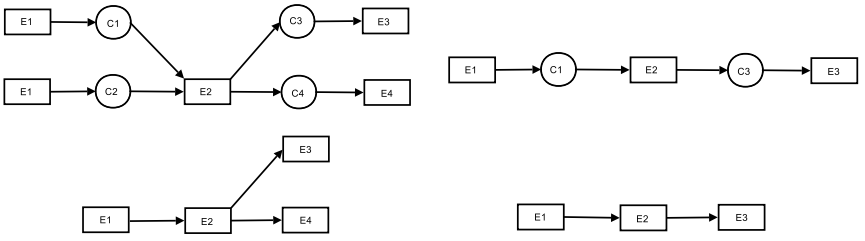



**Fig. 3.** *The effect of abstraction for metabolic networks.* Top figures in (a) and (b) illustrate two hypothetical metabolic networks with enzymes and compounds represented by letters $E$ and $C$, respectively. Bottom figures in (a) and (b) show the same networks after abstraction when the compounds are ignored. In (a) the two different paths between $E_1$ and $E_2$ in top figure are combined when compounds are ignored.

Abstraction loses this information and merges these two paths into a single interaction as seen in the bottom figure. After the abstraction, an alignment algorithm aligning the $E_1 \rightarrow E_2$ interactions in Figures 3(a) and 3(b) cannot realize through which path, out of two alternatives, the enzymes $E_1$ and $E_2$ are aligned. It is important to note that the amount of information lost due to abstraction grows exponentially with the number of branching entities.

– **Challenge 2.** Many of the existing methods limit the possible molecule mappings to only one-to-one mappings. As also pointed out by Deutscher *et al.* [25] considering each molecule one by one fails to reveal its function(s) in complex networks. This restriction prevents many methods from identifying biologically relevant mappings when different organisms perform the same function through varying number of steps. As an example, there are alternative paths for LL-2,6-Diaminopimelate production in different organisms [26,27]. LL-2,6-Diaminopimelate is a key intermediate compound since it lies at the intersection of different paths on the synthesis of L-Lysine. Figure 4 illustrates two paths both producing LL-2,6-Diaminopimelate starting from 2,3,4,5-Tetrahydrodipicolinate. The upper path represents the shortcut used by plants and *Chlamydia* to synthesize L-Lysine. This shortcut is not an option, for example, for *E.coli* or *H.sapiens* due to the lack of the gene encoding LL-DAP aminotransferase (2.6.1.83). *E.coli* and *H.sapiens* have to use a three step process shown with gray path in Figure 4 to do this transformation. Thus, a meaningful alignment should map the two paths when for instance, the lysine biosynthesis networks of human and a plant are aligned. However, since these two paths have different number of reactions traditional alignment methods, limited to one-to-one mappings, fail to identify this mapping.

The problem of metabolic network alignment is to find a mapping between the nodes of two networks which maximizes the alignment score. There are a

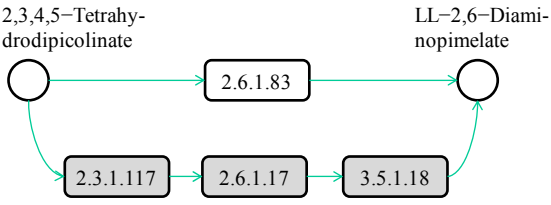



**Fig. 4.** A portion of Lysine biosynthesis network. Each reaction is represented by the Enzyme Commission (EC) number of the enzyme that catalyze it. Circles represent compounds (intermediate compounds are not shown). *E.coli* and *H.sapiens* (human) use the path colored by gray with three reactions, whereas plants and *Chlamydia* achieve this transformation directly through the path with a single reaction shown in white.

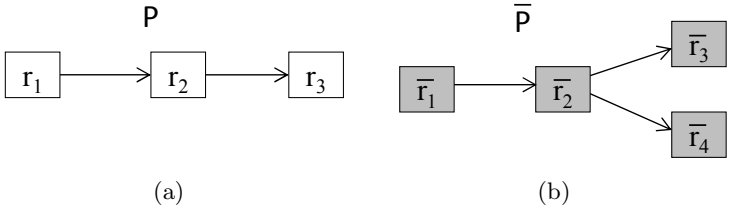



**Fig. 5.** A toy example with two hypothetical metabolic networks that will be used throughout this chapter. For simplicity, we only display the reactions of networks.

number of different schemes used in the literature for both mapping and scoring the alignments. Depending on the how scoring and mapping is defined, there are different approaches to align metabolic networks. In this chapter, we will describe two metabolic network alignment algorithms. In Section 2, we discuss an alignment algorithm that is free of abstraction and incorporates similarity of different types of entities. In Section 3, we present a more general alignment method, which also considers one-to-many mappings. We defer the formal definitions of these two versions of network alignment problem until the corresponding sections.

Throughout this chapter, we utilize a number of abbreviations and symbols to simplify the explanation of algorithms. Table 1 lists of the most frequently used symbols. We extend this notation list in Table 2 for the algorithm defined in Section 3. Also, we use the toy example in Figure 5 to explain different steps of the algorithms discussed in this chapter.

## 2 Metabolic Network Alignment with One-to-One Mappings

In this section, we present a network alignment algorithm that addresses the first challenge described in Section 1 [22,23]. This method considers all the biological

**Table 1.** Commonly used symbols in this chapter

| | |
|---|---|
| $P$, $P$ | Query metabolic networks |
| $\mathcal{R}, \mathcal{C}, \mathcal{E}$ | Set of all reactions, compounds and enzymes in a metabolic network |
| $r_i$, $\bar{r_j}$ | Reactions of query networks |
| $I_i$, $O_i$,$\mathcal{E}_i$ | Set of input compounds, output compounds and enzymes of reaction $r_i$ |
| $\phi$ | Relation which represents the alignment of the entities of query networks |
| $S_\mathcal{R}, S_\mathcal{C}, S_\mathcal{E}$ | Support matrices of reactions, compounds and enzymes |
| $H_\mathcal{R}, H_\mathcal{C}, H_\mathcal{E}$ | Homological similarity vectors of reactions, compounds and enzymes |
| $\alpha$ | Parameter adjusting relative weights of homology and topology |
| $\gamma_I, \gamma_O, \gamma_\mathcal{E}$ | Relative weights of similarities of input, output compounds and enzymes |

entities in metabolic networks namely compounds, enzymes and reactions that convert a set of compounds to other compounds by employing enzymes. At a high level this algorithm creates three eigenvalue problems one for compounds, one for reactions and one for enzymes. Then, it solves these eigenvalue problems using an iterative algorithm called power method [23]. The principal eigenvectors of each of these problems provide a good similarity score that is a mixture of homology and topology of corresponding entities. For each entity type, these scores define a weighted bipartite graph. The algorithm first extracts reaction mappings using maximum weighted bipartite matching algorithm on the corresponding bipartite graph. To ensure the consistency of the alignment, the next step of the method prunes the edges in the bipartite graphs of compounds and enzymes which lead to inconsistent alignments with respect to reaction mappings. Finally, the method applies maximum weighted bipartite matching algorithm to the pruned bipartite graphs of enzymes and compounds. The output of the method includes the extracted mappings of entities as an alignment together with their similarities and an overall similarity score.

The rest of this section is organized as follows: Section 2.1 describes the network model used by the algorithm. Section 2.2 outlines the algorithm. Sections 2.3 and 2.4 explains pairwise and topological similarities respectively. Section 2.5 shows how to combine these two similarities. Finally, sections 2.6 and 2.7 discusses how to extract and calculate the similarity score for the resulting alignment.

### 2.1 Model

Existing alignment methods use network models which only focus on a single type of entity, as stated in Challenge I. This simplification converts metabolic networks to the graphs with only compatible nodes. We use the word *compatible* for the entities that are of the same type. For metabolic networks, two entities are compatible if they both are reactions or enzymes or compounds. This algorithm uses a network model which considers all types of entities and interactions between them. In this section, we describe this model.

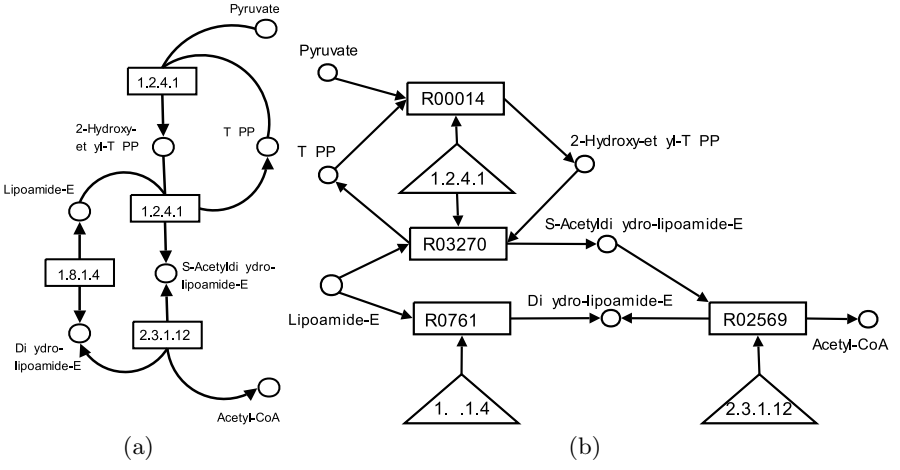



**Fig. 6.** *Graph representation of metabolic networks.* **(a)** A portion of the reference network of Alanine and aspartate metabolism from KEGG database **(b)** The graph representation used by the algorithm for this portion. Reactions are shown by rectangles, compounds are shown by circles and enzymes are shown by triangles.

Let $P$ be a metabolic network and $\mathcal{R}$=$\{\mathcal{R}_1,\mathcal{R}_2,\ldots,\mathcal{R}_{|\mathcal{R}|}\}$, $\mathcal{C}$=$\{\mathcal{C}_1,\mathcal{C}_2,\ldots,\mathcal{C}_{|\mathcal{C}|}\}$ and $\mathcal{E}$=$\{\mathcal{E}_1,\mathcal{E}_2,\ldots,\mathcal{E}_{|\mathcal{E}|}\}$ denote the sets of reactions, compounds and enzymes of this network respectively. Using this notation, the definition below formalizes the graph model employed by this algorithm:

**Definition 1.** *The directed graph, $P = (V, E)$, for representing the metabolic network $P$ is constructed as follows: The node set, $V = [\mathcal{R}, \mathcal{C}, \mathcal{E}]$, is the union of reactions, compounds and enzymes of $P$. The edge set, $E$, is the set of interactions between different nodes. An interaction is represented by a directed edge that is drawn from a node $x$ to another node $y$ if and only if one of the following three conditions holds:*
*1) $x$ is an enzyme that catalyzes reaction $y$.*
*2) $x$ is an input compound of reaction $y$.*
*3) $x$ is a reaction that produces compound $y$.* □

Figure 6 illustrates the conversion of a KEGG metabolic network to the graph model described above. As suggested, this model is capable of representing metabolic networks without losing any type of entities or interactions between these entities. This model avoids any kind of abstraction in alignment. Besides, this model is nonredundant since it avoids repetition of the same entity. In Figure 6(a) the enzyme 1.2.4.1 is shown twice to represent two different reactions, whereas in the latter model shown in Figure 6(b) it is represented as a single node.

## 2.2 Problem Formulation

This section formalizes the problem of aligning compatible entities of two metabolic networks with only one-to-one mappings. In this section, we first define the

alignment and consistency of an alignment. Then,6 we give the formal definition of the problem.

Let, $P, \bar{P}$ stand for the two query metabolic networks which are represented by graphs $P = (V, E)$ and $\bar{P} = (\bar{V}, \bar{E})$, respectively. Using the graph formalization given in Section 2.1, we replace $V$ with $[\mathcal{R}, \mathcal{C}, \mathcal{E}]$ where $\mathcal{R}$ denotes the set of reactions, $\mathcal{C}$ denotes the set of compounds and $\mathcal{E}$ denotes the set of enzymes of $P$. Similarly, we replace $\bar{V}$ with $[\bar{\mathcal{R}}, \bar{\mathcal{C}}, \bar{\mathcal{E}}]$.

**Definition 2.** *An* alignment *of two metabolic networks* $P = (V, E)$ *and* $\bar{P} = (\bar{V}, \bar{E})$*, is a* mapping $\phi : V \rightarrow \bar{V}$. □

Before arguing the consistency of an alignment, we discuss the reachability concept for entities. Given two compatible entities $v_i, v_j \in V$, $v_j$ is *reachable* from $v_i$ if and only if there is a directed path from $v_i$ to $v_j$ in graph $G$. As a shorthand notation, $v_i \Rightarrow v_j$ denotes $v_j$ is reachable from $v_i$.

Using the definition and the notation above, the definition of a consistent alignment is as follows:

**Definition 3.** *An alignment of two networks* $P = (V, E)$ *and* $\bar{P} = (\bar{V}, \bar{E})$ *defined by the mapping* $\phi : V \rightarrow \bar{V}$ *is* consistent *if and only if all the conditions below are satisfied:*

- *For all* $\phi(v) = \bar{v}$ *where* $v \in V$ *and* $\bar{v} \in \bar{V}$*,* $v$ *and* $\bar{v}$ *are compatible.*
- $\phi$ *is one-to-one.*
- *For all* $\phi(v_i) = \bar{v_i}$ *there exists* $\phi(v_j) = \bar{v_j}$ *such that* $v_i \Rightarrow v_j$ *and* $\bar{v_i} \Rightarrow \bar{v_j}$*, or* $v_j \Rightarrow v_i$ *and* $\bar{v_j} \Rightarrow \bar{v_i}$*, where* $v_i, v_j \in V$ *and* $\bar{v_i}, \bar{v_j} \in \bar{V}$. □

The first condition in Definition 3 filters out mappings of different entity types. The second condition ensures that none of the entities are mapped to more than one entity. The last condition restricts the mappings to the ones which are supported by at least one other mapping. That is, it eliminates the nonsensical mappings which may cause inconsistency as described in Figure 7.

Now, let, $SimP_\phi : (P, \bar{P}) \rightarrow \quad \cap[0, 1]$ be a pairwise network similarity function, induced by the mapping $\phi$. The maximum score (i.e., $SimP_\phi = 1$) is achieved when two networks are identical. In section 2.7, we will describe in detail how $SimP_\phi$ is computed after $\phi$ is created. In order to restate the problem, it is only necessary to know the existence of such similarity function.

In the light of the above definitions and formalizations, here is the *problem statement* considered in this section:

**Definition 4.** *Given two metabolic networks* $P = (V, E)$ *and* $\bar{P} = (\bar{V}, \bar{E})$*, the alignment problem is to find a consistent mapping* $\phi : V \rightarrow \bar{V}$ *that maximizes* $SimP_\phi(P, \bar{P})$. □

In the following sections, we describe the metabolic network alignment algorithm.

### 2.3 Pairwise Similarity of Entities

Metabolic networks are composed of different entities which are enzymes, compounds and reactions. The degree of similarity between pairs of entities of two networks is, usually, a good indicator of the similarity between the networks.

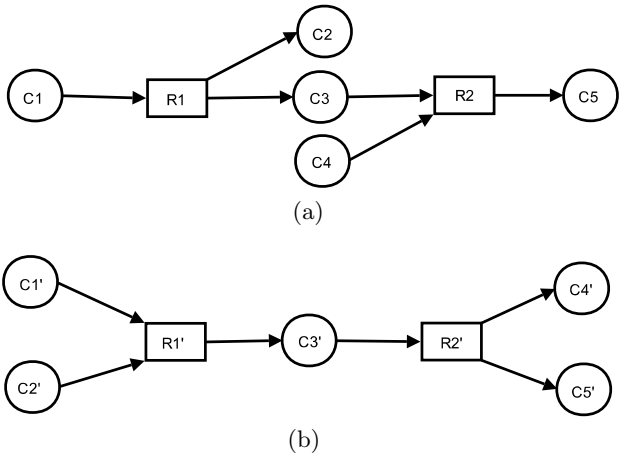



**Fig. 7.** *Consistency of an alignment.* Figures in (a) and (b) are graph representations of two query networks. Enzymes are not displayed for simplicity. Suppose that the alignment algorithm mapped the reactions R1 to R1' and R2 to R2'. In this scenario, a consistent mapping is C1-C1'. An example for a nonsensical mapping that causes inconsistency is C2' - C5, since it conflicts with the given mapping of reactions.

A number of similarity measures have been devised for each type of entity in the literature. In the rest of this section, we describe the similarity functions the algorithm uses for enzyme and compound pairs. We also discuss the similarity function the authors developed for reaction pairs. All pairwise similarity scores are normalized to the interval of $[0, 1]$ to ensure compatibility between similarity scores of different entities.

**Enzymes:** An enzyme similarity function is of the form $SimE : \mathcal{E} \times \bar{\mathcal{E}} \rightarrow \cap\,[0, 1]$. Two commonly used enzyme similarity measures are:

• *Hierarchical enzyme similarity score* [18] depends only on the Enzyme Commission (EC) [16] numbers which made up of 4 numerals. Starting from the leftmost numbers of two enzymes it adds 0.25 to similarity score for each common digit until two numbers differ. For instance, $SimE(6.1.2.4, 5.2.2.4) = 0$ since leftmost numbers are different, whereas $SimE(\mathbf{6}.1.2.4, \mathbf{6}.2.3.4) = 0.25$ and $SimE(\mathbf{6.1.2}.4, \mathbf{6.1.2}.103) = 0.75$.

• *Information content enzyme similarity score* [15] uses EC numbers of enzymes together with the information content of this numbering scheme. It is defined as $-log_2(h/|\ \ |)$ where $h$ is the number of enzymes that are elements of the smallest common subtree containing both enzymes and $|\ \ |$ is the number of all enzymes in the database. In order to maintain the compatibility, this score is normalized by dividing it to $log_2(|\ \ |)$. Figure 8 shows how to calculate information content enzyme similarity score for enzymes 6.1.2.4 and 6.1.2.103.

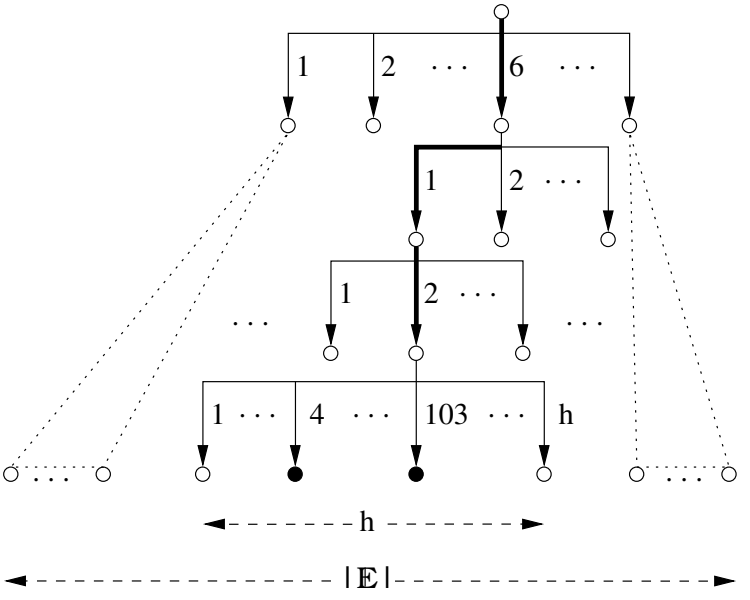



**Fig. 8.** Calculation of $SimE(6.1.2.4, 6.1.2.103)$ using information content enzyme similarity measure

**Compounds:** For compounds, the form of the similarity scores is $SimC : \mathcal{C} \times \bar{\mathcal{C}} \rightarrow \quad \cap[0, 1]$. Unlike the enzymes, there is no hierarchical numbering system for compounds. Therefore, using hierarchy is not an option in this case. Better approach for compounds is to consider the similarity of their chemical structures. Here is two different methods commonly used for compound similarity:

- *Identity score for compounds* is computing the similarity score as 1 if two compounds are identical and 0 otherwise.
- *SIMCOMP similarity score for compounds* is defined by Hattori *et al.* [28] This score is assessed by mapping chemical structures of compounds to graphs and then measuring the similarity between these graphs. This algorithm uses loose compound similarity scoring scheme of SIMCOMP.

**Reactions:** The authors define a similarity score for the reactions using the similarity scores for enzymes and compounds. An accurate similarity score for reactions should rather account for the process performed by the reaction than its label. This is because reactions catalyzed by enzymes affect the state of the network by transforming a set of input compounds to a set of output compounds. The similarity score for reactions depends on the similarities of the enzymes and the compounds that take place in this process.

The similarity function for reactions is of the form $SimR : \mathcal{R} \times \bar{\mathcal{R}} \rightarrow \quad \cap[0, 1]$. It employs the maximum weight bipartite matching technique. The following is a brief description of the maximum weight bipartite matching:

**Definition 5.** MAXIMUM WEIGHT BIPARTITE MATCHING. *Let, $A$ and $B$ be two disjoint node sets and $W$ be an $|A| \times |B|$ matrix representing edge weights between all possible pairs with one element from $A$ and one element from $B$, where existing edges correspond to a nonzero entry in $W$.* The Maximum Weight Bipartite Matching *of $A$ and $B$ is a one-to-one mapping of nodes, such that the sum of edge weights between the elements of these pairs is maximum. We denote this sum by $MWBM(A, B, W)$.* □

Let, $r_i$ and $\bar{r_j}$ be two reactions from $\mathcal{R}$ and $\bar{\mathcal{R}}$ respectively. The reaction $r_i$ is a combination of input compounds, output compounds and enzymes denoted by $[I_i, O_i, \mathcal{E}_i]$, where $I_i, O_i \subseteq \mathcal{C}$ and $\mathcal{E}_i \subseteq \mathcal{E}$. Similarly, define $\bar{r_j}$ as $[I_{\bar{j}}, O_{\bar{j}}, \mathcal{E}_{\bar{j}}]$. Additionally, compute the edge weight matrices $W_O$ and $W_I$ using the selected compound similarity score and $W_\mathcal{E}$ using the selected enzyme similarity.

The similarity score of $r_i$ and $\bar{r_j}$ is computed as:

$$\begin{aligned} SimR(r_i, \bar{r_j}) = {} & \gamma_I MWBM(I_i, I_{\bar{j}}, W_I) \\ & + \gamma_O MWBM(O_i, O_{\bar{j}}, W_O) \\ & + \gamma_\mathcal{E} MWBM(\mathcal{E}_i, \mathcal{E}_{\bar{j}}, W_\mathcal{E}) \end{aligned} \tag{1}$$

Here, $\gamma_I, \gamma_O, \gamma_\mathcal{E}$ are real numbers in $[0, 1]$ interval. They denote the relative weights of input compounds, output compounds and enzymes on reaction similarity respectively. Typical values for these parameters are $\gamma_I = \gamma_O = 0.3$ and $\gamma_\mathcal{E} = 0.4$. These values are empirically determined after a number of experiments. One more factor that defines reaction similarity is the choice of $SimE$ and $SimC$ functions. Since there are two options for each, in total there are four different options for reaction similarity depending on the choices of $SimE$ and $SimC$.

Now, it is time to create the pairwise similarity vectors ${H_\mathcal{R}}^0$, ${H_\mathcal{C}}^0$, ${H_\mathcal{E}}^0$ for reactions, compounds and enzymes, respectively. Since, calculation of these vectors is very similar for each entity type we just describe the one for reactions.

Figure 9 displays step by step computation of homology vector from the homological similarity matrix of our toy example. The entry ${H_\mathcal{R}}^0((i-1)|\mathcal{R}| + j)$ of ${H_\mathcal{R}}^0$ vector stands for the similarity score between $r_i \in \mathcal{R}$ and $\bar{r_j} \in \bar{\mathcal{R}}$, where $1 \leq i \leq |\mathcal{R}|$ and $1 \leq j \leq |\bar{\mathcal{R}}|$. We will use the notation ${H_\mathcal{R}}^0(i, j)$ for this entry since ${H_\mathcal{R}}^0$ can be viewed as a $|\mathcal{R}| \times |\bar{\mathcal{R}}|$ matrix. One thing to be careful about is that ${H_\mathcal{R}}^0, {H_\mathcal{C}}^0, {H_\mathcal{E}}^0$ vectors should be of unit norm. As we clarify in section 2.4, this normalization is crucial for stability and convergence of the algorithm. Therefore, we compute an entry of ${H_\mathcal{R}}^0$ as:

$${H_\mathcal{R}}^0(i, j) = \frac{SimR(r_i, \bar{r_j})}{||{H_\mathcal{R}}^0||_1} \tag{2}$$

In a similar fashion, all entries of ${H_\mathcal{C}}^0, {H_\mathcal{E}}^0$ are created by using $SimC$ and $SimE$ functions. These three vectors carry the homology information throughout the algorithm. Section 2.5 describes how they are combined with topology information to produce an alignment.

| | $\bar{r_1}$ | $\bar{r_2}$ | $\bar{r_3}$ | $\bar{r_4}$ |
|---|---|---|---|---|
| $r_1$ | 0.8 | 0.3 | 0.1 | 0.5 |
| $r_2$ | 0.1 | 0.9 | 0.2 | 0.7 |
| $r_3$ | 0 | 0.3 | 0.7 | 0.4 |

(a)

| | |
|---|---|
| $r_1$-$\bar{r_1}$ | 0.8 |
| $r_2$-$\bar{r_1}$ | 0.1 |
| $r_3$-$\bar{r_1}$ | 0 |
| $r_1$-$\bar{r_2}$ | 0.3 |
| $r_2$-$\bar{r_2}$ | 0.9 |
| $r_3$-$\bar{r_2}$ | 0.3 |
| $r_1$-$\bar{r_3}$ | 0.1 |
| $r_2$-$\bar{r_3}$ | 0.2 |
| $r_3$-$\bar{r_3}$ | 0.7 |
| $r_1$-$\bar{r_4}$ | 0.5 |
| $r_2$-$\bar{r_4}$ | 0.7 |
| $r_3$-$\bar{r_4}$ | 0.4 |

(b)

| | |
|---|---|
| $r_1$-$\bar{r_1}$ | 0.16 |
| $r_2$-$\bar{r_1}$ | 0.02 |
| $r_3$-$\bar{r_1}$ | 0 |
| $r_1$-$\bar{r_2}$ | 0.06 |
| $r_2$-$\bar{r_2}$ | 0.18 |
| $r_3$-$\bar{r_2}$ | 0.06 |
| $r_1$-$\bar{r_3}$ | 0.02 |
| $r_2$-$\bar{r_3}$ | 0.04 |
| $r_3$-$\bar{r_3}$ | 0.14 |
| $r_1$-$\bar{r_4}$ | 0.1 |
| $r_2$-$\bar{r_4}$ | 0.14 |
| $r_3$-$\bar{r_4}$ | 0.08 |

(c)

**Fig. 9.** Calculation of (a) similarity matrix, (b) similarity vector and (c) normalized similarity vector for our running example.

### 2.4 Similarity of Topologies

Previously, we discussed why and how pairwise similarities of entities are used. Although pairwise similarities are necessary, they are not sufficient. The induced topologies of the aligned entities should also be similar. In order to account for topological similarity, this section describes the notion of neighborhood for each compatibility class. After that, the method creates *support matrices* which allow the use of neighborhood information.

To be consistent with the reachability definition, the neighborhood relation definitions are in line with directions of interactions. In other words, these definitions distinguish between *backward neighbors* and *forward neighbors* of an entity. Let, $BN(x)$ and $FN(x)$ denote the backward and forward neighbor sets of an entity $x$. The construction of these sets for each entity type starts by defining neighborhood of reactions to build backbone for topologies of the networks. Then, using this backbone, neighborhood for compounds and enzymes are defined.

Consider two reactions $r_i$ and $r_u$ of the network $P$. If an output compound of $r_i$ is an input compound for $r_u$, then $r_i$ is a backward neighbor of $r_u$ and $r_u$ is a forward neighbor of $r_i$. The algorithm constructs the forward and backward neighbor sets of each reaction in this manner. For instance, in Figure 5(a), reaction $r_3$ is a forward neighbor of $r_2$, and $r_1$ is a backward neighbor of $r_2$.

A more generalized version of neighborhood definition can be given to include not only instant neighbors but also neighbors of neighbors and so on. However, this is not necessary since the method considers the support of indirect neighbors as we describe in section 2.5.

As stated before, neighborhood definitions of compounds and enzymes depend on the topology of reactions. Let, $c_s$ and $c_t$ be two compounds, $r_i$ and $r_u$ be two reactions of the network $P$. If $r_i \in BN(r_u)$ and $c_s$ is an input (output) compound of $r_i$ and $c_t$ is an input (output) compound of $r_u$ then $c_s \in BN(c_t)$ and $c_t \in FN(c_s)$. For example, in Figure 6(b), Pyruvate and Lipoamide-E are

neighbors since they are inputs of two neighbor reactions, namely R00014 and R03270. For enzymes the neighborhood construction is done similarly.

Utilizing the above neighborhood definitions, the algorithm creates support matrices for each compatibility class. These matrices represent the combined topological information of the network pair. Each entry of a support matrix represents the support given by the pair of entities that index the row to the pair of entities that index the column. Here, we only describe how to calculate the support matrix for reactions. The calculations for enzymes and compounds are similar.

**Definition 6.** *Let,* $P = ([\mathcal{R}, \mathcal{C}, \mathcal{E}], E)$ *and* $\bar{P} = ([\bar{\mathcal{R}}, \bar{\mathcal{C}}, \bar{\mathcal{E}}], \bar{E})$ *be two metabolic networks. The* support matrix *for reactions of* $P$ *and* $\bar{P}$ *is a* $|\mathcal{R}||\bar{\mathcal{R}}| \times |\mathcal{R}||\bar{\mathcal{R}}|$ *matrix denoted by* $S_{\mathcal{R}}$*. An entry of the form* $S_{\mathcal{R}}[(i-1)|\mathcal{R}| + \bar{j}][(u-1)|\mathcal{R}| + \bar{v}]$ *identifies the fraction of the total support provided by* $r_u, \bar{r_v}$ *mapping to* $r_i, \bar{r_j}$ *mapping. Let,* $N(u, \bar{v}) = |BN(\mathcal{R}_u)||BN(\bar{r_v})| + |FN(r_u)||FN(\bar{r_v})|$ *denote the number of possible mappings of neighbors of* $r_u$ *and* $\bar{r_v}$*.*

*Each entry of* $S_{\mathcal{R}}$ *is computed as:*

$$S_{\mathcal{R}}[(i-1)|\mathcal{R}|+\bar{j}][(u-1)|\mathcal{R}|+\bar{v}] = \begin{cases} \frac{1}{N(u,\bar{v})} & \text{if } (r_i \in BN(r_u) \text{ and } \bar{r_j} \in BN(\bar{r_v})) \\ & \text{or } (r_i \in FN(r_u) \text{ and } \bar{r_j} \in FN(\bar{r_v})) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

*After filling all entries, the zero columns of* $S_{\mathcal{R}}$ *are replaced with with* $|\mathcal{R}||\bar{\mathcal{R}}| \times 1$ *vector* $[\frac{1}{|\mathcal{R}||\bar{\mathcal{R}}|}, \frac{1}{|\mathcal{R}||\bar{\mathcal{R}}|} \ldots, \frac{1}{|\mathcal{R}||\bar{\mathcal{R}}|}]^T$*. This way support of the mapping indicated by the zero column is uniformly distributed to all other mappings.* □

Now, we describe the calculation of the entries of a support matrix on our running example in Figure 5. Let us focus on the support given by the mapping $(\{r_2\}, \{\bar{r_2}\})$ to mappings of their neighbors. We see that $FN(\{r_2\}) = 1$, $FN(\{\bar{r_2}\}) = 2$, $BN(\{r_2\}) = 1$ and $BN(\{\bar{r_2}\}) = 1$. Hence, the support of mapping $r_2$ to $\bar{r_2}$ should be equally distributed to its 3 (i.e. $1 \times 1 + 2 \times 1$) possible neighbor mapping combinations. This is achieved by assigning 1/3 to the corresponding entries of $S_{\mathcal{R}}$ matrix. Formation of the row corresponding to the support given by $r_2$-$\bar{r_2}$ mapping is illustrated in Figure 10.

We use the terms $S_{\mathcal{R}}$, $S_{\mathcal{C}}$ and $S_{\mathcal{E}}$ to represent the support matrices for reactions, compounds and enzymes, respectively. The power of these support matrices is that they enable distribution of the support of a mapping to other

| | $r_1$-$\bar{r_1}$ | ... | $r_3$-$\bar{r_3}$ | $r_3$-$\bar{r_4}$ |
|---|---|---|---|---|
| ... | ... | ... | ... | ... |
| $r_2$-$\bar{r_2}$ | $\frac{1}{3}$ | 0 | $\frac{1}{3}$ | $\frac{1}{3}$ |
| ... | ... | ... | ... | ... |

**Fig. 10.** Calculation of the support matrix for our running example. Only the row representing the support from $r_2$-$\bar{r_2}$ mapping to other mappings is shown (it only has three non-zero entries).

mappings according to distances between them. This distribution is crucial for favoring mappings whose neighbors can also be matched as well. In the following section, we describe an iterative process for appropriately distributing the mapping scores to the neighborhood mappings.

### 2.5 Combining Homology and Topology

Both the pairwise similarities of entities and the organization of these entities together with their interactions provide us great information about the functional correspondence and evolutionary similarity of metabolic networks. Hence, an accurate alignment strategy needs to combine these factors cautiously. In this subsection, we describe a strategy to achieve this combination.

From the previous sections, we have ${H_{\mathcal{R}}}^0$, ${H_{\mathcal{C}}}^0$, ${H_{\mathcal{E}}}^0$ vectors containing pairwise similarities of entities and $S_{\mathcal{R}}, S_{\mathcal{C}}, S_{\mathcal{E}}$ matrices containing topological similarities of networks. Using these vectors and matrices together with a weight parameter $\alpha \in [0,1]$ for adjusting the relative effect of topology and homology, the method transforms the problem into three eigenvalue problems as follows:

$$ {H_{\mathcal{R}}}^{k+1} = \alpha S_{\mathcal{R}} {H_{\mathcal{R}}}^k + (1-\alpha){H_{\mathcal{R}}}^0 \quad (4) $$

$$ {H_{\mathcal{C}}}^{k+1} = \alpha S_{\mathcal{C}} {H_{\mathcal{C}}}^k + (1-\alpha){H_{\mathcal{C}}}^0 \quad (5) $$

$$ {H_{\mathcal{E}}}^{k+1} = \alpha S_{\mathcal{E}} {H_{\mathcal{E}}}^k + (1-\alpha){H_{\mathcal{E}}}^0 \quad (6) $$

for $k \geq 0$.

In order to assure the convergence of these iterations, ${H_{\mathcal{R}}}^k$, ${H_{\mathcal{C}}}^k$ and ${H_{\mathcal{E}}}^k$ are normalized before each iteration.

**Lemma 1.** *$S_{\mathcal{R}}$, $S_{\mathcal{C}}$ and $S_{\mathcal{E}}$ are column stochastic matrices.*

**Proof.** This is the proof for the matrix $S_{\mathcal{R}}$ only. The proofs for $S_{\mathcal{C}}$ and $S_{\mathcal{E}}$ can be done similarly.

To prove that $|\mathcal{R}||\bar{\mathcal{R}}| \times |\mathcal{R}||\bar{\mathcal{R}}|$ matrix $S_{\mathcal{R}}$ is column stochastic, we need to show that all entries of $S_{\mathcal{R}}$ are nonnegative and the sum of the entries in each column is 1. The nonnegativity of each entry of $S_{\mathcal{R}}$ is assured by Definition 6. Now, let $c$ be an arbitrary column of $S_{\mathcal{R}}$ and $T_c$ be the sum of the entries of that column. Then, $\exists u \in [1, |\mathcal{R}|]$ and $\exists \bar{v} \in [1, |\bar{\mathcal{R}}|]$ such that $\lceil c/|\mathcal{R}| \rceil = u - 1$ and $c \equiv \bar{v}(mod|\mathcal{R}|)$. $u$ and $\bar{v}$ are indices of reactions $r_u$ and $\bar{r_v}$, respectively. By Definition 6, each entry of column $c$ is either 0 or $\frac{1}{N(u,\bar{v})}$, where $N(u,\bar{v}) = |BN(r_u)||BN(\bar{r_v})| + |FN(r_u)||FN(\bar{r_v})|$. Then, the $(i-1)|\mathcal{R}| + \bar{j}$th entry of column $c$ is $\frac{1}{N(u,\bar{v})}$ if and only if both $r_i$ and $\bar{r_j}$ are either forward neighbors or backward neighbors of $r_u$ and $\bar{r_v}$, respectively. Hence, $T_c = \sum_{t=1}^{N(u,\bar{v})} \frac{1}{N(u,\bar{v})} = 1$. Since $c$ is an arbitrary column, $T_x = 1$, for all $x \in [1, |\mathcal{R}||\bar{\mathcal{R}}|]$. Therefore, each column of $S_{\mathcal{R}}$ sums up to 1 and thus, $S_{\mathcal{R}}$ is column stochastic. □

**Lemma 2.** *Every column stochastic matrix has an eigenvalue $\lambda_1$ with $|\lambda_1| = 1$.*

**Proof.** Let, $A$ be any column stochastic matrix. Then, $A^T$ is a row stochastic matrix and $A^T e = e$ where $e$ is a column vector with all entries equal to 1. Hence, 1 is an eigenvalue of $A^T$. Since $A$ is a square matrix, 1 is also an eigenvalue of $A$. □

**Lemma 3.** *Let, $A$ and $E$ be $N \times N$ column stochastic matrices. Then, for any $\alpha \in [0,1]$ the matrix $M$ defined as:*

$$M = \alpha A + (1-\alpha)E \tag{7}$$

*is also a column stochastic matrix.*

**Proof.** Let, $C_i$, $D_i$, $T_i$ be the sum of the $i$th columns of $A$, $E$ and $M$, respectively. Since $A$ and $E$ are column stochastic for all $i \in [1,N]$, $C_i = D_i = 1$. Also, since $\alpha \in [0,1]$, $\alpha C_i + (1-\alpha)D_i = \alpha C_i - \alpha D_i + 1 = 1$. Hence, $T_i = \alpha C_i + (1-\alpha)D_i = 1$ for all $i \in [1,N]$ and $M$ is a column stochastic. □

**Lemma 4.** *Let, $A$ be an $N \times N$ column stochastic matrix and $E$ be an $N \times N$ matrix such that $E = He^T$, where $H$ is an N-vector with $||H||_1 = 1$ and $e$ is an N-vector with all entries equal to 1. For any $\alpha \in [0,1]$ define the matrix $M$ as:*

$$M = \alpha A + (1-\alpha)E \tag{8}$$

*The maximal eigenvalue of $M$ is $|\lambda_1| = 1$. The second largest eigenvalue of $M$ satisfies $|\lambda_2| \leq \alpha$.*

**Proof.** The proof is omitted. See Haveliwala *et al.* [29] □

Using an iterative technique called power method, the algorithm aims to find the stable state vectors of the equations (4), (5) and (6). Lemma 1 shows that $S_{\mathcal{R}}$, $S_{\mathcal{C}}$ and $S_{\mathcal{E}}$ are column stochastic matrices. By construction of ${H_{\mathcal{R}}}^0, {H_{\mathcal{C}}}^0, {H_{\mathcal{E}}}^0$, we have $||{H_{\mathcal{R}}}^0||_1 = 1, ||{H_{\mathcal{C}}}^0||_1 = 1, ||{H_{\mathcal{E}}}^0||_1 = 1$. Now, by the following theorem, we show that stable state vectors for equations (4), (5) and (6) exist and they are unique.

**Theorem 1.** *Let, $A$ be an $N \times N$ column stochastic matrix and $H^0$ be an N-vector with $||H^0||_1 = 1$. For any $\alpha \in [0,1]$, there exists a stable state vector $H^s$, which satisfies the equation:*

$$H = \alpha AH + (1-\alpha)H^0 \tag{9}$$

*Furthermore, if $\alpha \in [0,1)$, then $H^s$ is unique.*

**Proof.** *Existence:* Let, $e$ be the N-vector with all entries equal to 1. Then, $e^T H = 1$ since $||H||_1 = 1$ after normalizing $H$. Now, the Equation 9 can be rewritten as:

$$\begin{aligned} H &= \alpha AH + (1-\alpha)H^0 \\ &= \alpha AH + (1-\alpha)H^0 e^T H \\ &= (\alpha A + (1-\alpha)H^0 e^T)H \\ &= MH \end{aligned}$$

where $M = \alpha A + (1-\alpha)H^0 e^T$. Let, $E$ denote $H^0 e^T$. Then, $E$ is a column stochastic matrix since its columns are all equal to $H^0$ and $||H^0||_1 = 1$. By

Lemma 3 M is a column stochastic. Then by Lemma 4, $\lambda_1 = 1$ is an eigenvalue of M. Hence, there exists an eigenvector $H^s$ corresponding to the eigenvalue $\lambda_1$ which satisfies the equation $\lambda_1 H^s = MH^s$.

*Uniqueness:* Applying the Lemma 4 to the $M$ matrix defined in the existence part, we have $|\lambda_1| = 1$ and $|\lambda_2| \leq \alpha$. If $\alpha \in [0, 1)$, then $|\lambda_1| > |\lambda_2|$. This implies that $\lambda_1$ is *the* principal eigenvalue of M and $H^s$ is the unique eigenvector corresponding to it. □

Convergence rate of power method for equations (4), (5) and (6) are determined by the eigenvalues of the $M$ matrices (as defined in Equation 8) of each equation. Convergence rate is proportional to $O(\frac{|\lambda_2|}{|\lambda_1|})$, which is $O(\alpha)$, for each equation. Therefore, choice of $\alpha$ not only adjusts the relative importance of homology and topology, but it also affects running time of the algorithm. Since $S_{\mathcal{R}}, S_{\mathcal{C}}, S_{\mathcal{E}}$ are usually large matrices, every iteration of power method is costly. The idea of reducing the number of iterations by choosing a small $\alpha$ is problematic because it degrades the accuracy of the alignment. This algorithm performs well and converges quickly with $\alpha = 0.7$.

Before the first iteration of power method in equations (4), (5) and (6), we only have pairwise similarity scores. After the first iteration, an entry ${H_{\mathcal{R}}}^1[i, j]$ of an ${H_{\mathcal{R}}}^1$ vector is set to $(1-\alpha)$ times the pairwise similarity score of $r_i, \bar{r_j}$, plus $\alpha$ times the total support supplied to $r_i, \bar{r_j}$ mapping by all mappings $r_u, \bar{r_v}$ such that both $r_i, r_u$ and $\bar{r_j}, \bar{r_v}$ are first-degree neighbors. Intuitively, the first iteration combines the pairwise similarities of entities with the topological similarity of them by considering their first degree neighbors. If we generalize this to $k$th iteration, the weight of pairwise similarity score stays to be $(1 - \alpha)$, whereas weight of total support given by $(k-t)$th degree neighbors of $r_i, \bar{r_j}$ is $\alpha^{k-t}(1-\alpha)$. That way, when the equation system converges the neighborhood topologies of mappings are thoroughly utilized without ignoring the effect of initial pairwise similarity scores. As a result, stable state vectors calculated in this manner are a good mixture of homology and topology. Hence, using these vectors for extracting the entity mappings gives us an accurate alignment of query networks.

### 2.6 Extracting the Mapping of Entities

Having combined homological and topological similarities of query metabolic networks, it only remains to extract the mapping of entities. However, since the algorithm restricts the consideration to consistent mappings this extraction by itself is still challenging. Figure 7 points out the importance of maintaining consistency of an alignment. Aligning the compounds C2' and C5 in the figure creates inconsistency since it is not supported by the backbone of the alignment created by reaction mappings.

An alignment described by the mapping $\phi$ gives the individual mappings of entities. Lets denote $\phi$ as $\phi = [\phi_{\mathcal{R}}, \phi_{\mathcal{C}}, \phi_{\mathcal{E}}]$, where $\phi_{\mathcal{R}}$, $\phi_{\mathcal{C}}$ and $\phi_{\mathcal{E}}$ are mappings for reactions, compounds and enzymes respectively.

There are three conditions that $\phi$ should satisfy to be consistent. The first one is trivially satisfied for any $\phi$ of the form $[\phi_{\mathcal{R}}, \phi_{\mathcal{C}}, \phi_{\mathcal{E}}]$, since the algorithm

beforehand distinguished each entity type. For the second condition, it is sufficient to create one-to-one mappings for each entity type. Maximum weight bipartite matching creates one-to-one mappings $\phi_{\mathcal{R}}, \phi_{\mathcal{C}}$ and $\phi_{\mathcal{E}}$, which in turn implies $\phi$ is one-to-one since intersections of compatibility classes are empty.

The difficult part of finding a consistent mapping is combining mappings of reactions, enzymes and compounds without violating the third condition. For that purpose, the method chooses a specific ordering between extraction of entity mappings. In between different ordering options, creating the mapping $\phi_{\mathcal{R}}$ comes first. A discussion for reasons of *Reactions First* ordering can be found in Ay *et al*. [23]. The $\phi_{\mathcal{R}}$ mapping is extracted by using maximum weight bipartite matching on the bipartite graph constructed by the edge weights in ${H_{\mathcal{R}}}^s$ vector. Then, using the aligned reactions and the reachability sets, the algorithm prunes the edges from the bipartite graph of compounds (enzymes) for which the corresponding compound (enzyme) pairs are inconsistent with the reaction mapping. In other words, the algorithm prunes the edge between two compounds (enzymes) $x - \bar{x}$, if there exists no other compound (enzyme) pair $y - \bar{y}$ such that $x$ is reachable from $\bar{x}$ and $y$ is reachable from $\bar{y}$ or $\bar{x}$ is reachable from $x$ and $\bar{y}$ is reachable from $y$. Pruning these edges guarantees that for any $\phi_{\mathcal{C}}$ and $\phi_{\mathcal{E}}$ extracted from the pruned bipartite graphs $\phi = [\phi_{\mathcal{R}}, \phi_{\mathcal{C}}, \phi_{\mathcal{E}}]$ is consistent.

Recall that, the aim of the method is to find a consistent alignment which maximizes the similarity score $SimP_{\phi}$. The $\phi$ defined above satisfies the consistency criteria. The next section describes $SimP_{\phi}$ and then discusses that the algorithm finds the mapping that maximizes $\phi$ that maximizes this score.

### 2.7 Similarity Score of Networks

As we present in the previous section, the algorithm guarantees to find a consistent alignment represented by the mappings of entities. One can discuss the accuracy and biological significance of the alignment by looking at the individual mappings reported. However, this requires a solid background of the specific metabolism of different organisms. To computationally evaluate the degree of similarity between networks, it is necessary to devise an accurate similarity score. Using the pairwise similarities of aligned entities, an overall similarity score between two query networks, $SimP_{\phi}$, is defined as follows:

**Definition 7.** *Let, $P = ([\mathcal{R}, \mathcal{C}, \mathcal{E}], E)$ and $\bar{P} = ([\bar{\mathcal{R}}, \bar{\mathcal{C}}, \bar{\mathcal{E}}], \bar{E})$ be two metabolic networks. Given a mapping $\phi = [\phi_{\mathcal{R}}, \phi_{\mathcal{C}}, \phi_{\mathcal{E}}]$ between entities of $P$ and $\bar{P}$, similarity of $P$ and $\bar{P}$ is calculated as:*

$$SimP_{\phi}(P, \bar{P}) = \frac{\beta}{|\phi_{\mathcal{C}}|} \sum_{\forall (c_i, \bar{c_j}) \in \phi_{\mathcal{C}}} SimC(c_i, \bar{c_j}) + \frac{(1-\beta)}{|\phi_{\mathcal{E}}|} \sum_{\forall (e_i, \bar{e_j}) \in \phi_{\mathcal{E}}} SimE(e_i, \bar{e_j}) \tag{10}$$

*where $|\phi_{\mathcal{C}}|$ and $|\phi_{\mathcal{E}}|$ denote the cardinality of corresponding mappings and $\beta \in [0, 1]$ is a parameter that adjusts the relative influence of compounds and enzymes on the alignment score.* □

Calculated as above, $SimP_\phi$ gives a score between 0 and 1 such that a bigger score implies a better alignment between networks. Using $\beta = 0.5$ prevents a bias in the score towards enzymes or compounds. One can set $\beta = 0$ to have an enzyme based similarity score or $\beta = 1$ to have a compound based similarity score. Reactions are not considered while calculating this score since reaction similarity scores are already determined by enzyme and compound similarity scores.

Having defined the network similarity score, it is necessary to show that the consistent mapping $\phi = [\phi_\mathcal{R}, \phi_\mathcal{C}, \phi_\mathcal{E}]$ found in the previous section, is the one that maximizes this score. This follows from the fact that the algorithm uses maximum weight bipartite matching on the pruned bipartite graphs of enzymes and compounds. In other words, since maximality of the total edge weights of $\phi_\mathcal{C}$ and $\phi_\mathcal{E}$ are beforehand assured by the extraction technique, their weighted sum is guaranteed to give the maximum $SimP_\phi$ value for a fixed $\beta$.

### 2.8 Complexity Analysis

Given $P = ([\mathcal{R}, \mathcal{C}, \mathcal{E}], E)$ and $\bar{P} = ([\bar{\mathcal{R}}, \bar{\mathcal{C}}, \bar{\mathcal{E}}], \bar{E})$, there are three steps that contribute to the time complexity of the method. First, the algorithm calculates pairwise similarity scores for entities in $O(|\mathcal{R}||\bar{\mathcal{R}}| + |\mathcal{C}||\bar{\mathcal{C}}| + |\mathcal{E}||\bar{\mathcal{E}}|)$ time, since calculating the similarity score of a pair is constant. Second, it creates three support matrices and uses power method for finding stable state vectors. Both creation phase and a single iteration of power method take $O(|\mathcal{R}|^2|\bar{\mathcal{R}}|^2 + |\mathcal{C}|^2|\bar{\mathcal{C}}|^2 + |\mathcal{E}|^2|\bar{\mathcal{E}}|^2)$ time. In all experiments power method converges in small number of iterations ($<$15) for each entity type. Hence, the total cost of finding stable state vectors is also, $O(|\mathcal{R}|^2|\bar{\mathcal{R}}|^2 + |\mathcal{C}|^2|\bar{\mathcal{C}}|^2 + |\mathcal{E}|^2|\bar{\mathcal{E}}|^2)$. The last step is extracting mappings from stable state vectors by using maximum weight bipartite matching. This step takes $O(min(|\mathcal{R}|, |\bar{\mathcal{R}}|)|\mathcal{R}||\bar{\mathcal{R}}| + min(|\mathcal{C}|, |\bar{\mathcal{C}}|)|\mathcal{C}||\bar{\mathcal{C}}| + min(|\mathcal{E}|, |\bar{\mathcal{E}}|)|\mathcal{E}||\bar{\mathcal{E}}|)$ time in total. Therefore, the complexity of the algorithm is dominated by the power method part which results in an overall time complexity of $O(|\mathcal{R}|^2|\bar{\mathcal{R}}|^2 + |\mathcal{C}|^2|\bar{\mathcal{C}}|^2 + |\mathcal{E}|^2|\bar{\mathcal{E}}|^2)$.

## 3 Metabolic Network Alignment with One-to-Many Mappings

The algorithm we discussed in the previous section addressed the first challenge in Section 1. In this section, we present a new algorithm called **SubMAP**, which addresses challenge 2 as well as challenge 1. In other words, this algorithm considers subnetwork mappings of all types of biological entities in metabolic networks. This section begins by extending the notation defined in Section 2. Then, we formally state the network alignment problem by considering the second challenge and describe the SubMAP algorithm in detail.

**Extended Notation.** In addition to the notation we used so far, we define a *subnetwork* of a network. A subnetwork is defined as a reaction subset of a network such that the induced undirected graph of this subset is connected. Let $R_i \subseteq V$ be one such subnetwork of $P$. Let $\mathcal{R}_k$ be the set of all subnetworks

of $P$ that have at most $k$ reactions, denoted as $\mathcal{R}_k = \{R_1, R_2, \ldots, R_{N_k}\}$ where $|R_i| \leq k$ for all $i \in [1, N_k]$. Here, $|R_i|$ denotes the cardinality of the reaction set $R_i$. For instance, for the network $\bar{P}$ in our running example, the sets of subnetworks for $k = 1, 2, 3, 4$ are:

- $\bar{\mathcal{R}}_1 = \bar{\mathcal{R}} = \{\bar{r_1}, \bar{r_2}, \bar{r_3}, \bar{r_4}\}$
- $\bar{\mathcal{R}}_2 = \bar{\mathcal{R}}_1 \cup \{\{\bar{r_1}, \bar{r_2}\}, \{\bar{r_2}, \bar{r_3}\}, \{\bar{r_2}, \bar{r_4}\}\}$
- $\bar{\mathcal{R}}_3 = \bar{\mathcal{R}}_2 \cup \{\{\bar{r_1}, \bar{r_2}, \bar{r_3}\}, \{\bar{r_1}, \bar{r_2}, \bar{r_4}\}, \{\bar{r_2}, \bar{r_3}, \bar{r_4}\}\}$
- $\bar{\mathcal{R}}_4 = \bar{\mathcal{R}}_3 \cup \{\{\bar{r_1}, \bar{r_2}, \bar{r_3}, \bar{r_4}\}\}$

Using this notation, we define a binary relation that maps a reaction of a query network to a subnetwork of the other as follows:

**Definition 1.** *Let $P$ and $\bar{P}$ be two networks and $k$ be a positive integer. Also, let $\mathcal{R}_k = \{R_1, R_2, \ldots, R_{N_k}\}$ and $\bar{\mathcal{R}}_k = \{\bar{R_1}, \bar{R_2}, \ldots, \bar{R_{M_k}}\}$ be the sets of subnetworks with size at most $k$ of $P$ and $\bar{P}$ respectively. We define a binary relation $\varphi$ between $\mathcal{R}_k$ and $\bar{\mathcal{R}}_k$ that allows one-to-many reaction mappings as $\varphi : \varphi \subseteq \varphi_k = (\mathcal{R}_1 \times \bar{\mathcal{R}}_k) \cup (\mathcal{R}_k \times \bar{\mathcal{R}}_1)$.* □

We denote the number of reactions of $P$ and $\bar{P}$ with $n$ and $m$ respectively. The number of all possible one-to-many mappings between $P$ and $\bar{P}$ is:

$$|\varphi_k| = nM_k + mN_k - nm \tag{11}$$

The *alignment* of $P$ and $\bar{P}$ is a binary relation $\varphi$ that is a subset of all these possible mappings and consistent. Since allowing subnetwork mappings introduces new types of conflicting mappings, in the following, we extend the definition of conflict and redefine consistency of an alignment.

Recall that for a mapping $(R_i, \bar{R}_j) \in \varphi$ one of the $R_i$ or $\bar{R}_j$ can contain more than one reaction. Reporting this mapping as a part of our alignment implies that all the reactions of the subnetwork with multiple reactions are aligned to a single reaction of the other. To have a *consistent alignment* none of the reactions of these subnetworks can be included in any other mapping. In a network alignment which allows one-to-many mappings, we define the term *conflict* as follows:

**Definition 2.** *Let $\varphi$ be a binary relation and $R_i, R_u \in \mathcal{R}_k$ and $\bar{R}_j, \bar{R}_v \in \bar{\mathcal{R}}_k$. The distinct pairs $(R_i, \bar{R}_j) \in \varphi$ and $(R_u, \bar{R}_v) \in \varphi$ **conflict** if and only if $(R_i \cap R_u) \cup (\bar{R}_j \cap \bar{R}_v) \neq \varnothing$.* □

We also need to reconsider the similarity measure presented in Section 2.3 for subnetworks. In Section 3.3, we describe the generalization the scoring scheme to account for one-to-many mappings. Next, we state the alignment problem with subnetworks formally.

**Problem Formulation:** Given $k$ and two networks $P$ and $\bar{P}$, let $\mathcal{R}_k$ and $\bar{\mathcal{R}}_k$ be the sets of subnetworks with size at most $k$ of $P$ and $\bar{P}$ respectively. We want to find a *consistent* binary relation $\varphi \subseteq (\mathcal{R}_1 \times \bar{\mathcal{R}}_k) \cup (\mathcal{R}_k \times \bar{\mathcal{R}}_1)$ that *maximizes* the summation of the similarity scores of the aligned subnetworks.

In the following, we present the SubMAP algorithm. Section 3.1 and 3.2 discuss homological and topological similarities respectively. Section 3.3 describes the eigenvalue formulation that combines these similarities and explains the extraction of subnetwork mappings.

**Table 2.** Extended table of frequently used symbols in Section 3

| | |
|---|---|
| $k$ | Parameter for the largest subnetwork size |
| $R_i$, $\bar{R}_j$ | Subnetworks of query networks |
| $\mathcal{R}_k$, $\bar{\mathcal{R}}_k$ | Sets of all subnetworks with size at most $k$ |
| $n$, $m$ | Numbers of the reactions in query networks |
| $N_k$, $M_k$ | Numbers of all subnetworks of size at most $k$ |
| $I_i$, $O_i$,$\mathcal{E}_i$ | Set of input compounds, output compounds and enzymes of subnetwork $R_i$ |
| $\varphi$ | Relation which represents the alignment with subnetwork mappings |
| $\varphi_k$ | Set of all possible one-to-many mappings for a given k |
| $\lvert\varphi_k\rvert$ | Number of all possible one-to-many mappings for a given k |
| $S$ | Support matrix for reaction subnetwork mappings |
| $G_c$ | Conflict graph |

### 3.1 Homological Similarity of Subnetworks

Recall that the relation $\varphi$ maps a reaction to a subnetwork that can contain multiple reactions. This necessitates computing the similarity between reaction sets. In Section 2.3, we describe how to compute similarity between two reactions using similarities between their compounds ($SimC$) and enzymes ($SimE$). Here we describe how SubMAP extends this similarity to account for reaction sets. To do this, the method first constructs three sets for both reaction sets. These are the union of

1. The input compounds ($I_i$)
2. The output compounds ($O_i$)
3. The enzymes ($\mathcal{E}_i$)

of the reactions in each subnetwork $R_i$. For instance, in Figure 4 if we take the upper path as the subnetwork $R_i$, then $\mathcal{E}_i = \{2.3.1.117, 2.6.1.17, 3.5.1.8\}$.

Next, SubMAP computes the similarity of each of these three set pairs and combine them using weights (i.e., non-negative real numbers) to calculate the homological similarity of the two reaction sets. Let $MWBM(A, B, SimX)$ denote the similarity between two sets $A$ and $B$ with respect to the similarity score $SimX$ (i.e., $SimE$ or $SimC$), where $MWBM$ is calculated as the sum of the similarities of the pairs returned by their maximum weight bipartite matching. Similar to Section 2, let $\gamma_i, \gamma_o, \gamma_e$ denote the relative weights of the similarities of enzymes, input and output compounds respectively. Similarity of the reaction sets $R_i$ and $\bar{R}_j$ are

$$\begin{aligned} SimRSet(R_i, \bar{R}_j) = {} & \gamma_i MWBM(I_i, \bar{I}_j, SimC) \\ & + \gamma_o MWBM(O_i, \bar{O}_j, SimC) \\ & + \gamma_e MWBM(\mathcal{E}_i, \bar{\mathcal{E}}_j, SimE) \end{aligned} \tag{12}$$

The algorithm calculates $SimRSet$ for all possible one-to-many mappings between the subnetworks of two networks. So, it does this calculation $|\varphi_k|$ times in total. This way, the homological similarities between all possible subnetwork

mappings are assessed. Even though this scoring can be considered a good measure of similarity for subnetwork pairs, relying solely on it ignores the topological similarity. Next section focuses on this issue.

### 3.2 Topological Similarity of Subnetworks

SubMAP algorithm follows the intuition of the algorithm in Section 2 and the IsoRank algorithm [21] that if the subnetwork $R_i$ is to be mapped $\bar{R}_j$, then their neighbors in the corresponding networks should also be similar. With this motivation, it utilizes the topological similarity to favor mappings of subnetworks that induce similar topologies. In SubMAP, the authors do this by extending the formulation of the algorithm described in Section 2 to account for subnetwork mappings. They first expand the *neighborhood* definition of reactions to reaction subnetworks. Then, they introduce the notion of *support* between two subnetwork mappings.

**Definition 3.** *Let $R_i$, $R_u \in \mathcal{R}_k$. Then, $R_u$ is a **forward neighbor** of $R_i$ ($R_u \in FN(R_i)$) if and only if there exists $r_a \in R_i$ and $r_b \in R_u$ such that $r_b$ is a forward neighbor of $r_a$ or $R_i \cap R_u \neq \varnothing$. $R_i$ is a **backward neighbor** of $R_u$ ($R_i \in BN(R_u)$) if and only if $R_u$ is a forward neighbor of $R_i$.* □

**Definition 4.** *Let $R_i, R_u \in \mathcal{R}_k$ and $\bar{R}_j, \bar{R}_v \in \bar{\mathcal{R}}_k$. The mapping $(R_i, \bar{R}_j)$ **supports** the mapping $(R_u, \bar{R}_v)$ if and only if both $R_j \in FN(R_i)$ and $\bar{R}_v \in FN(\bar{R}_u)$ or both $R_j \in BN(R_i)$ and $\bar{R}_v \in BN(\bar{R}_u)$.* □

**Definition 5.** *Let $P$, $\bar{P}$ be two networks and $\mathcal{R}_k$, $\bar{\mathcal{R}}_k$ be the sets of their subnetworks that have at most $k$ reactions. **The support matrix** $S$ is a $|\varphi_k| \times |\varphi_k|$ matrix with each entry $S[i,j][u,v]$ identifying the fraction of the total support provided by $(R_u, \bar{R}_v)$ mapping to $(R_i, \bar{R}_j)$ mapping. Let $\mathrm{N}(u,v) = |BN(R_u)|\,|BN(\bar{R}_v)| + |FN(R_u)|\,|FN(\bar{R}_v)|$ denote the number of all possible mappings between backward neighbors of $R_u$ and $\bar{R}_v$ plus the ones between their forward neighbors. Then, each entry of $S$ is computed as:*

$$S[i,j][u,v] = \begin{cases} \frac{1}{\mathrm{N}(u,v)} & \text{if } (R_u, \bar{R}_v) \text{ supports } (R_i, \bar{R}_j) \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

□

Definition 3 defines forward and backward neighbors for subnetworks. Definition 4 states that the mapping of $R_i$ to $\bar{R}_j$ favors all possible mappings of forward (backward) neighbors of $R_i$ to those of $\bar{R}_j$. Finally, Definition 5 describes how to distribute the support of a mapping to neighborhood mappings. To see it on our running example, let us consider the case when $k = 2$ and focus on the mapping $(\{r_2\}, \{\bar{r_2}, \bar{r_3}\})$. By Definition 3 we have:

- $BN(\{r_2\})$=$\{\{r_1\},\{r_1, r_2\}\}$, $|BN(\{r_2\})| = 2$
- $BN(\{\bar{r_2}, \bar{r_3}\})$=$\{\{\bar{r_1}\},\{\bar{r_1}, \bar{r_2}\}\}$, $|BN(\{\bar{r_2}, \bar{r_3}\})| = 2$
- $FN(\{r_2\})$=$\{\{r_3\}, \{r_2, r_3\}\}$, $|FN(\{r_2\})| = 2$
- $FN(\{\bar{r_2}, \bar{r_3}\})$=$\{\{\bar{r_3}\}, \{\bar{r_4}\}, \{\bar{r_2}, \bar{r_4}\}\}$, $|FN(\{\bar{r_2}, \bar{r_3}\})| = 3$

Then, by Definition 5 we distribute the support of the mapping $(\{r_2\}, \{\bar{r_2}, \bar{r_3}\})$ to $2 \times 2 + 2 \times 3 = 10$ other mappings by placing $1/10$ in the corresponding entries of $S$. Namely, these mappings are $(\{r_1\}, \{\bar{r_1}\})$, $(\{r_1\}, \{\bar{r_1}, \bar{r_2}\})$, $(\{r_1, r_2\}, \{\bar{r_1}\})$, $(\{r_1, r_2\}, \{\bar{r_1}, \bar{r_2}\})$, $(\{r_3\}, \{\bar{r_3}\})$, $(\{r_3\}, \{\bar{r_4}\})$, $(\{r_3\}, \{\bar{r_2}, \bar{r_4}\})$, $(\{r_2, r_3\}, \{\bar{r_3}\})$, $(\{r_2, r_3\}, \{\bar{r_4}\})$ and $(\{r_2, r_3\}, \{\bar{r_2}, \bar{r_4}\})$.

There can be cases when one mapping does not provide support to any others. In such cases, its support equally distributed to all possible mappings ($|\varphi_k|$). Notice that, by construction, the entries in each column of $S$ sums up to 1. We will not recount the properties of support matrix as they are listed in Section 2.

At this point, support matrix calculation can become an issue if not properly handled. The trivial way of creating support matrix is to check each mapping against all the others to calculate the support values. However, such an exhaustive strategy will require computing a huge matrix $S$ of size $|\varphi_k| \times |\varphi_k|$. Since the creation of $S$ will incur prohibitive computational costs, SubMAP does not construct this matrix literally. Instead, for each mapping $(R_i, \bar{R_j})$, it uses the sets $FN(R_i)$, $FN(\bar{R_j})$ and $BN(R_i)$, $BN(\bar{R_j})$ to generate only the pairs supported by $(R_i, \bar{R_j})$. In other words, it maintains the support matrix $S$ in sparse matrix form.

### 3.3 Combining Homology and Topology

As discussed in Section 2, both the homological similarities of subnetworks and their topological organization are important factors in an accurate alignment. Here we describe the combination of these two similarities in SubMAP which is very similar to the one described in Section 2.5.

Let $k$ be a given parameter and $P$, $\bar{P}$ be two networks with connected subnetwork sets $\mathcal{R}_k = \{R_1, R_2, \ldots, R_{N_k}\}$ and $\bar{\mathcal{R}_k} = \{\bar{R_1}, \bar{R_2}, \ldots, \bar{R_{M_k}}\}$ respectively. The column vector $H$ denotes the homological similarity of all subnetwork pairs which is size $|\varphi_k|$. Each entry of $H$ denotes the homological similarity between two subnetworks one from each network which corresponds to a mapping.

Let $S$ be the $|\varphi_k| \times |\varphi_k|$ support matrix constructed as described in Section 3.2. Given a parameter $\alpha \in [0, 1]$ to adjust the relative weights of homology and topology, combining these two through power method iterations is done as follows:

$$H^{i+1} = \alpha S H^i + (1 - \alpha) H^0 \tag{14}$$

SubMAP iterates this equation until $H^{i+1} = H^i$ (i.e., it converges). Please refer to Section 2.5 for more details of this process.

### 3.4 Extracting Subnetwork Mappings

Recall that SubMAP aims to find an alignment that *maximizes* the summation of the similarity scores defined by $H^i$ while preserving the consistency between different mappings. Since SubMAP allows one-to-many mappings, extraction of a mapping that maximizes the alignment score is NP-hard. Here we first show that it is in NP-hard by a reduction from *the maximum weight independent set (MWIS)* problem in bounded degree graphs. MWIS problem, even for graphs

with largest degree 3, is NP-hard [30] and there is no constant factor approximation to the optimal solution in polynomial time [31,32]. We then describe how to construct a conflict graph from the alignment problem and employ a vertex-selection heuristic in order to extract the set of mappings that generates the alignment.

**Theorem 2.** (FINDING THE BEST ALIGNMENT IS NP-HARD) *Let $P$ and $\bar{P}$ be two networks with reaction sets $\{r_1, \ldots, r_n\}$ and $\{\bar{r_1}, \ldots, \bar{r_m}\}$ respectively. Let $\mathcal{R} = \{R_1, R_2, \ldots, R_N\}$ and $\bar{\mathcal{R}} = \{\bar{R_1}, \bar{R_2}, \ldots, \bar{R_M}\}$ be all possible reaction subsets of $P$ and $\bar{P}$ with size at most a given positive integer $k$. Also, let $w : (\mathcal{R}, \bar{\mathcal{R}}) \rightarrow [0, 1] \cup \{-\infty\}$ be a similarity function defining the score for each mapping and the conflicts between mappings be defined according to Definition 2. Then, finding a set of mappings (i.e., alignment) that maximizes the sum of mapping scores (i.e., alignment score) and has no conflicting pairs is NP-hard.*

**Proof.** We prove the NP-hardness of finding the best alignment by a reduction from the MWIS problem in bounded degree graphs. Let $G = (V, E, w)$ be a vertex weighted undirected graph with largest degree $k-1$ (i.e., $k = \max_{i=1,\ldots,|V|} deg(v_i)+1$). Let us set $n = |V|+|E|$, $m = |V|$. We will construct two hypothetical networks $P$ and $\bar{P}$ through a polynomial time reduction such that their best alignment is equivalent to the MWIS of $G$.

REDUCTION. Let us denote the networks $P$ and $\bar{P}$ as $P = (V_1, I_1)$ and $\bar{P} = (V_2, I_2)$. Here, $V_i$ and $I_i$ ($i \in \{1, 2\}$) denote the set of reactions (vertices) and interactions (edges) respectively.

We initialize $V_1$, $V_2$, $I_1$ and $I_2$ as the empty set. We then insert a vertex $r_i$ in $V_1$ for each $v_i \in V$. Similarly, we insert a vertex $\bar{r_i}$ in $V_2$ for each $v_i \in V$. At this point we have completed the construction of $\bar{P}$ but not $P$. We continue by inserting a new vertex in $V_1$ for each edge $e \in I$. Thus, each vertex in $V_1$ corresponds to either a vertex or an edge in $G$ while each vertex in $V_2$ corresponds to a vertex in $G$.

Next, we populate $I_1$. Let $r_i$ and $r_j$ be two vertices in $V_1$ which correspond to a vertex $v$ in $G$ and to an edge $e$ in $G$ respectively. We include an edge between $r_i$ and $r_j$ in $I_1$ if $e$ has $v$ at one of its ends in $G$. This completes the construction of $P$.

Figure 11 illustrates the construction of the two hypothetical networks $P$ and $\bar{P}$ from a simple instance of $G$. Figure 11(a) shows a sample graph $G$ with four vertices. Figure 11(b) presents the resulting networks. We do not show the weights of the vertices of $G$ to simplify Figure 11(a).

Following from the construction, we ensure that there is a connected subnetwork in $P$ that contains the reactions corresponding to each vertex and its edges in $G$. For instance, in Figure 11(a), the vertex labeled as "1" has two edges labeled as "a" and "b". In Figure 11(b) there are three reactions that have the labels "1", "a" and "b" and they make up a (connected) subnetwork. Thus, the set of subnetworks of $P$ with size up to $k$ is guaranteed to contain each vertex of $G$ jointly with all of its edges. Figure 11(c) demonstrates this. The vertices

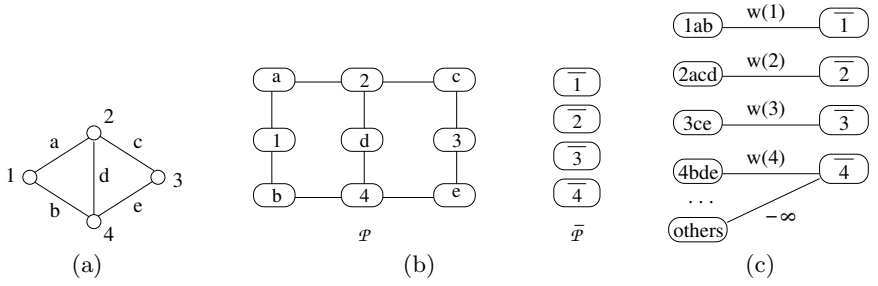



**Fig. 11.** An illustrative example for the reduction from the MWIS problem to the metabolic network alignment problem. (a) A vertex-weighted graph $G$ that is an input to the MWIS problem. Four vertices labeled from "1" to "4" have weights from w(1) to w(4) respectively. The edges are labeled from "a" to "e" and are unweighted and undirected. (b) Two networks $P$ and $\bar{P}$ constructed from $G$. We create one vertex for each vertex of $G$ in both network $P$ and $\bar{P}$. Then, for $P$ we add a vertex labeled with a letter for each edge of $G$ and add edges from it to the vertices on its both ends in $G$. In order to simplify the figure, we match the label of each vertex in $P$ with that of its corresponding vertex or edge in $G$. Similarly, we match the label of each vertex in $\bar{P}$ with that of its corresponding vertex in $G$. (c) The assignment of similarity scores for subnetwork pairs one from $P$ and the other from $\bar{P}$. Each vertex here shows a subnetwork from $P$ or $\bar{P}$. The label of each vertex lists the vertices contained in that subnetwork. For instance, label "1ab" indicates the subnetwork of $P$ that consists of the three vertices labeled as "1", "a" and "b". The edge weights show the similarity of the two subnetworks corresponding to the two vertices at its end points.

on the left side are the subnetworks of $P$ and those on the right side are the subnetworks of $\bar{P}$.

We complete the reduction by assigning similarities to subnetwork pairs one from $P$ and the other from $\bar{P}$. The similarities we assign correspond to the entries of the column vector $H$ described in Section 3.3.

Let $R_i$ be a subnetwork of $P$ that corresponds to a vertex $v$ in $G$ and all edges of $G$ which have $v$ on one end. In Figure 11(b), the subnetwork that contains the reactions labeled "1", "a" and "b" is an example to such $R_i$. Also, let $\bar{R}_i$ be the subnetwork in $\bar{P}$ that corresponds to vertex $v$ as well. We assign the similarity of $R_i$ and $\bar{R}_i$ as the weight of vertex $v$ (i.e., $w(v)$). We repeat this process for all $v \in V$. We assign the similarity between all remaining pairs of subnetworks ($R_i$, $\bar{R}_j$) as $-\infty$. Figure 11(c) depicts the assignment of similarities for the networks in Figure 11(b).

Correctness of reduction. We need to address two issues to prove the correctness of the reduction.

– *Cost of reduction.* For a given vertex weighted graph $G$, we can reduce the MWIS problem on $G$ to finding best alignment problem in polynomial time. This is because we create one reaction for each edge and two reactions for

each vertex in $G$. We also create two interactions for each edge in $G$. Thus, we conclude that the reduction is polynomial in the size of $G$.

– *Equivalence of the result.* Next, we prove that the optimal alignment of $P$ and $\bar{P}$ produces the optimal solution to the MWIS problem on $G$.
An alignment is a subset of subnetwork pairs from one from $P$ and the other from $\bar{P}$. By construction, $\bar{P}$ contains only subnetworks of size one. Each subnetwork of $\bar{P}$ corresponds to a vertex in $G$. We claim that the vertices of $G$ corresponding to the subnetworks of $\bar{P}$ in the optimal alignment constitute the MWIS of $G$.

Clearly, the optimal alignment can not contain a subnetwork pair whose similarity is $-\infty$. This is because it is possible to choose an arbitrary subnetwork pair that has a positive score. Also, the optimal alignment can not contain two overlapping subnetworks from the same network as they will conflict with each other. By construction, two subnetworks from $P$ conflict only if they share a common reaction for the same vertex or edge in $G$. For instance the subnetworks labeled "1ab" and "2acd" in Figure 11(c) conflict since they both contain "a" in Figure 11(b). The subnetworks in the optimal alignment can not contain such reactions. Hence, the optimal alignment constitute an independent set in $G$. The score of the alignment is the sum of the weights of the corresponding vertices in $G$. Therefore, we conclude that by maximizing the alignment score $P$ and $\bar{P}$, we optimally solve the MWIS problem for $G$. □

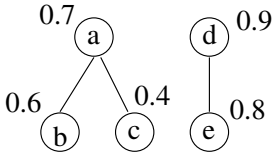
We demonstrate the result on the hypothetical example in Figure 11. Assume that all the vertices in Figure 11(a) have the same weight $w$. The MWIS of the graph in Figure 11(a) is $\{1, 3\}$. This is because nodes 2 and 4 conflict with all the remaining nodes. The optimal alignment of $P$ and $\bar{P}$ contains the following set of subnetwork pairs $\{(1ab, \bar{1}), (3ce, \bar{3})\}$ and has an alignment score of $2w$. This is because the remaining subnetwork pairs either conflict with each other or have a minus infinity ($-\infty$) score. Since the optimal alignment contains nodes $\bar{1}$ and $\bar{3}$, it suggests that the MWIS of $G$ is $\{1, 3\}$.

Since extracting the best alignment is NP-hard, SubMAP uses a heuristic strategy to tackle this problem. In the first step, the scores of mappings represented by $H^i$ and the definition of conflict (see Definition 2) creates a vertex weighted undirected graph $G_c = (V_c, I_c, w)$, namely *the conflict graph* as follows. Each subnetwork pair $(R_i, \bar{R}_j)$ corresponds to a vertex in $V_c$. The weight of each vertex $a = (R_i, \bar{R}_j) \in V_c$ (i.e., $w(a)$) is the similarity between $R_i$ and $\bar{R}_j$ as computed in $H^i$. An undirected edge between two vertices $a = (R_i, \bar{R}_j)$ and $b = (R_u, \bar{R}_v)$ exists if $(R_i \cap R_u) \cup (\bar{R}_j \cap \bar{R}_v) \neq \varnothing$ (i.e., $a$ and $b$ conflict). For instance, in Figure 12 there is an edge between $a$ and $b$ representing the fact that they conflict since reaction $r_1$ is common to both $a$ and $b$.

The second step is the greedy vertex-selection strategy which is adopted from Sakai *et al.* [33] in order to extract the MWIS of $G_c$ as the alignment. Let $N(v)$ denote the set of vertices that are connected to $v \in V_c$. At each iteration of this algorithm, this heuristic picks a vertex $v$ that maximizes $f(v) = \sum_{\forall u_i \in N(v)} \frac{w(v)}{w(u_i)}$. This strategy implies that a vertex is more likely to be picked if the mapping it

| | Subnetwork 1 ($\mathcal{P}$) | Subnetwork ($\bar{\mathcal{P}}$) | $H^i$ |
|---|---|---|---|
| *a*: | $R_1 = \{r_1, r_2\}$ | $\bar{R}_1 = \{\bar{r_1}\}$ | 0.7 |
| *b*: | $R_2 = \{r_1\}$ | $\bar{R}_2 = \{\bar{r_2}\}$ | 0.6 |
| *c*: | $R_3 = \{r_3\}$ | $\bar{R}_3 = \{\bar{r_1}\}$ | 0.4 |
| *d*: | $R_4 = \{r_4\}$ | $\bar{R}_4 = \{\bar{r_3}, \bar{r_4}, \bar{r_5}\}$ | 0.9 |
| *e*: | $R_5 = \{r_4, r_5\}$ | $\bar{R}_5 = \{\bar{r_5}\}$ | 0.8 |

(a)




(b)

**Fig. 12.** (a) Each row corresponds to a possible mapping between subnetworks from two hypothetical metabolic networks. The first column is the unique label for each mapping. Second and third columns are the reactions in the two subnetworks that can be mapped. The last column is the similarity between two subnetworks. (b) The conflict graph $G_c$ for the mappings in (a).

represents has large similarity score and conflicts with a small number of other mappings with small similarity scores. After picking a vertex $v$, it puts $v$ into the resulting set and removes $v$ and all the vertices connected to it from $G_c$ (i.e., $v \cup N(v)$). It also removes all the edges incident to at least one of the removed vertices. When there are no more vertices to remove from $G_c$, the result set contains a maximal weight independent set. For the alignment problem, this vertex set corresponds to a list of non-conflicting subnetwork mappings. As an example, in Figure 12, $d$ is the first vertex to be picked. Then, we remove $d$ and $e \in N(d)$ from the graph and insert $d$ to the result set. Next, we pick the vertex $b$ as $f(b) = \frac{0.6}{0.7} > f(a) = \frac{0.7}{0.6+0.4} > f(c) = \frac{0.4}{0.7}$. We remove $b$ and $a \in N(b)$ and include $b$ in the result set. Finally, only $c$ is left and taking it into the result set, the alignment is the mappings $b = (r_1, \bar{r_2})$, $c = (r_3, \bar{r_1})$ and $d = (r_4, \{\bar{r_3}, \bar{r_4}, \bar{r_5}\})$.

## 4 Significance of Network Alignment

An accurate alignment should reveal functionally similar entities or paths between different networks. More specifically, it is desirable to match the entities that can substitute each other or the paths that serve similar functions. Identifying these functionally similar parts of networks is important and useful for various applications. Some examples are, metabolic reconstruction of newly sequenced organisms [5], identification of network holes [34] and identification of drug targets [11,35].

### 4.1 Identification of Alternative Entities

Table 3 illustrates four cases in which the first algorithm we described successfully found the alternative enzymes with the corresponding reaction mappings. For instance, there are two different reactions that generate Asparagine (Asn) from Aspartate (Asp) as seen in Table 3. One is catalyzed by Aspartate ammonia ligase (EC:6.3.1.1) and uses Ammonium ($NH_3$) directly, whereas the other

**Table 3.** *Alternative enzymes that catalyze the formation of a common product using different compounds.* [1]Pathways: 00620-Pyruvate metabolism, 00252-Alanine and aspartate metabolism, 00860-Porphyrin and chlorophyll metabolism. [2]Organism pairs that are compared. [3]KEGG numbers of aligned reaction pairs. [4]EC numbers of aligned enzyme pairs. [5] Aligned compounds pairs are put in the same column. Common target products are underlined. Alternative input compounds are shown in **bold**. Abbreviations of compounds: MAL, malate; FAD, Flavin adenine dinucleotide; OAA, oxaloacetate; NAD, Nicotinamide adenine dinucleotide; Pi, Orthophosphate; PEP, phosphoenolpyruvate; Asp, L-Aspartate; Asn, L-Aspargine; Gln, L-Glutamine; PPi, Pyrophosphate; Glu, L-Glutamate; AMP, Adenosine 5-monophosphate; CPP, coproporphyrinogen III; PPHG, protoporphyrinogen; SAM, S-adenosylmethionine; Met, L-Methionine.

| Id[1] | Org[2] | Reaction | | |
|---|---|---|---|---|
| | | R. Id[3] | Enzyme[4] | Compounds[5] |
| 620 | *sau* | 01257 | 1.1.1.96 | MAL + ***FAD*** → OAA + $FADH_2$ |
| | *hsa* | 00342 | 1.1.1.37 | MAL + ***NAD*** → OAA + NADH |
| 620 | *ath* | 00345 | 4.1.1.31 | OAA + ***Pi*** → PEP + $CO_2$ |
| | *sau* | 00341 | 4.1.1.49 | OAA + ***ATP*** → PEP + $CO_2$ + ADP |
| 252 | *chy* | 00578 | 6.3.5.4 | Asp + ATP + ***Gln*** → Asn + AMP + PPi |
| | *cpv* | 00483 | 6.3.1.1 | Asp + ATP + $\boldsymbol{NH_3}$ → Asn + AMP + Glu |
| 860 | *sau* | 06895 | 1.3.99.22 | CPP + $\boldsymbol{O_2}$ → PPHG + $CO_2$ |
| | *hsa* | 03220 | 1.3.3.3 | CPP + ***SAM*** → PPHG + $CO_2$ + Met |

is catalyzed by transaminase (EC:6.3.5.4) that transfers the amino group from Glutamine (Gln). The alignment results for the other three examples in Table 3 are also consistent with the results in Kim *et al*. [36].

### 4.2 Identification of Alternative Subnetworks

The first row of Table 4 corresponds to alternative subnetworks in Figure 13(a) (also in Figure 4). The reaction R07613 represents the top path in Figure 13(a) that plants and *Chlamydia trachomatis* use to produce LL-2,6- Diaminopimelate from 2,3,4,5- Tetrahydrodipicolinate. The SubMAP algorithm discovers and reports this path as a shortcut on the L-Lysine synthesis path for plants and *C.trachomatis* which is not present in *H.sapiens* or *E.coli* [26,27]. Also, Watanabe *et al.* [26] suggests that since humans lack the catalyzer of the reaction R07613, namely LL-DAP aminotransferase (EC:2.6.1.83), this is an attractive target for the development of new drugs (antibiotics and herbicides). Moreover, the alignment of the Lysine biosynthesis networks of *H.sapiens* and *A.thaliana* (a plant) reveals the reaction R07613 of *A.thaliana* is the alternative of three reactions that *H.sapiens* has to use to transform 2,3,4,5- Tetrahydrodipicolinate to LL-2,6- Diaminopimelate (R02734, R04365, R04475).

Another interesting example is the second row that is extracted from the same alignment described above. In this case, the three reactions that can independently produce L-Lysine for *A.thaliana* are aligned to the only reaction that produces L-Lysine for *H.sapiens* (Figure 13(b)). R00451 is common to both

**Table 4.** Alternative subnetworks that produce same or similar output compounds from the same or similar input compounds in different organisms. [1] Main input compound utilized by the given set of reactions. [2] Main output compound produced by the given set of reactions. [3] Reactions mappings that corresponds to alternative paths. Reactions are represented by their KEGG identifiers.

| Pathway | Organisms | Input Comp.[1] | Output Comp.[2] | Reaction Mappings[3] |
|---|---|---|---|---|
| Lysine biosynthesis | *A.thaliana*<br>*H.Sapiens* | 2,3,4,5-Tetra-hydrodipico. | LL-2,6-Di-aminopimelate | **R07613 ⇔ R02734 + R04365 + R04475** |
| Lysine biosynthesis | *A.thaliana*<br>*H.sapiens* | L-Saccharo.<br>meso-2,6-Di. | L-Lysine | **R00451 + R00715 + R00716 ⇔ R00451** |
| Pyruvate metabolism | *E.coli*<br>*H.sapiens* | Pyruvate | Oxaloacetate | **R00199 + R00345 ⇔ R00344** |
| Pyruvate metabolism | *E.coli*<br>*H.sapiens* | Oxaloacetate | Phosphoenol-pyruvate | **R00341 ⇔ R00431 + R00726** |
| Pyruvate metabolism | *T.acidophilum*<br>*A.tumefaciens* | Pyruvate | Acetyl-CoA | **R01196 ⇔ R00472 + R00216 + R01257** |
| Glycine,serine, threonine met. | *H.sapiens*<br>*R.norvegicus* | Glycine | Serine<br>L-Threonine | **R00945 ⇔ R00751 + R00945 + R06171** |
| Fructose and mannose met. | *E.coli*<br>*H.sapiens* | L-Fucose | L-Fucose 1-p<br>L-Fuculose 1-p | **R03163 + R03241 ⇔ R03161** |
| Citrate cycle | *S.aureus N315*<br>*S.aureus COL* | Isocitrate | 2-Oxoglutarate | **R00268 + R01899 ⇔ R00709** |
| Citrate cycle | *H.sapiens*<br>*A.tumefaciens* | Succinate | Succinyl-CoA | **R00432 + R00727 ⇔ R00405** |
| Citrate cycle | *H.sapiens*<br>*A.tumefaciens* | Isocitrate<br>Citrate | 2-Oxoglutarate<br>Oxaloacetate | **R00709 ⇔ R00362** |

organisms and it utilizes meso-2,6- Diaminopimelate to produce L-Lysine. The reactions R00715 and R00716 take place and produce L-Lysine in *A.thaliana* in the presence of L-Saccharopine [37].

For the alignment of Pyruvate metabolisms of *E.coli* and *H.sapiens*, the third and fourth rows show two mappings that are found by SubMAP. The first one maps the two step process in *E.coli* that first converts Pyruvate to Orthophosphate (R00199) and then Orthophosphate to Oxaloacetate (R00345) to the single reaction that directly produces Oxaloacetate from Pyruvate (R00344) in *H.sapiens* (Figure 13(c)). The second one shows another mapping in which a single reaction of *E.coli* is replaced by two reactions of *H.sapiens* (Figure 13(d)). The first two rows for Citrate cycle also report similar mappings for other organism pairs (Figure 13(e)).

Note that in nature, there are numerous examples of a reaction replacing a number of reactions. In order to be able to discover such replacements, the alignment algorithm needs to allow one-to-many mappings. Also, if we look at the EC numbers of the enzymes catalyzing these reactions (1.1.1.41 and 4.1.3.6) their similarity is zero (see Information content enzyme similarity [23]). If we were to consider only the homological similarities, these two reactions could not have been mapped to each other. However, both these reactions are the neighbors of two other reactions R01325 and R01900 that are present in both organisms. The mappings of R01325 to R01325 and R01900 to R01900 support the mapping of their neighbors R00709 to R00362. Therefore, by incorporating the topological similarity SubMAP is able to find meaningful mappings with similar topologies and distinct homologies.

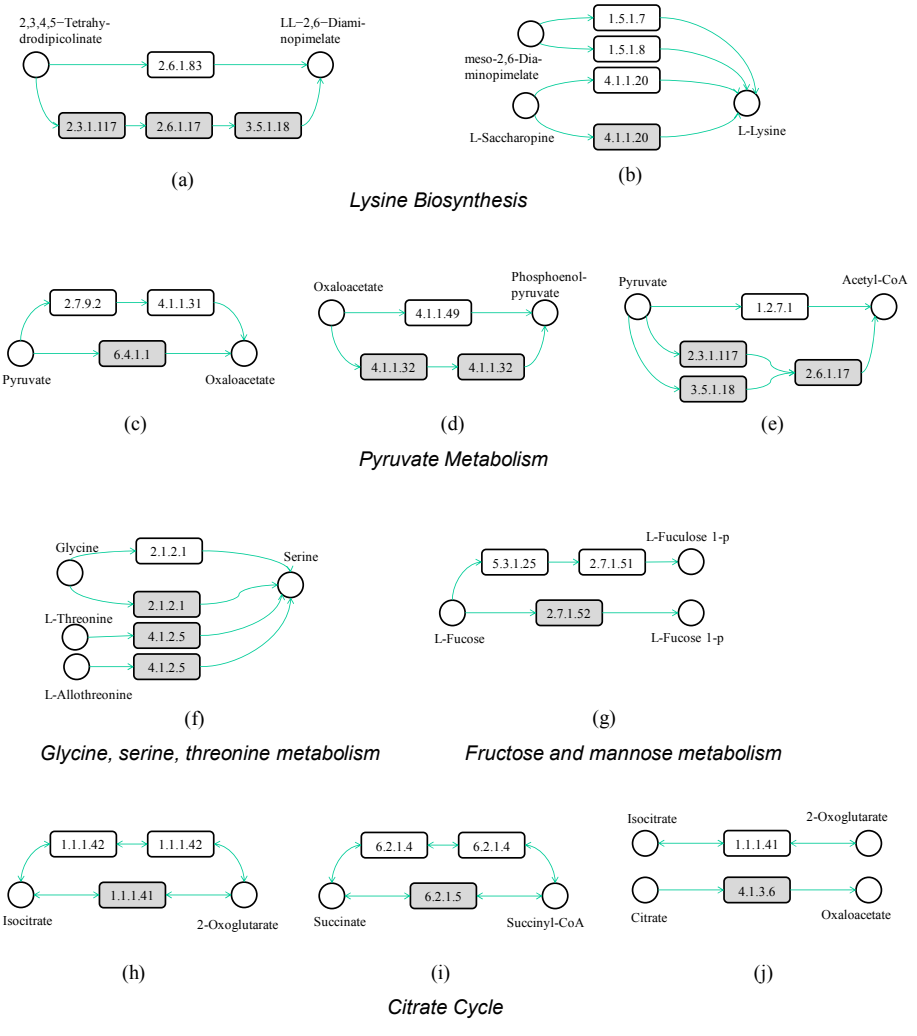



**Fig. 13.** Visual representations of subnetwork mappings reported in Table 4. Figures (a) through (j) correspond to rows 1 through 10 of Table 4. Enyzmes are represented by their Enzyme Commission (EC) numbers [16].

### 4.3 One-to-Many Mappings within and across Major Clades

In Section 4.2, we demonstrated that SubMAP algorithm can find alternative subnetworks on a number of examples. An obvious question that follows is: How frequent are such alternative subnetworks and what are their characteristics? In other words, is there really a need to allow one-to-many mappings in alignment. In this section, we present an exhaustive comparison between 9 different organisms, 3 from each major phylogenic clades. These organisms are:

*T.acidophilum*, *Halobacterium sp.*, *M.thermoautotrophicum* from Archaea;
*H.sapiens*, *R.norvegicus*, *M.musculus* from Eukaryota;
*E.coli*, *P.aeruginosa*, *A.tumefaciens* from Bacteria.

The comparisons are performed on 10 common networks for these 9 organisms from KEGG. For each of these common networks, the results include all possible pairs of the 9 organisms ($\binom{9}{2}$= 36). These alignments are performed using SubMAP algorithm, for $k = 2$, 3 and 4. More details on how this comparison is done can be found in Ay *et al*. [38].

**Table 5.** Percentages of 1-to-1, 1-to-2, 1-to-3 and 1-to-4 mappings in between and across three major clades. (**A**: Archaea, **E**: Eukaryota, **B**: Bacteria)

| | **1-to-1** | **1-to-2** | **1-to-3** | **1-to-4** |
|---|---|---|---|---|
| **E-E** | 89.6 | 8.8 | 1.1 | 0.5 |
| **B-B** | 80.1 | 16.0 | 3.1 | 0.8 |
| **A-A** | 78.3 | 15.7 | 4.7 | 1.3 |
| **B-E** | 69.1 | 23.1 | 6.3 | 1.5 |
| **A-B** | 60.5 | 28.3 | 8.5 | 2.7 |
| **A-E** | 55.8 | 31.0 | 10.4 | 2.8 |

Table 5 summarizes the results of this comparison. The percentages of each mapping type between two clades is shown as a row in this table. The first three rows corresponds to alignments within a clade and the last three represents alignments across two different clades. An important outcome of these results is that there are considerably large number of one-to-many mappings between organisms of different clades. In the extreme case (last row), nearly half of the mappings are one-to-many. The results show that one-to-one mappings is more frequent for alignments within the clades compared to across clades due to high similarity between the organisms of the same clade. For instance, for both the first and last row one side of the query set is the Eukaryota. However, going from first row to last, we see around 40% decrease in the number of one-to-one mappings and 250%, 850% and 450% increase in the number of 1-to-2, 1-to-3 and 1-to-4 mappings respectively. Considering Archaea are single-cell microorganisms (e.g., Halobacteria) and Eukaryota are complex organisms (e.g., animals and plants), these jumps in the number of one-to-many mappings suggest that the individual reactions in Archaea are replaced by a number of reactions in Eukaryota. These results have two major implications. *(i) One-to-many mappings are frequent in nature. To obtain biologically meaningful alignments we need to allow such mappings. (ii) The characteristics of the alternative subnetworks can help in inferring the phylogenic relationship among different organisms.*

## 5 Summary

In this chapter, we presented the pairwise alignment problem for metabolic networks, and the challenges this problem introduce. We described two algorithms

which successfully tackle this problem, as well as the challenges it offer. The first algorithm we described uses a nonredundant graph model for representing networks. Using this model, it aligns reactions, compounds and enzymes of metabolic networks. The algorithm considers both the pairwise similarities of entities (homology) and the organization of networks (topology) for the final alignment. The algorithm reports a consistent alignment, and a meaningful similarity score for the resulting alignment.

The second algorithm tackles the second challenge as well as the first challenge. Given two metabolic networks $P$ and $\bar{P}$ and an upper bound $k$ on the size of the connected subnetworks, the SubMAP algorithm finds the consistent mapping of the subnetworks of $P$ and $\bar{P}$ with the maximum similarity. The algorithm transforms the alignment problem to an eigenvalue problem. The solution to this eigenvalue problem produced a good mixture of homological and topological similarities of the subnetworks. Using these similarity values, the algorithm constructs a vertex weighted graph that connects conflicting mappings with an edge. Then, the alignment problem is transformed into finding the maximum weight independent subset of this graph. A heuristic method is used to solve MWIS problem. The result of this method provides an alignment that has no conflicting pair of mappings.

In addition to the two algorithms mentioned above, we discussed the importance of metabolic network alignment. We presented alternative entities and subnetworks uncovered by the algorithms we described. We also compared the frequency of subnetwork mappings across major clades and discussed the results of this comparison. Applications of the algorithms show that network alignment can be used in a great number of tasks such as metabolic reconstruction of newly sequenced organisms, identification of network holes and identification of drug targets.

## 6 Further Reading

The efforts on comparative analysis of biological networks mainly focused on the alignment of two types of networks, namely metabolic and protein-protein interaction networks (PPI). In this chapter, we discussed in detail the two recent algorithms that are successfully used for alignment of metabolic networks. In this section, we aim to provide the interested reader with pointers of other network alignment algorithms. First, we mention briefly a number of methods developed for aligning PPI networks. We, then, provide short summaries of metabolic network alignment algorithms which are not discussed in earlier sections.

In order to identify conserved interaction paths and complexes between two PPI networks, Kelley *et al.* developed a method called PathBLAST [39,40]. This method is an analog of sequence alignment algorithm BLAST, hence the name, and searches for high-scoring alignments between two protein interaction paths by pairing orthologs proteins one from each path according to the sequence similarity between them and their orders in the corresponding paths. PathBLAST identified large number of conserved paths and duplication events between two

distantly related species (*S. cerevisiae* and *H. pylori*) [39]. In an effort to generalize the topology of the alignment from paths to general graph structures, Koyuturk *et al.* modeled the alignment as a graph optimization problem and proposed efficient algorithms to solve this [24,41,42]. They based their framework, MAWISH, on duplication/divergence model to capture the evolutionary behavior of PPI networks. MAWISH first constructs an alignment graph with insertions and deletions allowed. Then, it finds the maximum weight induced subgraph of this graph and reports it as the resulting alignment. In another work, Berg *et al.* used simulated annealing to align two PPI networks [43,44]. They devised a scoring function for derived motifs from families of similar but not necessarily identical patterns. They developed a graph search algorithm that aims to find the maximum-score alignment(s) of the query networks. The method proposed by Narayanan *et al.* relies on splitting the networks recursively and matching the subnetworks to construct the overall alignment of query PPI networks [45]. Their algorithm gave provable guarantees the correctness of the alignment as well as its efficiency. Dutkowski *et al.* [46] built ancestral networks guided by evolutionary history of proteins together with a stochastic model of interaction emergence, loss and conservation. The application of their method on PPI networks of different species revealed that the most probable conserved ancestral interactions are often related to known protein complexes. QNet, developed by Dost *et al.*, employed color coding technique to decrease the number of iterations necessary to find the highest scoring alignment between two query networks [47]. It extended their earlier method, QPath [48], that works for only linear queries and allowed non-exact matches. QNet limited the query networks to trees and performed efficient alignments with up to nine proteins. Singh *et al.* proposed a new framework that avoids any type of topology restrictions on query networks [20,21]. This framework, IsoRank, inspired by Google's PageRank algorithm, provides an efficient heuristic that defines a similarity score capturing both the sequence similarities of the proteins and the topological similarity of query networks. Mapping the alignment problem to graph isomorphism problem, it reports the alignment as the highest scoring common subgraph between the query networks. Recently, they extended this framework to include protein clusters and to allow multiple networks (IsoRankN [49]). One last method, NetworkBLAST, that aligns PPI networks is based on an earlier algorithm of Sharan *et al.* [50,51]. NetworkBLAST constructs a network alignment graph from queries and uses a heuristic seed-extension method to search for conserved paths or cliques in this alignment graph. It has been extended to NetworkBLAST-M that allows alignment of multiple PPI networks and relies on an efficient representation that is only linear in their size of query networks.

Alignment of metabolic networks is motivated by its importance in understanding the evolution of different organisms, reconstructing metabolic networks of newly sequenced genomes, identifying drug targets and missing enzymes. To the best of our knowledge, the first systematic method to comparatively analyze metabolic networks of different organisms is proposed by Dandekar *et al.* [52]. They focused on glycolytic pathway and combined elementary flux mode

analysis with network alignment to reveal novel aspects of glycolysis. They identified alternative enzymes (i.e., isoenzymes) as well as several potential drug targets in different species. Shortly after, Ogata *et al.* proposed a heuristic graph comparison algorithm and applied it to metabolic networks to detect functionally related enzyme clusters [53]. This method allowed extraction of functionally related enzyme clusters from comparison of complete metabolic networks of 10 microorganisms. By relying on the hierarchy of Enzyme Commission (EC) numbers, Tohsato *et al.* developed an alignment method for comparing multiple metabolic networks [18]. Their idea was to express reaction similarities by the similarities between EC numbers of the enzymes of respective reactions. They devised a new similarity score for enzymes, information content enzyme similarity score, and a dynamic programming algorithm that maximizes this score while aligning query networks. Chen and Hofestadt developed an algorithm and its web tool named PathAligner that is only capable of aligning linear subgraphs of metabolic networks [54,55]. It provided a framework for prediction and reconstruction of metabolic networks using alignment.

In 2005, Pinter *et al.* published a key method in metabolic network alignment with the name MetaPathwayHunter [15]. This dynamic programming method was based on an efficient pattern matching algorithm for labeled graphs. They considered the scenario where one of the input networks is the query and the other is the database or text. They showed that their approximate labeled subtree homeomorphism algorithm works in polynomial time when query networks are limited to multi-source trees (i.e., no cycles). Wernicke and Rasche proposed a fast and simple algorithm that does not limit the topologies of the query networks [56]. This algorithm relied on what they called "local diversity property" of metabolic networks. Exploiting this property, they searched for maximum-score embedding of the query network to the database networks. An alternative approach to network alignment using integer quadratic programming (IQP) was developed by Li *et al.* [57]. They used a similarity score that combines pairwise similarities of molecules and topological similarities of networks. They formulated the problem as searching for a feasible solution that maximizes this similarity between the query networks. For cases where IQP can be relaxed into the corresponding quadratic programming (QP), this method almost always guarantees an integer solution and hence the alignment problem becomes tractable without any approximation.

One of the more recent methods for metabolic network alignment, by Tohsato and Nishimura [17], used similarity of chemical structures of the compounds of a network. Realizing the problems with their earlier work due to discrepancies in the EC hierarchy of enzymes [18], the authors focused solely on the chemical formula of compounds instead of EC numbers of enzymes. They employed fingerprint-based similarity score which utilizes presence or absence of important atoms or molecular substructures of the compounds to define their similarity. The alignment phase of the algorithm uses a simple dynamic programming formulation. Yet another similarity score and an alignment algorithm for metabolic networks was proposed by Li *et al.* [58,59]. They aimed at seeking for

diversity and alternatives in highly conserved metabolic networks. Their similarity score incorporated functional similarity of enzymes with their sequence similarity. Taking reaction directions into account, they first constructed all building blocks of the alignment and sequentially match and score to find the best alignment exhaustively. Latterly, Cheng *et al.* developed a tool, MetNetAligner, that allows a certain number of insertions and deletions of enzymes in for metabolic network alignment [60,19]. It used an enzyme-to-enzyme functional similarity score with the goal of identifying and filling the metabolic network holes by the resulting alignments. MetNetAligner limited one of the query networks to a directed graph with restricted cyclic structure whereas the other query graph is allowed to have arbitrary topology.

## References

1. Margolin, A.A., Wang, K., Lim, W.K., Kustagi, M., Nemenman, I., Califano, A.: Reverse engineering cellular networks. Nature Protocols 1(2), 662–671 (2006)
2. Akutsu, T., Miyano, S., Kuhara, S.: Identification of genetic networks from a small number of gene expression patterns under the Boolean network model. In: Pacific Symposium on Biocomputing (PSB), vol. 4, pp. 17–28 (1999)
3. Wong, S.L., Zhang, L.V., Tong, A.H., Li, Z., Goldberg, D.S., King, O.D., Lesage, G., Vidal, M., Andrews, B., Bussey, H., Boone, C., Roth, F.P.: Combining biological networks to predict genetic interactions. Proceedings of the National Academy of Sciences (PNAS) 101(44), 15682–15687 (2004)
4. Wu, X., Zhu, L., Guo, J., Zhang, D.Y., Lin, K.: Prediction of yeast protein-protein interaction network: insights from the Gene Ontology and annotations. Nucleic Acids Research 34(7), 2137–2150 (2006)
5. Francke, C., Siezen, R.J., Teusink, B.: Reconstructing the metabolic network of a bacterium from its genome. Trends in Microbiology 13(11), 550–558 (2005)
6. Cakmak, A., Ozsoyoglu, G.: Mining biological networks for unknown pathways. Bioinformatics 23(20), 2775–2783 (2007)
7. Ogata, H., Goto, S., Sato, K., Fujibuchi, W., Bono, H., Kanehisa, M.: KEGG: Kyoto Encyclopedia of Genes and Genomes. Nucleic Acids Research 27(1), 29–34 (1999)
8. Keseler, I.M., Collado-Vides, J., Gama-Castro, S., Ingraham, J., Paley, S., Paulsen, I.T., Peralta-Gil, M., Karp, P.D.: EcoCyc: a comprehensive database resource for Escherichia coli. Nucleic Acids Research 33, 334–337 (2005)
9. Schaefer, C.F., Anthony, K., Krupa, S., Buchoff, J., Day, M., Hannay, T., Buetow, K.H.: PID: The Pathway Interaction Database. Nucleic Acids Research 37, 674–679 (2009)
10. Salwinski, L., Miller, C.S., Smith, A.J., Pettit, F.K., Bowie, J.U., Eisenberg, D.: The Database of Interacting Proteins: 2004 update. Nucleic Acids Research 32(1), 449–451 (2004)
11. Sridhar, P., Kahveci, T., Ranka, S.: An iterative algorithm for metabolic network-based drug target identification. In: Pacific Symposium on Biocomputing (PSB), vol. 12, pp. 88–99 (2007)
12. Clemente, J.C., Satou, K., Valiente, G.: Finding Conserved and Non-Conserved Regions Using a Metabolic Pathway Alignment Algorithm. Genome Informatics 17(2), 46–56 (2006)

13. Heymans, M., Singh, A.: Deriving phylogenetic trees from the similarity analysis of metabolic pathways. Bioinformatics 19, 138–146 (2003)
14. Möhring, R.H. (ed.): WG 1990. LNCS, vol. 484, pp. 72–78. Springer, Heidelberg (1991)
15. Pinter, R.Y., Rokhlenko, O., Yeger-Lotem, E., Ziv-Ukelson, M.: Alignment of metabolic pathways. Bioinformatics 21(16), 3401–3408 (2005)
16. Webb, E.C.: Enzyme nomenclature 1992. Academic Press, London (1992)
17. Tohsato, Y., Nishimura, Y.: Metabolic Pathway Alignment Based on Similarity of Chemical Structures. Information and Media Technologies 3(1), 191–200 (2008)
18. Tohsato, Y., Matsuda, H., Hashimoto, A.: A Multiple Alignment Algorithm for Metabolic Pathway Analysis Using Enzyme Hierarchy. In: Intelligent Systems for Molecular Biology (ISMB), pp. 376–383 (2000)
19. Cheng, Q., Harrison, R., Zelikovsky, A.: MetNetAligner: a web service tool for metabolic network alignments. Bioinformatics 25(15), 1989–1990 (2009)
20. Singh, R., Xu, J., Berger, B.: Pairwise Global Alignment of Protein Interaction Networks by Matching Neighborhood Topology. In: Speed, T., Huang, H. (eds.) RECOMB 2007. LNCS (LNBI), vol. 4453, pp. 16–31. Springer, Heidelberg (2007)
21. Singh, R., Xu, J., Berger, B.: Global alignment of multiple protein interaction networks with application to functional orthology detection. Proceedings of the National Academy of Sciences (PNAS) 105, 12763–12768 (2008)
22. Ay, F., Kahveci, T., de Crecy-Lagard, V.: Consistent alignment of metabolic pathways without abstraction. In: Computational Systems Bioinformatics Conference (CSB), vol. 7, pp. 237–248 (2008)
23. Ay, F., Kahveci, T., Crecy-Lagard, V.: A fast and accurate algorithm for comparative analysis of metabolic pathways. Journal of Bioinformatics and Computational Biology 7(3), 389–428 (2009)
24. Koyuturk, M., Grama, A., Szpankowski, W.: An efficient algorithm for detecting frequent subgraphs in biological networks. In: Intelligent Systems for Molecular Biology (ISMB), pp. 200–207 (2004)
25. Deutscher, D., Meilijson, I., Schuster, S., Ruppin, E.: Can single knockouts accurately single out gene functions? BMC Systems Biology 2, 50 (2008)
26. Watanabe, N., Cherney, M.M., van Belkum, M.J., Marcus, S.L., Flegel, M.D., Clay, M.D., Deyholos, M.K., Vederas, J.C., James, M.N.: Crystal structure of LL-diaminopimelate aminotransferase from Arabidopsis thaliana: a recently discovered enzyme in the biosynthesis of L-lysine by plants and Chlamydia. Journal of Molecular Biology 371(3), 685–702 (2007)
27. McCoy, A.J., Adams, N.E., Hudson, A.O., Gilvarg, C., Leustek, T., Maurelli, A.T.: L,L-diaminopimelate aminotransferase, a trans-kingdom enzyme shared by Chlamydia and plants for synthesis of diaminopimelate/lysine. Proceedings of the National Academy of Sciences (PNAS) 103(47), 17909–17914 (2006)
28. Hattori, M., Okuno, Y., Goto, S., Kanehisa, M.: Development of a chemical structure comparison method for integrated analysis of chemical and genomic information in the metabolic pathways. Journal of the American Chemical Society (JACS) 125(39), 11853–11865 (2003)
29. Haveliwala, T.H., Kamvar, S.D.: The Second Eigenvalue of the Google Matrix. Stanford University Technical Report (March 2003)
30. Lovasz, L.: Stable set and polynomials. Discrete Mathematics 124, 137–153 (1994)
31. Austrin, P., Khot, S., Safra, M.: Inapproximability of Vertex Cover and Independent Set in Bounded Degree Graphs. In: IEEE Conference on Computational Complexity, pp. 74–80 (2009)

32. Berman, P., Karpinski, M.: On some tighter inapproximability results. LNCS (1999)
33. Sakai, S., Togasaki, M., Yamazaki, K.: A note on greedy algorithms for the maximum weighted independent set problem. Discrete Applied Mathematics 126, 313–322 (2003)
34. Green, M.L., Karp, P.D.: A Bayesian method for identifying missing enzymes in predicted metabolic pathway databases. BMC Bioinformatics 5, 76 (2004)
35. Sridhar, P., Song, B., Kahveci, T., Ranka, S.: Mining metabolic networks for optimal drug targets. In: Pacific Symposium on Biocomputing (PSB), pp. 291–302 (2008)
36. Kim, J., Copley, S.D.: Why Metabolic Enzymes Are Essential or Nonessential for Growth of Escherichia coli K12 on Glucose. Biochemistry 46(44), 12501–12511 (2007)
37. Saunders, P.P., Broquist, H.P.: Saccharopine, an intermediate of aminoadipic acid pathway of lysine biosynthesis. Journal of Biological Chemistry 241, 3435–3440 (1966)
38. Ay, F., Kahveci, T.: SubMAP: Aligning metabolic pathways with subnetwork mappings. In: Berger, B. (ed.) RECOMB 2010. LNCS, vol. 6044, pp. 15–30. Springer, Heidelberg (2010)
39. Kelley, B.P., Sharan, R., Karp, R.M., Sittler, T., Root, D.E., Stockwell, B.R., Ideker, T.: Conserved pathways within bacteria and yeast as revealed by global protein network alignment. Proceedings of the National Academy of Sciences (PNAS) 100(20), 11394–11399 (2003)
40. Kelley, B.P., Yuan, B., Lewitter, F., Sharan, R., Stockwell, B.R., Ideker, T.: PathBLAST: a tool for alignment of protein interaction networks. Nucleic Acids Research 32(2), 83–88 (2004)
41. Koyutürk, M., Grama, A., Szpankowski, W.: Pairwise Local Alignment of Protein Interaction Networks Guided by Models of Evolution. In: Miyano, S., Mesirov, J., Kasif, S., Istrail, S., Pevzner, P.A., Waterman, M. (eds.) RECOMB 2005. LNCS (LNBI), vol. 3500, pp. 48–65. Springer, Heidelberg (2005)
42. Koyuturk, M., Kim, Y., Topkara, U., Subramaniam, S., Szpankowski, W., Grama, A.: Pairwise alignment of protein interaction networks. Journal of Computational Biology 13(2), 182–199 (2006)
43. Berg, J., Lassig, M.: Local graph alignment and motif search in biological networks. Proceedings of the National Academy of Sciences (PNAS) 101(41), 14689–14694 (2004)
44. Berg, J., Lassig, M.: Cross-species analysis of biological networks by Bayesian alignment. Proceedings of the National Academy of Sciences (PNAS) 103(29), 10967–10972 (2006)
45. Narayanan, M., Karp, R.M.: Comparing Protein Interaction Networks via a Graph Match-and-Split Algorithm. Journal of Computational Biology 14(7), 892–907 (2007)
46. Dutkowski, J., Tiuryn, J.: Identification of functional modules from conserved ancestral protein protein interactions. Bioinformatics 23(13), 149–158 (2007)
47. Dost, B., Shlomi, T., Gupta, N., Ruppin, E., Bafna, V., Sharan, R.: QNet: A tool for querying protein interaction networks. In: Speed, T., Huang, H. (eds.) RECOMB 2007. LNCS (LNBI), vol. 4453, pp. 1–15. Springer, Heidelberg (2007)
48. Shlomi, T., Segal, D., Ruppin, E., Sharan, R.: QPath: A Method for Querying Pathways in a Protein-Protein Interaction Network. BMC Bioinformatics 7(199) (2006)

49. Liao, C.S., Lu, K., Baym, M., Singh, R., Berger, B.: IsoRankN: spectral methods for global alignment of multiple protein networks. Bioinformatics 25(12), 238–253 (2009)
50. Sharan, R., Suthram, S., Kelley, R.M., Kuhn, T., McCuine, S., Uetz, P., Sittler, T., Karp, R.M., Ideker, T.: Conserved patterns of protein interaction in multiple species. Proceedings of the National Academy of Sciences (PNAS) 102, 1974–1979 (2005)
51. Kalaev, M., Smoot, M., Ideker, T., Sharan, R.: NetworkBLAST: comparative analysis of protein networks. Bioinformatics 24(4), 594–596 (2008)
52. Dandekar, T., Schuster, S., Snel, B., Huynen, M., Bork, P.: Pathway alignment: application to the comparative analysis of glycolytic enzymes. Biochemistry Journal 343, 115–124 (1999)
53. Ogata, H., Fujibuchi, W., Goto, S., Kanehisa, M.: A heuristic graph comparison algorithm and its application to detect functionally related enzyme clusters. Nucleic Acids Research 28, 4021–4028 (2000)
54. Chen, M., Hofestadt, R.: PathAligner: metabolic pathway retrieval and alignment. Appl. Bioinformatics 3(4), 241–252 (2004)
55. Chen, M., Hofestadt, R.: Prediction and alignment of metabolic pathways. Bioinformatics of Genome Regulation and Structure II, 355–365 (2011)
56. Wernicke, S., Rasche, F.: Simple and fast alignment of metabolic pathways by exploiting local diversity. Bioinformatics 23(15), 1978–1985 (2007)
57. Li, Z., Zhang, S., Wang, Y., Zhang, X.S., Chen, L.: Alignment of molecular networks by integer quadratic programming. Bioinformatics 23(13), 1631–1639 (2007)
58. Li, Y., Ridder, D., de Groot, M.J.L., Reinders, M.J.T.: Metabolic Pathway Alignment (M-Pal) Reveals Diversity and Alternatives in Conserved Networks. In: Asia Pacific Bioinformatics Conference (APBC), pp. 273–286 (2008)
59. Li, Y., Ridder, D., de Groot, M.J.L., Reinders, M.J.T.: Metabolic pathway alignment between species using a comprehensive and flexible similarity measure. BMC Systems Biology 2(1), 111 (2008)
60. Cheng, Q., Berman, P., Harrison, R., Zelikovsky, A.: Fast Alignments of Metabolic Networks. In: IEEE International Conference on Bioinformatics and Bioengineering (BIBE), pp. 147–152 (2008)

# Chapter 6
# Estimation of Distribution Algorithms in Gene Expression Data Analysis

Elham Salehi and Robin Gras

Department of Computer Science, University of Windsor,
Windsor, Ontario, N9B 3P4
{salehie,rgras}@uwindsor.ca

**Abstract.** Estimation of Distribution Algorithm (EDA) is a relatively new optimization method in the field of evolutionary algorithm. EDAs use probabilistic models to learn properties of the problem to solve from promising solutions and use them to guide the search process. These models can also reveal some unknown regularity patterns in search space. These algorithms have been used for solving some challenging NP-hard bioinformatics problems and demonstrated competitive accuracy. In this chapter, we first provides an overview of different existing EDAs and then review some of their application in bioinformatics and finally we discuss a specific problem that have been solved with this method in more details.

## 1 Introduction

A large number of problems in computational biology and bioinformatics can be formulated as optimization problems either single or multi-objective [1], [2].Therefore powerful heuristic search techniques are needed to tackle them. Population-based search algorithms have shown great performance in finding the global optimum. Unlike classical optimization methods, population based optimization methods do not limit their exploration to a small region of the solution space and are able to explore a vast search space.

A population-based method uses a set of solutions rather than a single solution in each iteration of the algorithm and provide a natural, intrinsic way to explore the search space. They are inspired from living organisms which adapt themselves to their environment. The most well-known population-based method is Evolutionary Computation (EC) including Genetic Algorithms (GAs) [9], [10]. EC algorithms usually use a operator called crossover/recombination to recombine two or more solutions (called also individuals) to generate new individuals. Another operator used in these algorithms is mutation which is kind of modifier which can change the composition of an individuals. Selection of individuals are based on a quality measure such as the value of an objective function or the results of some experiment. Selection can be considered as the driving force in EC algorithms. Individuals with higher fitness have the higher chance to be chosen for producing the next iteration individuals/search points set. The general idea behind the concept of EC is that the exploration of the solution space is guided by some information about the previous

D.E. Holmes, L.C. Jain (Eds.): Data Mining: Found. & Intell. Paradigms, ISRL 25, pp. 101–121.
springerlink.com 

step of the exploration. These information come from the use of a set of solutions from which statistical properties can be extracted giving some insights about the structure of the optimisation problem to solve. These statistical properties can in turn be used to generate new promising potential solutions. In GA, this is the crossover operator which use the statistical information of the population as it generate a new solution by combining two previously generated solutions. The mutation operator gives the possibility to bring new information into the population that cannot be discovered by just combining the existing solutions. Finally, the selection process allows to drift the exploration process toward the solutions with higher fitness.

The recombination operator in GAs manipulates the partial solutions of an optimization problem. These partial solutions are called *building blocks* [3], [6]. It often happens that the building blocks are loosely distributed in a problem domain. Therefore a fixed crossover operator can break the building blocks and lead to convergence to local optimum. This problem is called *linkage problem*[24]. This problem makes the classical genetic algorithm inefficient in solving problems composed of sum of simple sub-problems[24], [30]. Another problem with classical GA is to define the parameters such as the crossover and mutation probabilities. In order to solve these deficiencies another group of evolutionary algorithms called Estimation of Distribution Algorithms (EDAs) [4], [8], [9] also called Probabilistic Model Building Genetic Algorithms (PMGAs) have been proposed [4]. EDAs learn distributions or in other words probabilistic models from the most promising solutions, to guide the search process and preserve the important building block in the next generation. EDAs have been used in different data mining problems such as feature subset selection[40] and classifier systems [42] in many bioinformatics problems.

This chapter is organized as follows. In Section 2 we discuss the different EDAs in more details. Section 3 is dedicated to application of EDA in gene expression data analysis, especially the problem of non-unique oligonucleotide probe selection is discussed in more details. finally we conclude in section 4.

## 2 Estimation of Distribution of Algorithms

EDA also known as Probabilistic Model Building Genetic Algorithm (PMBGA) is a family of population based search algorithms and can be considered as an extension of genetic algorithm [3],[4]. It has been introduced by Mühlenbein and Paaß[4] to improve some deficiencies of genetic algorithms in solving some problems such as optimizing deceptive and non-separable functions[40]. For this purpose EDA uses the correlation of variables in samples of high fitness solutions and, instead of crossover and mutation operators used in genetic algorithm, EDA exploits the probability distribution obtained from to current population of promising solutions to generate the new population or in other word new search points.

The simplest form of EDA proposed by Baluja [5] is named Probabilistic Incremental Learning (PBIL). In PBIL a population is generated from the probability vector which is updated in each generation based on the fittest individual and using a mutation operator. Numerous variation of EDA have been introduced up to now. Most of these algorithms are designed to solve discrete problems and the solutions are represented as binary vectors. Although several EDAs have been proposed for

continuous and mixed problems as well [83], [84]. In this chapter we focus on the discrete domain and introduce main classes of EDAs.

The steps of an EDA are summarized in following algorithm.

1. Generate a random initial set of solution S
2. Calculate the fitness of individuals in S
3. Select a subset of promising solution in S
3. Build a probabilistic model P of the selected solutions
4. Generate new set of solutions by sampling the model P and replace S with this set
5. If the termination criteria are not meet go to step 2

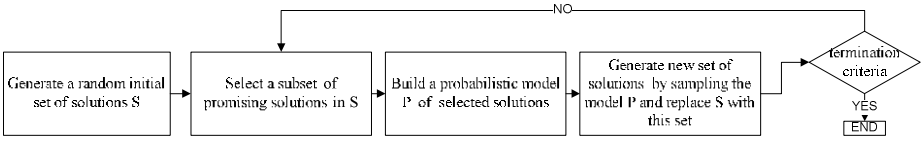



**Fig. 1.** EDA flow chart

In addition to guiding through search space, the probabilistic models learned in each iteration of EDAs can represent a priori information about the problem structure. This information can be used for a more efficient search of optimum solutions. In case of *Black box optimization problems* the probabilistic model can reveal important unknown information about the structure of the problems[85]. The probabilistic models learned during the execution of the algorithm can also be considered as models of the function to be optimized and therefore might be used for predicting the values of the function when the function is unknown.

The main difference between different EDAs is due to the class of probabilistic models they use. Although the different selection and replacement strategies used in them can also have significant effects in their efficiency. Choosing the best EDA for a specific problem can be difficult when the structure of the problem is unknown and it might be useful to try different probabilistic models and selection and replacement methods to find the best combination of them for a given problem [11].

In the next section we discuss several probabilistic model usually used in EDA.

## 2.1 Model Building in EDA

Different probabilistic models have been used to estimate the distribution of a selected set of solutions in EDA. The learning algorithm tries to detect the dependencies between variables of an optimization problem and to represent the probabilistic dependencies between them using probabilistic models. Then, in the sampling phase of the algorithm, these statistical dependencies are used to generate new solutions. The class of probabilistic models used in EDAs can have a great effect on the ability of learning an accurate representation of the dependencies between variables. An accurate estimate can capture the building-block structure of an optimization problem and ensure an effective mixing and reproduction of building-blocks. However finding an accurate model can be very costly, therefore a trade-off

between the efficiency and accuracy need to be found. The EDAs usually are classified based on the complexity of the probabilistic models used in them. In some models the variables are considered independent or only pairwise dependencies are considered. However there are EDAs with more complex models which are able to model problems with very complex structure with overlapping multivariate building blocks. In Fig. 2 different kind of dependencies among the variables are presented.

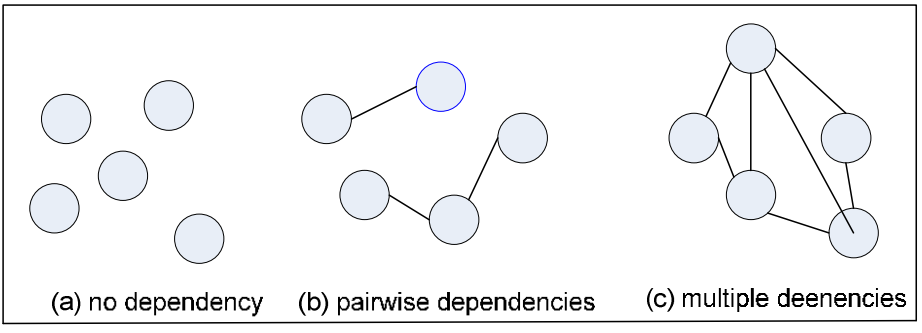



**Fig. 2.** Different kind of dependencies among variables of a problem

### 2.2 Notation

In this section, we adopted the notion used in [41]. The notations represent each search point/individual in a population by a fixed-length vector $X = (X_1, X_2, ..., X_n)$ where $X_i$ , i=0,…,n is a random variable, for a problem with n variables. The value of each variable is represented by $x_i$. Usually $X_i$ is a binary variable but it can also gets its value from a finite discrete set or even take a real value. Let $x = (x_1, x_2, ..., x_n)$ be an instantiation of vector X. Then $P(X_i = x_i)$, or simply $P(x_i)$, is the univariate marginal distribution of the variable $X_i$ and $P(X = x)$, or $P(x)$, is the joint probability distribution function of *x*. $P(X_i = x_i \mid X_j = x_j)$, or simply $P(x_i \mid x_j)$, is the conditional probability density function of the variable $X_i$ taking value $x_i$ given the value $x_j$ taken by $X_j$.

### 2.3 Models with Independent Variables

Assuming that all the variables in a problem are independent it is possible to model them simply by considering a set of frequencies of all values of each variable of the selected set of individuals. In this case, all the variables are considered as univariate and the joint probability distribution is the product of marginal probabilities of the n variables.

Population Based Incremental Learning (PBIL) [5], [12], Univariate Marginal Distribution Algorithm (UMDA) [13], and compact Genetic Algorithm (cGA) [14] consider no interaction among variables.

PBIL, also referred as incremental univariate marginal distribution with learning, [15] and Incremental Univariate Marginal Distribution Algorithm (IUMDA)[13], uses a probability vector $(p_1, p_2,..., p_n)$ as the model for generating the new solution. $p_i$ denotes the probability of having the value 1 for the variable $i$. The initial value for each $p_i$ is 0.5. In each iteration of the algorithm the probability vector is updated based on the best solution of the selected promising solution using

$$p_i = p_i + \lambda(x_i - p_i),$$

where $\lambda \in (0,1)$ is the learning rate and $x_i$ is the value of $i^{th}$ variable.

In PBIL and cGA the population is modeled by a probability vector. However, the probability vector modification is performed in a way that a direct correspondence between this vector and the population represented by this vector exists. like PBIL, each entry $p_i$ in the probability vector is initialized to 0.5. In each iteration a variant of binary tournament in which the worst of the two solutions is replaced by the best one is used to update the probability vector using a population of size N. If $b_i$ and $w_i$ represents the $i^{th}$ position of the best and the worst of the two solutions then the probability vector update is as follow:

$$p_i = p_i + (b_i - w_i)\frac{1}{N}.$$

Unlike cGA and PBIL, UMDA selects a population of promising solutions in each iteration similar to traditional GAs. Then the frequencies of the values of each variable in the selected set are used to generate new solutions that replace the old ones and this process repeated until the termination criteria are met.

All of these algorithms that do not consider interdependencies of variables are not able to solve the problems with strong dependencies among their variables. However, they are able to solve problems which are decomposable into sub problems of order one efficiently. Since univariate EDAs are simple and fast and also they scale up quite well they are widely used in many applications specially in problems with high cardinality such as bioinformatics problems.

### 2.4 Models with Pair Wise Dependencies

To encode the pair wise dependencies between the variables of a problem several probabilistic models have been used. They use a chain, a tree or a forest as a model for representing the interdependencies among variables[16,17,19,20].

One of the algorithm which has been proposed to model the pair-wise interaction between variables is Mutual Information-Maximizing Input Clustering(MIMIC) algorithm[17]. The graphical model used in MIMIC is a chain structure that maximizes the mutual information of the neighboring variables. To specify this model an ordering of variables, the probability of the first position and the conditional probability of other variables, given their preceding variable in the chain should be specified. It leads to the following joint probability distribution for an given order. $\pi = i_1, i_2,..., i_n$.

$$P(x) = P(x_{i_1} \mid x_{i_2})P(x_{i_2} \mid x_{i_3})...P(x_{i_{n-1}} \mid x_{i_n})P(x_{i_n}).$$

MIMIC uses an greedy algorithm to find an order that maximizes the mutual information of neighboring variables and minimize the kullback-liebler divergence[18] between the chain and the complete joint distribution. Although using chain allows a very limited representation of dependencies, it can encode some few dependencies between variables in the solution vectors which is not possible when using a uniform or one point crossover.

Another algorithm for encoding pair wise dependencies is Combining Optimizer with Mutual Information Trees(COMIT) [19]. COMIT uses a tree structure to model the best solutions and uses a Maximum Weight Spaning Tree(MWST) to construct the tree structure. The joint probability distribution in CMMIT can be presented by:

$$P(x) = \prod_{i=1}^{i=n} P(x_i \mid x_j) \ ,$$

where $X_j$ is the parent of $X_i$ in the tree.

One of the algorithm which has been proposed to model the pair wise interaction between variables is the Bivariate Marginal Distribution Algorithm(BMDA) [16]. BMDA is an extension to UMDA. BMDA uses a forest (a set of mutually independent tree ) to model the promising solutions. A Pearson's chi-square test [21] is used to measure the dependencies and to define the pair of variables which should be connected.

These models are able to capture some of the dependencies of order 2. Therefore, an EDAs with pair wise probabilistic models can be efficient on problems decomposable into sub-problems of order at most two. In order to model higher order interaction between variables of a problem, trees and forest can be combined[20].

### 2.5 Models with Multiple Dependencies

Using more complex models to encode multivariate dependencies in EDAs, makes them powerful algorithms. However, complex model learning algorithms used in these EDAs are very time consuming and finding the global optimal model is not guaranteed. Factorized Distribution Algorithm (FDA)[22], Extended Compact Genetic Algorithm (ECGA)[23], Bayesian Optimization Algorithm (BOA)[24], Estimation of Bayesian Networks Algorithm (EBNA)[25] are examples of the EDAs with probabilistic models able to capture multiple dependencies among variables of the problems. They use statistics of order greater than two to factorize the joint probability distribution.

Factorize Distribution Algorithm uses a fixed factorization of distribution which should be given by a an expert. Although the model in FDA is allowed to contain multivariate marginal and conditional probabilities, it just learns the probabilities and the structure is fixed by the expert. Therefore the problem should be first decomposed and then the decomposition is factorized. FDA can use prior information about search space regularities but it is not able to learn them which is the main idea of black box optimization. FDA is applied to additively decomposed functions.

ECGA uses a marginal product model in which variables are partitioned into several clusters. In order to avoid complex models ECGA uses a variant of minimum length metric MDL [26] to discriminate models. ECGA uses a greedy algorithm that

starts with one variable in each cluster and then in each iteration merge some of the current clusters in a way that maximize the metric. The probability model in ECGA can be written as:

$$P(x) = \sum_{i=0}^{k} P(x_{c_i}) \ ,$$

where, $P(x_{c_i})$ is the marginal probability of a cluster i of dependent variables and k is the number of clusters. ECGA performs well for the problems that do not contain overlapping building blocks or, in other words, are decomposable into independent sub-problems.

Bayesian Optimization Algorithm (BOA) [27] or as it named in [28] Estimation of Bayesian Network Algorithm (EBNA) has been proposed in order to build a more general EDA with less restrictive assumption about the dependencies and structure of the problems. BOA models the population of promising solutions as a Bayesian network [36] and samples this network to generate the new solutions.

A Bayesian network can be considered as the graphical factorization of a probability distributions. The conditional dependencies/ independencies among the variables are coded as a directed acyclic graph G and the factorization of the probability distribution can be written as:

$$P(x) = \prod_{i=1}^{n} p(x_i \mid par(x_i)) \ ,$$

where $par(x_i)$ represents a set of the corresponding values of parent set of $X_i$ in the graph G ( variables from which there is an arc to $X_i$ ). Fig. 3 shows a simple Bayesian network. In this Example the parent set of Node Par($X_4$ )={$X_2$, $X_3$}. The joint probability $P(X_1 , X_2 , X_3, X_4 , X_5)$factorizes into the product $P(X_1)P(X_2|X_1)P(X_3|X_1)P(X_4|X_2,X_3)P(X_5|X_3)$. Each node $X_i$ in the graph also has a conditional probability distribution (CPD) table or a set of local parameters which define the distribution of the variable knowing the values of its parents.

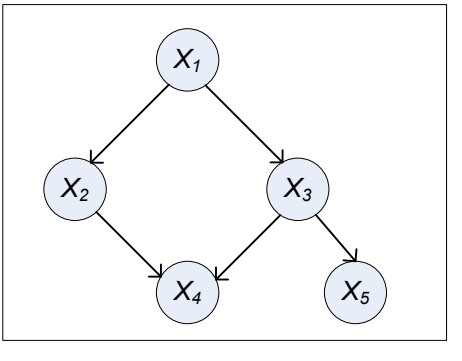



**Fig. 3.** A simple Bayesian network

Unlike FDA, in BOA both structure and parameters of the factorization are learned from the selected population of promising solution and it does not need any extra information about the structure of the dependencies among the variable of the problem to be solved.

The major challenge in using Bayesian networks is learning their structures. Based on the nature of the modeling, structure learning methods are classified in two groups: constrained-based, score-and-search [29]. The first group tries to discover the structure by finding the independency relations among the subsets of variables and gives as an output a directed acyclic graph. The second group uses some scoring function to measure the quality of every candidate network. In fact scoring function measures the fit of the network to the data. A search algorithm is used to explore the space of the possible networks and find the network with the highest score. The most straightforward method for learning Bayesian networks is using a simple hill climbing heuristics. Starting from an empty or random network all the possible moves (adding, deleting, and reversing an edge) are considered and the one that improves the score of the network the most is chosen at each step. The repeated version of this algorithm is used for escaping from local optimum. It returns the network with the highest score in various runs. Scoring functions are based on different principles. Some examples of scoring metrics are: Bayesian-Drichlet, entropy and Minimum Description length (MDL)/BIC [29]. Usually just simple hill-climber is used for learning the structure of Bayesian network and the only operator used is addition of a new arc. BOA [27] uses Bayesian-Drichlet metric to measure the quality of the model. The improved version of FDA named Learning Factorized Distribution Algorithm (LFDA)[35] is similar to BOA and also does not need to know the structure of the problem in advance. However it uses a variant of MDL score called Bayesian Information Criterion (BIC) for measuring the quality of network. In [30] BOA has been extended to Hierarchical BOA (HBOA) which is able to solve hierarchically decomposable problems.

In recent years some exact algorithms have been proposed for learning Bayesian networks which are able to find the optimal network when the number of variables are less than 30 variables [32],[33],[34]. Exact-EBNA has been introduced in [31] using a exact Bayesian network learning algorithm. Exact-EBNA can provide more accurate information about the structure of the problem, though the efficiency of the EDA using them might be decreased.

## 3 Application of EDA in Gene Expression Data Analysis

You As a sequence of recent advances in genomic technologies and molecular biology, huge biological information has been generated that requires to be analyzed in order to extract useful knowledge for scientific community. On the other hand huge growth in computational power in last decades made it possible to use the evolutionary algorithms, especially genetic algorithms, in high-dimensional bioinformatics problems. Application of evolutionary computation in bioinformatics can be found in [37].

EDA seems to be a good alternative to GAs when a randomized population search is needed especially in problems for which ordinary GAs fail because of high interdependencies among the variables. Simple EDAs, especially UMDA, have been

already successfully used in some bioinformatics and biomedical problems . Therefore, one can expect that EDAs will find more applications in this area as the number of new problems is increasing. However in order to use the abilities of EDAs in considering the dependencies among variables more efficient or problem specific multivariate EDAs need to be designed.

In most bioinformatics applications in which EDAs have been used, the problem has been formulated as a feature subset selection problem [38],[39]. The goal of these problems is to reduce the number of features needed for a particular task, for example classification. Therefore, a solution of the feature subset selection problem is a subset of the initial features. . For these problems, each individual in EDA or GA represent a subset of the features of the problem and is presented as a binary vector. A value 1 in position i in an individual indicate that the feature i has been selected in this solution. Selecting the individual is based on the value of an objective function which measure the quality of the subset represented by that individual. Such measure can be for example the accuracy of a classifier using that subset of variables for classification. A review of different feature subset selections methods in bioinformatics can be found in [37],[40].

Feature subset selection can be considered as a preprocessing task for many pattern recognition problems, especially in bioinformatics domain, in which lots of irrelevant feature exists. I It can make model building faster and efficient prevent over fitting in classification or provide better clusters and also provide better understanding of the underlying data generation process.

Selecting a subset of features by generating and evaluating random subset of features is called wrapper methods. Figure 4 taken from [86] illustrates how this method works in case of using EDA.

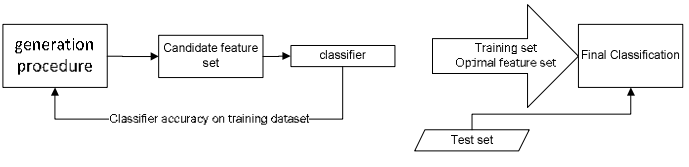



**Fig. 4.** Wrapper methods for feature selection

A procedure similar to the one presented in figure 4 can be used for other problems. It is possible to use some other experiment or function evaluation instead of using a classifier. Using EDA in wrapper methods provides a deep insight into the structure of the problems and integrate the model building and optimization task together. It also decreases the possibility of getting stuck in local optimum.

Using a general probabilistic model such as Bayesian network in EDA can lead to better performance and also provides more information about the search space. However, it also can be the bottleneck of this method and restrict its application just for cases in which the importance of the accuracy highly overweighs the time efficiency. Up to now for most of the applications of EDA in bioinformatics only simple model building such as UMDA has been used.

### 3.1 State-of-Art of the Application of EDAs in Gene Expression Data Analysis

During last decades EDAs have been successfully applied to many NP-hard problems in different medical informatics, genomics and proteomics problems. Although due to the high dimensionality of these problems usually the simplest form of EDAs such as UMDA or PBIL in combination with other statistical methods have been used. Using the strengths of EDAs in considering the dependencies among the variables of the bioinformatics problems necessitates more efficient EDAs. In particular, designing efficient model building algorithm for high-dimensional data sets is needed.

As we mentioned in previous section, in most of the biomedical or bioinformatics applications of EDA, they have been used for feature selections. The First attempts for using EDA based feature selection in a large scale biomedical application was initiated in [44]. In this work PBIL and a Tree-based EDA are used for increasing the accuracy of a classifier for predicting the survival of cirrhotic patients treated with TIPS [45] which is an interventional treatment for cirrhotic patients with portal hypertension [44]. The tree based EDA feature selection could increase the accuracy of prediction significantly. The number of attribute in the problem was 77.

In recent years numerous large genomics data sets have been obtained using high-throughput biotechnology devices. EDAs have been effectively used in solving some of the genomics NP-hard problems including gene structure analysis and gene expression but also inference of genetic networks, classification and clustering microarray data.

#### 3.1.2 Classification

**Gene structure Prediction**

Finding the structure of the genes and their locations in a genome is necessary for the annotation of genomes. Various machine learning methods have been used for these purposes[46]. The problem of identifying a structural elements of a gene such as start and end of the genes and the transition between the coding and non-coding parts (splice sites), is usually modeled as a classification problem. Feature subset selection methods can be used in order to find the most important features among the large number of sequence features. Then, these features can be used for training the classifiers and subsequently discovering the structural elements of the genes. Sayes used EDA based feature selection in gene structure prediction for the first time [47], [48]. Instead of using the traditional EDA, Sayes derived a feature ranking method named UMDA-R. Unlike traditional EDA that returns the best solution UMDA-R uses the distribution estimated from the final population as whole to provide more information about the selected features and returns an array of sorted features based on their relevance. This method, along with a Naive Bayes classifier and a Support Vector Machine (SVM) classifier, were used for the problem of splice sites recognition [48].

**Cancer Classification**

DNA microarray technology is a powerful tool which is being used in biomedical research to study various important areas such as cancer-spreading patterns of gene activity and a drug's therapeutic value [50]. DNA microarrays can measure the

expression level of thousands of genes simultaneously. The data obtained from microarray experiments can be represented as a matrix called gene expression matrix. The rows of gene expression matrix represent genes and columns represent the experimental conditions. The value of each position represents the expression levels of a certain gene under a particular experimental condition. Expression matrix can also include other biological information such as experimental conditions. If this information is used for splitting the dataset (e.g. to healthy and diseased classes) then supervised learning such as classification can be used to analyze the expression data. Otherwise unsupervised analysis (clustering) can be used. Due to the huge dimension of DNA microarrays, different dimensionality reduction methods are necessary as a part of any kind of expression data analysis. Discovering the small number of genes which can cause a particular disease is a preliminary step which is necessary in order to build an accurate classifier.

Simple EDA based feature selection with univariate probabilistic model building approaches have been successfully applied to some microarray gene expression data. Blanco et al. [50] use UMDA and Naive Bayes classifier [54] for cancer classification using two gene expression datasets, related to colon and leukemia. The results shows significant improvement in accuracy of naive Bayes classifier with significant reduction of the number of features. Paul and Iba use a PBIL based feature selection approach with two kind of classifiers: Naive Bayes and weighted voting classifier [55]. Competitive results are achieved in three different benchmark gene expression datasets.

Bielza et al. [53] and González et al. [66] use logic regression based EDA for cancer classification problem. Although logic regression [56] is widely used in biomedical problems for classification of disease using microarray data, it does not perform well when it is used directly on them. Usually, penalized logic regression which uses a penalized likelihood correction is used in order to handle the problem of multicollinearity of DNA microarray [67]. Having a large number of features (genes) and a limited number of samples in microarray data causes another problem and leads to unstable parameter estimates in logistic regression used for DNA array classification. Therefore it is usually used along with some dimensionality reduction techniques. Estimating the model coefficients in logic regression can be considered as an optimization problem. González et al. [66] apply a filtering model to reduce the dimensionality of the data set and then use a real value version of UMDA to optimize the penalized logic regression parameters. Then, they use this method on breast cancer data set and obtain better result comparing to classical logic regression.

Bielza et al. [53] also use a real-value UMDA for regularizing the logistic regression [56] used for microarray classification. The EDA based regularization technique shrinks the parameter estimates and optimizes the likelihood function during its evolution process. The EDA is embedded in an adapted recursive feature elimination procedure which selects the genes that are the best markers for the classification. This method shows excellent performance on four microarray data sets: Breast , Colon , Leukemia and Prostate.

### 3.1.2 Clustering

Clustering DNA micro array data is grouping together genes with similar expression patterns which can reveal new biological information. Since genes in the same cluster

respond similarly in different conditions, they might share a common function. Clustering can also be used as a preprocessing step in gene expression analysis for dimensionality reduction by using a set of representative genes from each cluster instead of using the whole set of genes. Evolutionary algorithms have been successfully used in a large number of clustering problems. A comprehensive survey of evolutionary algorithms designed for clustering including different coding schemes can be found in [62].

Pena et al. [57] use UMDA for unsupervised learning of Bayesian networks which are used for clustering the genes with similar expression profiles. This approach has been evaluated with synthetic and real data including an application to gene expression data clustering for the leukemia database and biologically meaningful clusters have been obtained. Cano et al.[58] use GA and UMDA for non-exclusive clustering of gene expression data. Using overlapping clusters, it is possible to identify genes with more than one functionality. A combination of *Gene Shaving* [63] and EDA or GA is used for this purpose. Gene shaving is a clustering algorithm which finds coherent clusters with high variance across samples.

Biological meaning of the clusters obtained from a real microarray data set has been evaluated using gene ontology term finder [64].

### 3.1.3 Biclustering

If Biclustering is a data mining technique in which the rows and columns of a matrix are clustered simultaneously. It has been introduced in [59]. Like clustering, biclustering is also a NP-hard problem. A bicluster is a subset of genes and a subset of conditions with a high similarity score. Here, the similarity is a measure of the coherence of the genes and of the conditions in the bicluster. By projecting these biclusters onto the dimension of genes or dimension of conditions, a clustering of either the genes or the conditions can be obtained. Biclustering is based on the assumption that several genes change their expression level within a certain subset of conditions [60].

Placios et al. [61] use UMDA and several memetic algorithms [65] to search through the possible bicluster space. Each bicluster can be coded as a binary vector $(r_1, r_2, ..., r_n, c_1, c_2, ..., c_m)$, with the first $n$ positions representing the rows (genes) of the microarray and the last $m$ positions represent the columns (experimental conditions) of the microarray. A value of 1 for $r_i$ indicates that the $i^{th}$ gene has been included in the bicluster and a value of 1 for $c_j$ indicates that $j^{th}$ condition has been also selected in the bicluster. The efficiency of the algorithms has been evaluated using a yeast microarray dataset and the results compared with the algorithm proposed in [60]. Based on the results, the EDA method is the fastest and produces the best bicluster quality followed by the GA. Cano et al[71] consider gene shaving as a multi steps feature selection and use UMDA feature selection method for both non-exclusive clustering and biclustering for gene expression data. They also proposed a biclusteing algorithm based on the principle component analysis and an integrate approach using all tree methods in one platform and evaluate it with two benchmark data sets (yeast and lymphoma). EDA-biclustering out perform all methods in terms of quality using GAP statistics [63] as a quality measure. The results are also validated using the annotation of Gene Ontology.

In most of the biclustering methods the similarity measure for clustering is measured on the whole range of conditions, however in some cases it is possible that expression level of genes does not shows coherency in all conditions and co-regulated genes in some condition might behave independently in other conditions. To solve this problem, Fei et al [70] propose a hybrid multi objective algorithm by combining NSGA-II [75,76] and EDA for biclustering of gene expression data. Biclustering methods try to identify maximal data submatrices including maximal subsets of genes and conditions in which genes show highly correlated expression behavior over a range of different conditions. Therefore, two conflicting objective which should be met are maximizing sub matrices while obtaining high coherency in them. As the size of a bicluster increases the coherency might decrease. This method is evaluated using yeast data set. Better results are achieved and also number of parameters are reduced comparing to blustering using just NSGA-II.

### 3.1.4 Minimum Subset Selection

Finding the smallest subset of a set which satisfies some conditions is another NP-hard problem [77] that can be solved using EDA. The disease-gene [74] association problem and non-unique oligonocleotide probe selection problem [72,78] are two examples of minimum subset selection in bioinformatics which have been solved using EDA. We explain the later in more details.

#### Disease-gene Association

Santana et al. [74] use tree-based EDA to find the minimal subset of tagging single nucleotide polymorphisms (SNPs) which is useful for identifying DNA variations related to specific a disease. SNPs are the sites in human genome where a single nucleotide is different between people. Most SNPs can have two possible nucleotides (alleles). If a SNP in a chromosome has one of its two possible particular nucleotide with high probability another SNP close to it also has one of its two nucleotide which means that the allele frequency difference or in other words the diffrence between the frequency of having each of the two possible nucleotide for both of them are the same. The non-random association of alleles frequencies of two or more SNPs on a chromosome is called linkage disequilibrium (LD) and usually is measured by the correlation coefficient. A tag SNP should have has high LD with other SNPs, in other words a SNP tags other SNP if their correlation coefficient is greater than some threshold.

A subset $T_s$ of a set S of n SNPs is a single-marker of them if each SNP in S is tagged by at least one SNP in $T_s$. A multi-marker tag is defined using a generalized form of correlation coefficient among more than one SNP. In this case a subset $T_m$ of a set S of n SNPs is a multi-marker if each SNP in S is tagged by a subset of $T_m$. To identify a multi-marker for a set of SNPs S, Santana et al. use a tree-based EDA to search through all valid solution (multi-markers). A possible solution is coded with a binary vector $(x_1,...,x_n)$ where $x_i = 1$ if the $i^{th}$ SNP in S is part of the tagging set. The fitness function is the diffrence between n and number of 1s in the solution. The results of this approach for some SNP a problem benchmark which includes 40 SNP problem instances are compared with three other algorithms and show significant reduction in the number of tagging SNPs needed to cover the set S.

**Non-unique Oligonucleotide Probe Selection**

Expression level of genes in an experimental condition are measured based on the amount of mRNA sequences hybridized to their complementary sequences affixes on the surface of a microarray. These complementary sequences are usually short DNA strands called probes. Presence of a biological component (target) in a sample can also be recognized by observing the hybridization patterns of the sample to the probes. Therefore selecting an appropriate set of probes to be affixed on the surface of the microarray is necessary to identify the unknown targets in a sample.

Soltan Ghoraie et al. [72,78] use EDA for the probe selection problem. This problem can be considered as another example of using EDA for minimum subset selection problem. A good probe selection is the one with the minimum number of probes and maximum ability in identifying the targets of the sample. There are two different approaches for this problem: unique and non-unique probe selection. In unique probe selection, for each target there is one unique probe affixed on the microarray to hybridize exclusively to that target. However, this approach is not practical in many cases due to similarities in closely related gene sequences. In non-unique probe selection approach, each probe is designed to hybridize with more than one target. Therefore, in this problem, the smallest set of probes which is able to identify a set of target should be found. Soltan Ghoraie et al. [78] propose a method to analyze and minimize a given a design of candidate none-unique probs.

An initial design is presented as a target-probe incidence matrix. Table 1 taken from [78] is an illustrative example of target-probe incidence matrix $H = (h_{ij})$ of a set of three ($t_1$,$t_2$,$t_3$) and five probes ($p_1$,...,$p_5$ ). In this matrix $h_{ij} = 1$, if probe *j* hybridizes to target *i,* and $h_{ij} = 0$ otherwise. A real example usually has few hundred target and several thousands of probes.

The problem is to find the minimum set of probes which identifies all targets in the sample. If we assume that a sample contains a single probe, then using the probe set of $\{p_1,p_2\}$ we can recognize the target. In the case of having multiple target this set cannot recognize between having $\{t_1,t_2\}$ and $\{t_2,t_3\}$. In this case the probe set of $\{p_3,p_4,p_5\}$ is the minimum probe set to identify all the possible situations.

**Table 1.** Sample target-probe incidence matrix[78]

| | $p_1$ | $p_2$ | $P_3$ | $p_4$ | $p_5$ |
|---|---|---|---|---|---|
| $t_1$ | 0 | 1 | 1 | 0 | 0 |
| $t_2$ | 1 | 0 | 0 | 1 | 0 |
| $t_3$ | 1 | 1 | 0 | 0 | 1 |

To present the problem in a formal way, two parameters $s_{\min}$ (minimum separation coverage) and $c_{\min}$ (minimum coverage constraint) should be defined. Then the problem is to select a minimum probe set given target-probe the incidence matrix H in such a way that each target is hybridized by at least $c_{\min}$ probes and any two subsets

of targets are separated by at least $s_{\min}$ probes which means there are $s_{\min}$ probes that exclusively hybridize with one of the two subset of targets [79, 80].

Soltan Ghoraie et al. use a Bayesian network based EDA(BOA) and a heuristic named dominated row covering (DRC)[81]. In each iteration of BOA a set of solutions is generated. Each solution is a binary vector which represents a subset of probs. The feasibility of a solution (the coverage and the separation constraints) is guaranteed using DRC heuristic.

The single target version of problem, which means that it is considered that only one unique target is present in the sample to identify, is considered as a one-objective optimization problem while the objective is minimizing the number of selected probs. The case of multiple targets version of the problem is considered as a two objective optimization problem. The first objective is minimizing the size of the probe set and the second objective is the ability of the selected set in identifying predetermined number of targets in the sample simultaneously. These two objectives conflict with each other. Average ranking(AR) which is a modified version of WAR[87] method of Bentley and Wakefield is used for this problem. For the first goal the inverse of the cardinality of the probe set (number of ones in the solution) is used. For the second goal a decoding method proposed by Schliep [82] is used. This method uses a Bayesian framework based on Monte Carlo Markov chain sampling, to infer the presence of the targets in the sample. The decoding procedure returns a ranked list of targets based on probability of their presence in the sample. This list is searched for $l$ randomly selected true targets then the position $p_1, p_2, \ldots, p_l$ of each of them in the sorted list produced by decoding procedure is determined (). Therefore the second objective is defined as :

$$Obj_2 = \frac{1}{\sum_{i=1}^{l} p_i} .$$

The maximum of this objective obtained when all true targets ranked in first l positions.

This approach is evaluated using two real datasets HIV1, HIV2 and ten artificial data set and the obtained results in the single target case are compared favorably with the state-of-the-art including integer linear programming(ILP)[88], [89] optimal cutting plane algorithm(OCP)[90] and genetic algorithm(DRC-GA)[91]. Table 2 shows the summary of this comparison. Significant improvement are also obtained using the decoding and the two-objective approaches comparing to the single-objective case.

**Table 1.** Comparison between BOA+DRC and ILP, OCP, and DRC-GA: Number of datasets for which our approach has obtained results better or worse than or equal to methods ILP, OCP, and DRC-GA[78]

| | worse | equal | better |
|---|---|---|---|
| LIP | 2 | 0 | 8 |
| OCP | 5 | 0 | 7 |
| GA-DRC | 0 | 5 | 7 |

## 4 Conclusion

In this chapter, we reviewed EDAs, a class of evolutionary optimization algorithms and different probabilistic model building methods used in them, in order to explore the search space. Then, we reviewed the application of EDAs in different NP-hard problems including feature subset selection for classification, clustering, biclustering of microarray data, and some bioinformatics examples of minimum subset selections.

In most of these applications, due to the high dimensionality of microarray data sets for example, only the simplest models of EDAs, such as UMDA or Tree-based EDA, are used. That means that only a low order of interdependencies among the variables has been considered. Therefore more efficient general probabilistic model buildings are needed in order to capture and use higher order dependencies among the variables in bioinformatics problems. Using the fast Bayesian network learning algorithms which are specifically designed for very high dimensional data sets can be promising. Parallelizing the probabilistic model building or designing specific model building considering the characteristic of the problems such as sparsity of the dependencies might also be helpful.

## References

1. Cohen, j.: Bioinformatics—an Introduction for Computer Scientists. ACM Computing Survey
2. Handi, J., Kell Douglas, B., Knowles, J.: Multiobjective Optimization in Bioinformatics and Computational Biology. IEEE/ACM Transaction on Computational Biology and Bioinformatics 4(2), 279–292 (2007)
3. Pelikan, M., Goldberg, D.E., Lobo, F.G.: A survey of Optimization by Building and Using Probabilistic Models. University of Illinois Genetic AlgorithmsLaboratory, Urbana, IL. IlliGAL Report No. 99018 (1999)
4. Mühlenbein, H., Paaß, G.: From Recombination of Genes to the Estimation of Distributions I. Binary parameters. In: Ebeling, W., Rechenberg, I., Voigt, H.-M., Schwefel, H.-P. (eds.) PPSN 1996. LNCS, vol. 1141, pp. 178–187. Springer, Heidelberg (1996)
5. Baluja, S.: Population Based Incremental learning: A method for integrating genetic search based function optimization and competitive learning. Carnegie Mellon University, Pittsburgh, PA. Technical Report No. CMUCS94163 (1994)
6. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading (1989)
7. Larrañaga, P., Lozano, J.A. (eds.): Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation. Kluwer Academic Publishers, Dordrecht (2002)
8. Lozano, J.A., Larrañaga, P., Inza, I., Bengoetxea, E.: Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms. Springer, Heidelberg (2006)
9. Holland, J.H.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. University of Michigan Press, Ann Arbor (1975)
10. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading (1989)

11. Santana, R., Larranaga, P., Lozano, J.A.: Adaptive Estimation of Distribution Algorithms. In: Cotta, C., Sevaux, M., Sorensen, K. (eds.) Adaptive and Multilevel Metaheuristics. Studies in Computational Intelligence, vol. 136, pp. 177–197. Springer, Heidelberg (2008)
12. Baluja, S., Caruana, R.: Removing the Genetics from Standard Genetics Algorithm. In: Prieditis, A., Russell, S. (eds.) Proceedings of the International Conference on Machine Learning, vol. 46, pp. 38–46. Morgan Kaufmann, San Francisco (1995)
13. Mühlenbein, H.: The Equation for Response to Selection and its Use for Prediction. Evolutionary Computation 5(3), 303–346 (1998)
14. Harik, G.R., Lobo, F.G., Goldberg, D.E.: The Compact Genetic Algorithm. In: Proceedings of the IEEE Conference on Evolutionary Computation, pp. 523–528 (1998)
15. Kvasnicka, V., Pelikan, M., Pospichal, J.: Hill Climbing with Learning (An Abstraction of Genetic Algorithm). Neural Network World 6, 773–796 (1996)
16. Pelikan, M., Muhlenbein, H.: The Bivariate Marginal Distribution Algorithm. In: Advances in Soft Computing – Engineering Design and Manufacturing, pp. 521–535 (1999)
17. De Bonet, J.S., Isbell, C.L., Viola, P.: MIMIC: Finding Optima by Estimating Probability Densities. In: Advances in Neural Information Processing Systems (NIPS-1997), vol. 9, pp. 424–431 (1997)
18. Kullback, S., Leibler, R.A.: On Information and sufficiency. Annals of Math. Stats. 22, 79–86 (1951)
19. Baluja, S., Davies, S.: Using Optimal Dependency-trees for Combinatorial Optimization: Learning the structure of the search space. In: Proceedings of the International Conference on Machine Learning, pp. 30–38 (1997)
20. Santana, R., Ponce de Leon, E., Ochoa, A.: The Edge Incident Model. In: Proceedings of the Second Symposium on Artificial Intelligence (CIMAF-1999), pp. 352–359 (1999)
21. Marascuilo, L.A., McSweeney, M.: Nonparametric and Distribution Free Methods for the Social Sciences. Brooks/Cole Publishing Company, CA (1977)
22. Muhlenbein, H., Mahnig, T., Rodriguez, A.O.: Schemata, Distributions and Graphical Models in Evolutionary Optimization. Journal of Heuristics 5, 215–247 (1999)
23. Harik, G.: Linkage Learning Via Probabilistic Modeling in the ECGA. IlliGAL Report No. 99010, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, IL (1999)
24. Pelikan, M., Goldberg, D.E., Cant´u-Paz, E.: Linkage Problem, Distribution Estimation, and Bayesian Networks. IlliGAL Report No. 98013. University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, IL (1998)
25. Etxeberria, R., Larrañaga, P.: Global Optimization Using Bayesian Networks. In: Rodriguez, A.A.O., Ortiz, M.R.S., Hermida, R.S. (eds.) Second Symposium on Artificial Intelligence (CIMAF-1999), pp. 332–339. Institute of Cybernetics, Mathematics, and Physics and Ministry of Science, Technology and Environment, Habana, Cuba (1999)
26. Rissanen, J.: Modelling by Shortest Data Description. Automatica 14, 465–471 (1978)
27. Pelikan, M., Goldberg, D.E., Cant´u-Paz, E.: Linkage Problem, Distribution Estimation, and Bayesian Networks. IlliGAL Report No. 98013. University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, IL (1998)
28. Etxeberria, R., Larrañaga, P.: Global Optimization Using Bayesian Networks. In: Rodriguez, A.A.O., Ortiz, M.R.S., Hermida, R.S. (eds.) Second Symposium on Artificial Intelligence (CIMAF-1999), pp. 332–339. Institute of Cybernetics, Mathematics, and Physics and Ministry of Science, Technology and Environment, Habana, Cuba (1999)
29. Larranaga, P., Lozano, J.A.: Estimation of Distribution Algorithms. Kluwer Academic Publishers, Dordrecht (2002)
30. Pelikan, M.: Bayesian optimization algorithm: from single level to hierarchy, Ph.D. Thesis. University of Illinois (2002)

31. Echegoyen, C., Santana, R., Lozano, J.A., Larrañaga, P.: The Impact of Exact Probabilistic Learning Algorithms in EDAs Based on Bayesian Networks. Linkage in Evolutionary Computation, 109–139 (2008)
32. Eaton, D., Murphy, K.: Exact Bayesian Structure Learning from Uncertain Interventions. In: Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (2007)
33. Koivisto, M., Sood, K.: Exact Bayesian Structure Discovery in Bayesian networks. Journal of Machine Learning Research 5, 549–573 (2004)
34. Silander, T., Myllymaki, P.: A Simple Approach for Finding the Globally Optimal Bayesian Network Structure. In: Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-2006), Morgan Kaufmann Publishers, San Francisco (2006)
35. Muhlenbein, H., Mahnig, T.: FDA – A Scalable Evolutionary Algorithm for the Optimization of Additively Decomposed Functions. Evolutionary Computation 7(4), 353–376 (1999)
36. Pal, S.K., Bandyopadhyay, S., Ray, S.: Evolutionary Computation in Bioinformatics: A Review. IEEE Transactions on Systems, Man and Cybernetics, Part C 36(2), 601–615 (2006)
37. Saeys, Y., Inza, I., Larrañaga, P.: A Review of Feature Selection Techniques in Bioinformatics. Bioinformatics 23(19), 2507–2517 (2007)
38. Guyon, I., Elisseeff, A.: An Introduction to Variable and Feature Selection. J. Mach. Learn. Res. 3, 1157–1182 (2003)
39. Liu, H., Motoda, H.: Feature Selection for Knowledge Discovery and Data Mining. Kluwer Academic Publishers, Norwell (1998)
40. Inza, I., Larrañaga, P., Etxebarria, R., Sierra, B.: Feature Subset Selection by Bayesian Networks Based Optimization. Artificial Intelligence 27, 143–164 (1999)
41. Liu, H., et al.: A comparative Study on Feature Selection and Classification Methods Using Gene Expression Profiles and Proteomic patterns. Genome Inform. 13, 51–60 (2002)
42. Larrañaga, P., Lozano, j.a.: Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation. Kluwer Academic Publishers, Dordrecht (2002)
43. Butz, M., Pelikan, M., Llora, X., Goldberg, D.E.: Effective and Reliable Online Classification Combining XCS with EDA Mechanisms. In: Pelikan, Sastry, Cantu-Paz (eds.) Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications, pp. 227–249. Springer, Heidelberg (2006)
44. Inza, I., Merino, M., Larrañnaga, P., Quiroga, J., Sierra, B., Girala, M.: Feature Subset Selection by Genetic Algorithms and Estimation of Distribution Algorithms – A Case Study in the Survival of Cirrhotic Patients Treated with TIPS. Artificial Intelligence in Medicine 23(2), 187–205 (2001)
45. Rossle, M., Richter, M., Nolde, G., Palmaz, J.C., Wenz, W., Gerok, W.: New Non-perative Treatment for Variceal Haemorrhae. Lancet 2, 153 (1989)
46. Majoros, W.: Methods for Computational Gene Prediction. Cambridge University Press, Cambridge (2007)
47. Saeys, Y.: Feature Selection for Classification of Nucleic Acid Sequences. PhD thesis Ghent University, Belgium (2004)
48. Saeys, Y., Degroeve, S., Aeyels, D., Rouzé, P., van de Peer, Y.: Feature Selection for Splice Site Prediction: A New Method Using EDA-based Feature Ranking. BMC Bioinformatics 5, 64 (2004)
49. Draghici, S.: Data Analysis Tools for DNA Microarrays. Chapman and Hall/CRC Press (2005)

50. Blanco, R., Larranaga, P., Inza, I., Sierra, B.: Gene Selection for Cancer Classification Using Wrapper Approaches. International Journal of Pattern Recognition and Artificial Intelligence 18(8), 1373–1390 (2004)
51. Paul, T.K., Iba, H.: Identification of Informative Genes for Molecular Classification Using Probabilistic Model Building Genetic Algorithm. In: Deb, K., et al. (eds.) GECCO 2004. LNCS, vol. 3102, pp. 414–425. Springer, Heidelberg (2004)
52. Paul, T., Iba, H.: Gene Selection for Classification of Cancers using Probabilistic Model Building Genetic Algorithm. BioSystems 82(3), 208–225 (2005)
53. Bielza, C., Robles, V., Larranaga, P.: Estimation of Distribution Algorithms as Logistic Regression Regularizers of Microarray Classifiers. Methods Inf. Med. 48(3), 236–241 (2008)
54. Cestnik, B.: Estimating Probabilities: A crucial Task in Machine Learning. In: Proceedings of the European Conference on Artificial Intelligence, pp. 147–149 (1990)
55. Golub, G.R., et al.: Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. Science 286(15), 531–537 (1999)
56. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, Heidelberg (2000)
57. Pena, J., Lozano, J., Larranaga, P.: Unsupervised Learning of Bayesian Networks via Estimation of Distribution Algorithms: An Application to Gene Expression Data Clustering. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 12, 63–82 (2004)
58. Cano, C., Blanco, A., Garcia, F., Lopez, F.J.: Evolutionary Algorithms for Finding Interpretable Patterns in Gene Expression Data. International Journal on Computer Science and Information System 1(2), 88–99 (2006)
59. Morgan, J., Sonquistz, J.: Problems in the Analysis of Survey Data, and a Proposal. Journal of the American Statistical Association 58, 415–434 (1963)
60. Cheng, Y., Church, G.M.: Biclustering of Expression Eata. In: Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology, pp. 93–103. AAAI Press, Menlo Park (2000)
61. Palacios, P., Pelta, D.A., Blanco, A.: Obtaining Biclusters in Microarrays with Population Based Heuristics. In: Evo. Workshops, pp. 115–126. Springer, Heidelberg (2006)
62. Hruschka, E.R., Campello, R.J.G.B., Freitas, A.A., de Carvalho, A.C.P.L.F.: A Survey of Evolutionary Algorithms for Clustering. IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews 39(2), 133–155 (2009)
63. Hastie, T., et al.: Gene Shaving as a Method for Identifying Distinct Set of Genes With Similar Expression Patterns. Genome Biology 1(2), 1–21 (2000)
64. Boyle, E.I., et al.: GO::TermFinder – Open Source Software for Accessing Gene Ontology Information and Finding Significantly Enriched Gene Ontology Terms Associated with a List of Genes. Bioinformatics 20, 973–980 (2004)
65. Hart, W., Krasnogor, N., Smith, J. (eds.): Recent Advances in Memetic Algorithms. Studies in Fuzziness and Soft Computing. Physica-Verlag, Heidelberg (2004)
66. González, S., Robles, V., Peña, J.M., Cubo, O.: EDA-Based Logistic Regression Applied to Biomarkers Selection in Breast Cancer. In: En, X. International Work-Conference on Artificial Neural Networks, Salamanca, Spain (2009)
67. Shen, L., Tan, E.C.: Dimension Reduction-based Penalized Logistic Regression for Cancer Classification Using Microarray Data. IEEE/ACM Trans. Comput. Biol. Bioinformatics 2(2), 166–175 (2005)
68. Armananzas, R., Inza, I., Larranaga, P.: Detecting Reliable Gene Interactions by a Hierarchy of Bayesian Network Classifiers. Comput. Methods Programs Biomed. 91(2), 110–121 (2008)

69. Dai, C., Liu, J.: Inducing Pairwise Gene Interactions from Time Series Data by EDA Based Bayesian Network. In: Conf. Proc. IEEE Eng. Med. Biol. Soc, vol. 7, pp. 7746–7749 (2005)
70. Fei, L., Juan, L.: In: The 2nd International Conference on Bionformatics and Biomedical Engineering, ICBBE 2008, pp. 1912–1915 (2008)
71. Cano, C., Garcia, F., Lopez, J., Blanco, A.: Intelligent System for the Analysis of Microarray Data using Principal Components and Estimation of Distribution Algorithms. Expert Systems with Applications 42(2) (2008)
72. Soltan Ghoraie, L., Gras, R., Wang, L., Ngom, A.: Bayesian Optimization Algorithm for the Non-unique Oligonucleotide Probe Selection Problem. In: Kadirkamanathan, V., Sanguinetti, G., Girolami, M., Niranjan, M., Noirel, J. (eds.) PRIB 2009. LNCS, vol. 5780, pp. 365–376. Springer, Heidelberg (2009)
73. Santana, R., Mendiburu, A., Zaitlen, N., Eskin, E., Lozano, J.A.: Multi-marker Tagging Single Nucleotide Polymorphism Selection Using Estimation of Distribution Algorithms. Artificial Intelligence in Medicine (2010) (article in Press)
74. Deb, K., Pratap, A.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA- II. IEEE Transactions on Evolutionary computation 6(2), 182–197 (2002)
75. Mitra, S., Banka, H.: Multi-objective Evolutionary Biclustering of Gene Expression Data. Pattern Recognition, 2464–2477 (2006)
76. Chen, B., Hong, J., Wang, Y.: The Minimum Feature Subset Selection Problem. Journal of Computer Science and Technology 12(2), 145–153 (1997)
77. Soltan Ghoraie, L., Gras, R., Wang, L., Ngom, A.: Optimal Decoding and Minimal Length for the Non-unique Oligonucleotide Probe Selection Problem. Neurocomputing 15(13-15), 2407–2418 (2010)
78. Klau, G.W., Rahmann, S., Schliep, A., Vingron, M., Reinert, K.: Integer linear programming approaches for non-unique probe selection. Discrete Applied Mathematics 155, 840–856 (2007)
79. Klau, G.W., Rahmann, S., Schliep, A., Vingron, M., Reinert, K.: Optimal Robust Non-unique Probe Selection Using Integer Linear Programming. Bioinformatics 20, i186–i193 (2004)
80. Wang, L., Ngom, A.: A Model-based Approach to the Non-unique Oligonucleotide Probe Selection Problem. In: Second International Conference on Bio-Inspired Models of Net work, Information, and Computing Systems (Bionetics 2007), Budapest, Hungary, December 10–13 (2007) ISBN: 978-963-9799-05-9
81. Schliep, A., Torney, D.C., Rahmann, S.: Group Testing with DNA Chips: Generating Designs and Decoding Experiments. In: IEEE Computer Society Bioinformatics Conference (CSB 2003), pp. 84–91 (2003)
82. Bosman, P.A., Thierens, D.: Mixed IDEAs. Utrecht UniversityTechnical Report UU-CS-2000-45. Utrecht University, Utrecht, Netherlands (2000b)
83. Larrañaga, P., Etxeberria, R., Lozano, J.A., Pena, J.M.: Optimization in Continuous Domains by Learning and Simulation of Gaussian Networks. In: Workshop Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000), pp. 201–204 (2000)
84. Pelikan, M., Sastry, K., Goldberg, D.E.: Evolutionary Algorithms+ Graphical Models = Scalable Black-box Optimization. IlliGAL ReportNo. 2001029, Illinois Genetic Algorithms Laboratory. University of Illinois at Urbana-Champaign, Urbana, IL (2001)
85. Yang, Q., Salehi, E., Gras, R.: Using feature selection approaches to find the dependent features. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010. LNCS, vol. 6113, pp. 487–494. Springer, Heidelberg (2010)
86. Bentley, P.J., Wakefield, J.P.: Finding Acceptable Solutions in the Pareto-Optimal Range using Multiobjective Genetic Algorithms. In: Chawdhry, P.K., Roy, R., Pant, R.K. (eds.) Soft Computing in Engineering Design and Manufacturing, pp. 231–240. Springer Verlag London Limited, London (1997)

87. Klau, G.W., Rahmann, S., Schliep, A., Vingron, M., Reinert, K.: Integer Linear Programming Approaches for Non-unique Probe selection. Discrete Applied Mathematics 155, 840–856 (2007)
88. Klau, G.W., Rahmann, S., Schliep, A., Vingron, M., Reinert, K.: Optimal Robust Non-unique Probe Selection Using Integer Linear Programming. Bioinformatics 20, i186–i193 (2004)
89. Ragle, M.A., Smith, J.C., Pardalos, P.M.: An optimal cutting-plane algorithm for solving the non-unique probe selection problem. Annals of Biomedical Engineering 35(11), 2023–2030 (2007)
90. Wang, L., Ngom, A., Gras, R.: Non-unique oligonucleotide microarray probe selection method based on genetic algorithms. In: 2008 IEEE Congress on Evolutionary Computation, Hong Kong, China, June 1-6, pp. 1004–1010 (2008)

# Chapter 7
# Gene Function Prediction and Functional Network: The Role of Gene Ontology

Erliang Zeng[1], Chris Ding[2], Kalai Mathee[3], Lisa Schneper[3], and Giri Narasimhan[4]

[1] Deprtment of Computer Science and Engineering,
University of Notre Dame, Notre Dame, 46556, USA
ezeng@nd.edu
[2] Department of Computer Science and Engineering,
University of Texas at Arlington, Texas, 76019, USA
CHQDing@uta.edu
[3] Department of Molecular Microbiology, College of Medicine
Florida International University Miami, Florida, 33199, USA
{matheek,schnepel}@fiu.edu
[4] Bioinformatics Research Group (BioRG),
School of Computing and Information Sciences
Florida International University Miami, Florida, 33199, USA
giri@cs.fiu.edu

**Abstract.** Almost every cellular process requires the interactions of pairs or larger complexes of proteins. The organization of genes into networks has played an important role in characterizing the functions of individual genes and the interplay between various cellular processes. The Gene Ontology (GO) project has integrated information from multiple data sources to annotate genes to specific biological process. Recently, the semantic similarity (SS) between GO terms has been investigated and used to derive semantic similarity between genes. Such semantic similarity provides us with a new perspective to predict protein functions and to generate functional gene networks. In this chapter, we focus on investigating the semantic similarity between genes and its applications. We have proposed a novel method to evaluate the support for PPI data based on gene ontology information. If the semantic similarity between genes is computed using gene ontology information and using Resniks formula, then our results show that we can model the PPI data as a mixture model predicated on the assumption that true protein-protein interactions will have higher support than the false positives in the data. Thus semantic similarity between genes serves as a metric of support for PPI data. Taking it one step further, new function prediction approaches are also being proposed with the help of the proposed metric of the support for the PPI data. These new function prediction approaches outperform their conventional counterparts. New evaluation methods are also proposed. In another application, we present a novel approach to automatically generate a functional network of yeast genes using Gene Ontology (GO) annotations. An semantic similarity (SS) is

D.E. Holmes, L.C. Jain (Eds.): Data Mining: Found. & Intell. Paradigms, ISRL 25, pp. 123–162.
springerlink.com 

calculated between pairs of genes. This SS score is then used to predict linkages between genes, to generate a functional network. Functional networks predicted by SS and other methods are compared. The network predicted by SS scores outperforms those generated by other methods in the following aspects: automatic removal of a functional bias in network training reference sets, improved precision and recall across the network, and higher correlation between a genes lethality and centrality in the network. We illustrate that the resulting network can be applied to generate coherent function modules and their associations. We conclude that determination of semantic similarity between genes based upon GO information can be used to generate a functional network of yeast genes that is comparable or improved with respect to those that are directly based on integrated heterogeneous genomic and proteomic data.

**Keywords:** Semantic Similarity, Gene Ontology, Gene Function Prediction, Functional Gene Network.

## 1 Introduction

Gene Ontology (GO) is the most comprehensive gene function annotation system and is widely used. GO consists of sets of structured vocabularies each organized as a rooted directed acyclic graph (DAG), where every node is associated with a GO term and edges represent either a IS-A or a PART-OF relationship [1]. Three independent sets of vocabularies are provided:

**Cellular component** is a description of where in an organism the gene products operate.

**Molecular function** is defined as the specific activities which the gene products are involved in.

**Biological process** describes the role a gene plays in terms of larger pathways and multi-step procedures.

Generally, a gene is annotated by one or more GO terms. The terms at the lower levels correspond to more specific functional descriptions. Therefore if a gene is annotated with a GO term, it is also annotated with the ancestors of that GO term. Thus, the terms at the higher levels have more associated genes. The GO database is constantly being updated. Figure 1 is an example showing a fragment of the structure of GO terms and gene associations in yeast. In Figure 1, each box reports both the number of genes associated with the stated GO term and contains, in parenthesis, the total number of genes associated with the descendants of that GO term. For example, 7(+52) indicates that *seven* genes are directly annotated by that term, and a combined total of 52 genes annotated by the descendants of that term. The GO system provides standardized and integrated gene function annotations that applies uniformly to all organisms.

Recently, the semantic similarity (SS) between GO terms has been investigated. The methods to calculate the semantic similarity between two concepts have been proposed by Resnik, Jiang and Conrath, Lin, and Schlicker, and were

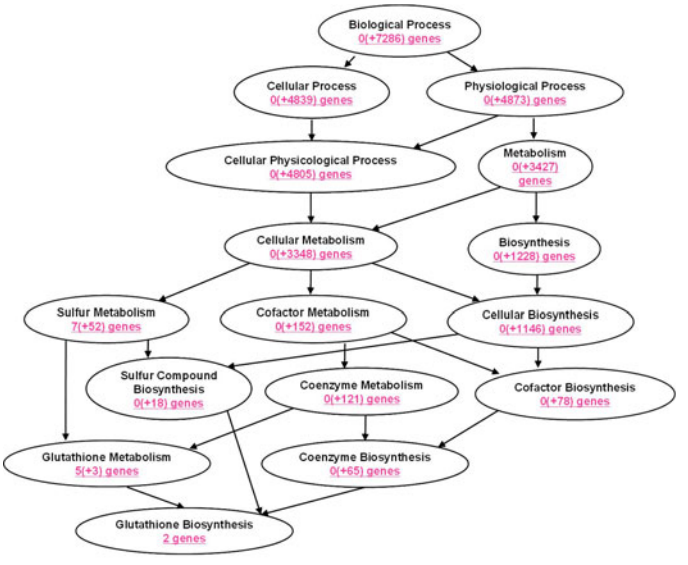



**Fig. 1.** An example showing a fragment of the hierarchy of GO terms and gene associations in yeast

previously applied to other systems such as Wordnet [2–5]. They were first used for GO annotations by Lord *et al.* to investigate the relationship between GO annotations and gene sequences [6]. In this chapter, we introduce the concepts of semantic similarity between genes calculated based on gene ontology, and investigate inherent properties of some previously known methods used to calculate such similarity. We then show how to use such similarity to improve solutions for two biology problems: the problems of gene function prediction and functional gene network generation.

Next, we briefly introduce these two problems and related work.

### 1.1 Gene Function Prediction

Functional annotation of proteins is a fundamental problem in the post-genomic era. To date, a large fraction of the proteins have no assigned functions. Even for one of the most well-studied organisms such as *Saccharomyces cerevisiae*, about a quarter of the proteins remain uncharacterized [7]. Meanwhile, recent advances in genomic sequencing have generated an astounding number of new putative genes and hypothetical proteins whose biological functions remains a mystery.

The recent availability of protein-protein interaction data for many organisms has spurred the development of computational methods for interpreting such data in order to elucidate protein functions. The basic assumption is that if proteins are interacting, they are very likely to share some functions. High throughput techniques such as *yeast two-hybrid* system now produce huge amounts of pairwise PPI data. These data are commonly represented as graphs, with nodes

representing proteins and edges representing interaction between proteins. Some proteins in the graph have known functional annotations, while others do not (uncharacterized proteins). The goal of gene functional annotation is to assign appropriate functional terms to uncharacterized proteins based on neighboring annotated proteins in the PPI networks (Figure 2). Computational methods for protein functional prediction fall into two categories (Figure 2): direct annotation schemes, which infer the function of a protein based on the functional annotations of the proteins in its neighborhood in the network, and module-assisted schemes, which first identify modules of related proteins and then annotate each module based on the known functions of its members. The scheme from Figure 2 shows a network in which the functions of some proteins are known (top), where each function is indicated by a different color. Unannotated proteins are in white. In the direct methods (left), these proteins are assigned a color that is unusually prevalent among their neighbors. The direction of the edges indicates the influence of the annotated proteins on the unannotated ones. In the module-assisted methods (right), modules are first identified based on their density. Then, within each module, unannotated proteins are assigned a function that is unusually prevalent in the module. In both methods, proteins may be assigned with several functions.

Examples of direct annotation schemes include neighborhood-based methods [8–10], probabilistic methods [11, 12], graph-theoretical methods [13–15], and methods integrating multiple data sources [16–19]. Module-assisted approaches

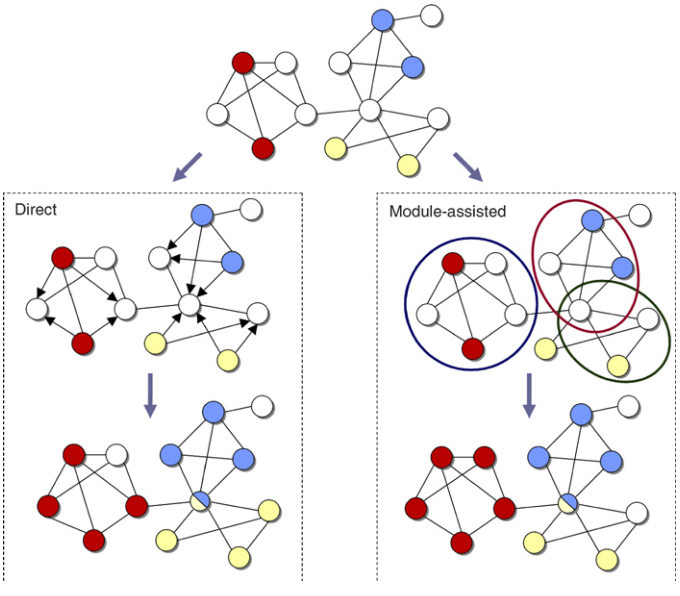



**Fig. 2.** Direct versus module-assisted approaches for functional annotation. Figure reproduced from Sharan et al. [7].

include algorithms that explores modules or clusters from PPI data. Several examples of such methods have been published [20–24]. By functional module one typically means a group of cellular components and their interactions that can be attributed to a specific biological function. A module-assisted approach attempts to first identify coherent groups of genes and then assign functions to all the genes in each group. The module-assisted methods differ mainly in their module detection technique. In the following section, we focus on some direct annotation methods.

The simplest and most direct method for function prediction determines the function of a protein based on the known function of proteins lying in its neighborhood in the PPI network. Schwikowski *et al.*[8] used the so-called majority-voting technique to predict up to three functions that are frequently found among the annotations of its network neighbors. Hishigaki *et al.*[9] approached this problem by also considering the background level of each function across the whole genome. The $\chi^2$-like score was computed for every predicted function. Chua *et al.*[10] proposed to improve the prediction accuracy by investigating the relation between network topology and functional similarity.

In contrast to the local neighborhood approach, several methods have been proposed to predict functions using global optimization. Vazquez *et al.*[13] and Nabieva *et al.*[15] formulated the function prediction problem as a minimum multiway cut problem and provided an approximation algorithm to this NP-hard problem. Vazquez *et al.*[13] used a simulated annealing approach and Nabieva *et al.*[15] applied a integer programming method. Karaoz *et al.*[14] used a similar approach but handled one annotation label at a time. Several probabilistic models were also proposed for this task such as the *Markov random field model* used by Letovsky *et al.*[11] and Deng *et al.*[12], and a statistical model used by Wu *et al.*[25].

### 1.2 Functional Gene Network Generation

Traditional web-lab experiments for detecting protein-protein interactions are time-consuming and costly. Several approaches were recently proposed to predict protein-protein interaction (PPI) networks based on evidence from multi-source data. These include the use of a Bayes classifier to combine multiple data sources [26], the construction of a decision tree to predict co-complexed protein pairs [27], the prediction of PPI by kernel methods [28], the prediction of sets of interacting proteins using a Mixture-of-Experts method [29], and many more.

Although PPI networks have been invaluable for protein complex and protein function prediction as described in Section 1.1, they only account for physical interactions between proteins and thus represent only a subset of important biological relationships. Lee *et al.* sought to construct a more accurate and extensive gene network (i.e., functional gene network) by considering functional, rather than physical associations [30, 31]. They developed a probabilistic framework to derive numerical likelihoods for integrating multi-source data. In such a framework, protein-protein linkages were probabilistic summaries representing functional coupling between proteins. In addition to including direct protein-protein

interaction data, their analysis also included associations not mediated by physical contact, such as regulatory, genetic, or metabolic coupling. The proteins and their regulated genes are among regulatory relationships. Sometimes mutations in two genes produce a phenotype that is surprising in light of each mutation's individual effects. Such a phenomenon is called a *genetic interaction*. When genes are involved in the same metabolic pathway, they form metabolic coupling relationship. Examination of functional networks rather than solely physical interaction networks allows more diverse classes of experiments to be integrated and results in the deciphering of a wider range of biological relationships.

### 1.3 Related Work and Limitations

For the function prediction problem, despite some successful applications of the algorithms discussed in Section 1.1, many challenges still remain. One of the big challenges is that PPI data has a high degree of noise [32]. Most methods that generate interaction networks or perform functional prediction do not have a preprocessing step to clean the data or filter out the noise. Methods that include the reliability of experimental sources have been previously proposed [15]. For example, *mass spectrometry* method is considered more reliable than the *yeast two-hybrid systems* [15]. However, the reliability estimations are crude and do not consider the variations in the reliability of instances within the same experimental source. Some approaches were proposed to predict protein-protein interaction based on evidence from multi-source data. The evidence score calculated from multi-source data is a type of reliability measure of the protein-protein interaction data. Several such approaches have been previously developed [26–31, 33, 34]. Jansen *et al.* combined multiple sources of data using a Bayes classifier [26]. Bader *et al.* developed statistical methods that assign a confidence score to every interaction [34]. Zhang *et al.* predicted co-complexed protein pairs by constructing a decision tree [27]. Ben-Hur *et al.* used kernel methods for predicting protein-protein interactions [28]. Lee *et al.* developed a probabilistic framework to derive numerical likelihoods for interacting protein pairs [30]. Qi *et al.* used a *Mixture-of-Experts* (ensemble) method to predict the set of interacting proteins [29]. The challenges of integrating multi-source data are mainly due to the heterogeneity of the data. For PPI data, data integration also involves eliminating the effect of a functionally-biased reference set [31]. For example, distinguishing positive interactions from negative ones on the basis of the performance of training interactions in the data sets under analysis is contingent upon the quality of the reference training sets. Another problem is that most multi-source data are unstructured but often correlated. However, the correlation can be difficult to measure because of both data incompleteness (a common problem) and sampling biases.

Another important shortcoming of most function prediction methods is that they do not take all annotations and their relationships into account. Instead, they have either used arbitrarily chosen functional categories from one level of annotation hierarchy or some arbitrarily chosen so-called informative functional categories based on some *ad hoc* thresholds. Such arbitrarily chosen functional categories only cover a small portions of the whole annotation hierarchy, making

the predictions less comprehensive and hard to compare. Predicting functions using the entire annotation system hierarchy is necessary and is a main focus of this chapter.

For another focus of this chapter, i.e., functional gene network generation, the challenges lie in the heterogeneity of the data and the systematic bias of each method. Methods have their own sensitivity and specificity for relationships between proteins and only investigate limited aspects of such relationships. Another problem is that most multi-source data are often correlated but the degrees of correlation are hard to estimate, hence making integration of different data difficult.

In this chapter, we propose methods to address the problems mentioned above using GO information.

For the problem of gene function prediction, we hypothesize that functionally related proteins have similar GO annotations. Since similarity can be measured, we further hypothesize that the distribution of similarity values of pairs of proteins can be modeled as a sum of two log-normal distributions (i.e., a mixture model) representing two populations – one representing pairs of proteins that interact with high support (high confidence), and the other representing pairs that interact with low support (low confidence) (Section 3.1). The parameters of the mixture model were then estimated from a large database. This mixture model was then used to differentiate interactions with high confidence from the ones that have low confidence, and was integrated into the function prediction methods. A new evaluation method was also proposed to evaluate the predictions (see Section 3.4). The new evaluation method captures the similarity between GO terms and reflects the relative hierarchical positions of predicted and true function assignments.

For the problem of generating functional gene network, we present a novel approach to automatically generate a functional network of yeast genes using GO annotations. The semantic similarity (SS) is calculated between pairs of genes. This SS score is then used to predict linkages between genes, to generate a functional network. GO annotations are generated by integrating information from multiple data sources, many of which have been manually curated by human experts. Thus GO annotations can be viewed as a way in which unstructured multiple data sources are integrated into a structured single source. It is therefore a valuable data source for inferring functional gene networks. We hypothesized that functional relationships between two genes are proportional to the semantic similarities between them. We determined semantic similarities between all pairs of genes in yeast to generate a whole genome functional network of yeast genes.

Note that while PPI data involves proteins, GO terms are associated with genes and their products. For the rest of this chapter, we will use the terms *genes* and their associated *proteins* interchangeably.

## 2 GO-Based Gene Similarity Measures

As described in Section 1, the semantic similarity between genes plays an important role in solving biology problems of interest. In this section, we first

introduce the concepts of semantic similarity between genes calculated based on gene ontology. Next, we investigate inherent properties of some previously known methods used to calculate such similarity.

Let us assume that a gene $A$ is associated with the following GO terms $\{t_{a1}, ..., t_{ai}\}$, and that a gene $B$ is associated with the following GO terms $\{t_{b1}, ..., t_{bj}\}$. All the methods known reduce the similarity between genes to similarity between GO terms. Thus, the similarity between genes $A$ and $B$ based on gene ontology can be defined as follows:

$$sim_X(A, B) = \max_{i,j}\{sim_X(t_{ai}, t_{bj})\}, \tag{1}$$

where $sim_X(t_{ai}, t_{bj})$ is the similarity between the GO terms $t_{ai}$ and $t_{bj}$ using method $X$.

Below we discuss the various methods proposed for measuring the similarity between GO terms. These methods include the ones proposed by Resnik [2], Jiang and Conrath [3], Lin [4], and Schlicker *et al.* [5]. The methods proposed by Resnik, Jiang and Conrath, and Lin have been used in other domains and were introduced to this area by Lord *et al.* [6]. Note that the method by Jiang and Conrath proposed the complementary concept of a distance measure instead of a similarity measure.

**Resnik:**

$$sim_R(t_1, t_2) = \max_{t \in S(t_1, t_2)} \{IC(t)\} \tag{2}$$

**Jiang-Conrath:**

$$dist_{JC}(t_1, t_2) = \min_{t \in S(t_1, t_2)} \{IC(t_1) + IC(t_2) - 2IC(t)\} \tag{3}$$

**Lin:**

$$sim_L(t_1, t_2) = \max_{t \in S(t_1, t_2)} \left\{ \frac{2IC(t)}{IC(t_1) + IC(t_2)} \right\} \tag{4}$$

**Schlicker:**

$$sim_S(t_1, t_2) = \max_{t \in S(t_1, t_2)} \left\{ \frac{2IC(t)}{IC(t_1) + IC(t_2)} (1 + IC(t)) \right\}. \tag{5}$$

Here $IC(t)$ is the information content of term $t$:

$$IC(t) = -\log(p(t)), \tag{6}$$

where $p(t)$ is defined as $freq(t)/N$, $freq(t)$ is the number of genes associated with term $t$ or with any child term of $t$ in the data set, $N$ is total number of genes in the genome that have at least one GO term associated with them, and $S(t_1, t_2)$ is the set of common subsumers of the terms $t_1$ and $t_2$.

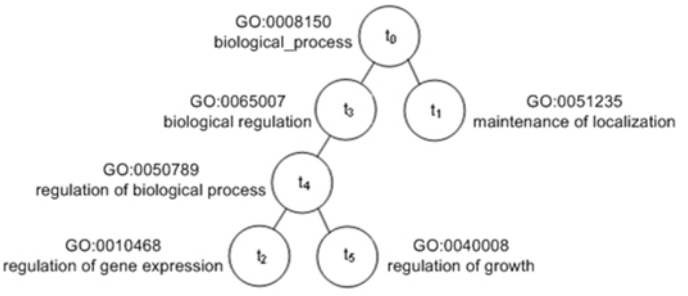



**Fig. 3.** An example showing the hierarchy of sample GO terms

The basic objective of these methods is to capture the specificity of each GO term and to calculate the similarity between two GO terms in a way that reflects their positions in the GO hierarchy. However, as discussed below, we argue that the methods of Lin and Jiang-Conrath have serious shortcomings. For example, consider the non-root terms $t_2$ (GO:0010468) and $t_3$ (GO:0065007) in Figure 3. Clearly, $dist_{JC}(t_2, t_2) = dist_{JC}(t_3, t_3) = 0$, and $sim_L(t_2, t_2) = sim_L(t_3, t_3) = 1$. In other words, the methods of Lin and Jiang-Conrath cannot differentiate between two pairs of genes, one of which has both genes from the pair associated with the term $t_2$ (GO:0010468), and the other has both genes from the pair associated with $t_3$ (GO:0065007). It ignores the fact that $t_2$ (GO:0010468, regulation of gene expression) is more specific than $t_3$ (GO:0065007, biological regulation). In contrast, if Resnik method is used, then $sim_R(t_2, t_2) = -\log p(t_2) > sim_R(t_3, t_3) = -\log p(t_3)$, if $t_2$ is more specific than $t_3$. Thus it reflects the relative positions (and the specificities) of $t_2$ and $t_3$ in the GO hierarchy. For example, in *Saccharomyces cerevisiae*, genes *YCR042C* and *YMR227C* encode TFIID subunits. Both are annotated with GO terms GO:0000114 (G1-specific transcription in mitotic cell cycle) and GO:0006367 (transcription initiation from RNA polymerase II promoter). According to the definition, $sim_L$(YCR042C, YMR227C) = 1 and $dist_{JC}$(YCR042C, YMR227C) = 0. Now consider another pair of genes *YCR046C* and *YOR063W*, both of which encode components of the ribosomal large subunits, however, one is mitochondrial and the other cytosolic. Both are annotated with the GO term GO:0006412 (translation). Again, according to the definition, $sim_L$(YCR046C, YOR063W)=1 and $dist_{JC}$(YCR046C, YOR063W)= 0. Thus, we have

$$sim_L(\text{YCR042C}, \text{YMR227C}) = \text{sim}_\text{L}(\text{YCR046C}, \text{YOR063W}) = 1, \text{and}$$

$$dist_{JC}(\text{YCR042C}, \text{YMR227C}) = \text{dist}_\text{JC}(\text{YCR046C}, \text{YOR063W}) = 0.$$

But clearly, the annotations of genes YCR042C and YMR227C are much more specific than the annotations of genes YCR046C and YOR063W. This is evidenced from the fact that GO:000014 and GO:0006367 have seven and 44

genes annotated by those terms, while GO:0006412 has 687 genes annotated by that term. So the similarity between genes YCR042C and YMR227C should be greater than the similarity between genes YCR046C and YOR063W. The similarity between genes calculated by the method of Resnik reflects this fact, since

$$sim_R(\text{YCR042C}, \text{YMR227C}) = -\log \text{p}(\text{GO} : 0000114) = 9.69$$
$$> sim_R(\text{YCR046C}, \text{YOR063W}) = -\log \text{p}(\text{GO} : 0006412) = 4.02.$$

The advantages of using GO information is that it has hierarchical structure. However, most previous research treat each GO terms equally and ignore the properties of GO hierarchy. The semantic similarity between GO terms provides us a means to explore the entire GO hierarchy. Furthermore, the semantic similarity between genes derived from the semantic similarity between GO terms the genes associated to provides us new perspective to investigate large-scale biological data, such as PPI data. In the following two Sections (Section 3 and Section 4), we show how GO information is used to estimate the reliability of PPI data, thus improve the function prediction, and show that GO is also a valuable data source for inferring functional gene networks.

## 3 Estimating Support for PPI Data with Applications to Function Prediction

### 3.1 Mixture Model of PPI Data

As mentioned earlier, PPI data generated using high throughput techniques contain a large number of false positives [32]. Thus the PPI data set contains two groups, one representing true positives and the other representing false positives. However, differentiating the true and false positives in a large PPI data set is a big challenge due to the lack of good quantitative measures. We assume that the similarity measure suggested in Section 2 is a good measure. However, we are still left with differentiating between high similarity values and low similarity values. An *ad hoc* threshold can be used for the differentiation, but is not desirable. Our proposed method avoids such choices. Instead, we propose a mixture model to differentiate the two groups in a large PPI data set. One group contains pairs of experimentally detected interacting proteins with strong support from the similarity of their GO annotations, and the other contains pairs of experimentally detected interacting proteins that have weak or unknown support. Here we provide evidence that the similarity between genes based on Gene Ontology using the method of Resnik (see Equation (2)) helps to differentiate between the two groups in the PPI data. As mentioned earlier, this is based on the assumption that the true positives will have higher gene similarity values than the false positives. A mixture model is used to model the distribution of the similarity values (using the Resnik method for similarity of *Biological Process* GO terms). In particular, the mixture model suggests that

$$p(x) = w_1 p_1(x) + w_2 p_2(x), \tag{7}$$

where $p_1(x)$ is the probability density function for the similarity of pairs of genes with true interactions among the experimentally detected interactions, and $p_2(x)$ is the probability density function for the similarity of pairs of genes in the false positives; $w_1$ and $w_2$ are the weights for $p_1$ and $p_2$, respectively. Given a large data set, $p_1$, $p_2$, $w_1$, and $w_2$ can be inferred by the maximum likelihood estimation (MLE) method. For our case, we conclude that the similarity of pairs of genes can be modeled as a mixture of two log-normal distributions with probability density functions

$$p_1(x) = \frac{1}{\sqrt{2\pi}\sigma_1 x} exp\left(-\frac{(\log x - \mu_1)^2}{2\sigma_1^2}\right) \tag{8}$$

and

$$p_2(x) = \frac{1}{\sqrt{2\pi}\sigma_2 x} exp\left(-\frac{(\log x - \mu_2)^2}{2\sigma_2^2}\right). \tag{9}$$

After parameter estimation, we can calculate a value $s$ such that for any $x > s$, $p(x \in \text{Group 2}) > p(x \in \text{Group 1})$. This value $s$ is the threshold meant to differentiate the pairs of proteins in PPI data with high support (Group 2) from those with low support (Group 1). The further away the value is from $s$, the greater is the confidence. Furthermore, the confidence can be measured by computing the $p$-value since the parameters of distribution are given by the model.

Thus our mixture model suggests a way of differentiating the true positives from the false positives by only looking at the similarity value of pairs of genes (using the method of Resnik in Equation (2) for similarity of *Biological Process* GO terms), and by using a threshold value specified by the model (Group 1 contains false positives and Group 2 contains the true positives). Note that no *ad hoc* decisions are involved.

### 3.2 Data Sets

Function prediction methods based on a protein-protein interaction network can make use of two data sources - the PPI data set and a database of available functional annotations. Next, we introduce the two data sources used in our experiments.

**Gene Ontology.** We used the available functional annotations from the Gene Ontology (GO) database [35], as introduced in Chapter **??**. Generally, a gene is annotated by one or more GO terms. The terms at the lower levels correspond to more specific functional descriptions. If a gene is annotated with a GO term, it is also annotated with the ancestors of that GO term. Thus, the terms at the higher

levels have more associated genes. The GO database is constantly being updated; we used version 5.403, and the gene-term associations for *Saccharomyces cerevisiae* from version 1.1344 from SGD (`http://www.yeastgenome.org/`).

**Protein-Protein Interaction Data.** Several PPI data sets were used in this chapter for our experiments. The first PPI data set was downloaded from the BioGRID database [36]. Henceforth, we will refer to this data set as the *BioGRID* data set. The *confirmation number* for a given pair of proteins is defined as the number of independent confirmations that support that interaction. A *pseudo-negative* data set was also generated by picking pairs of proteins that were not present in the PPI data set. Thus each pair of proteins in the pseudo-negative data set has a confirmation number of 0. There were 87920 unique interacting pairs in total with confirmation numbers ranging from 0 to 40. This data set was used to estimate the metric of support for all pairs of proteins.

Two so-called *gold-standard data sets* (gold-standard positive and gold-standard negative) were used to test the performance of our method. These two gold-standard data sets were hand-crafted by Jansen *et al.* [26]. The gold-standard positives came from the MIPS (Munich Information Center for Protein Sequence) complexes catalog [37] since the proteins in a complex are strongly likely to interact with each other. The number of gold-standard positive pairs used in our experiments was 7727. A gold-standard negative data set is harder to define. Jansen *et al.* created such a list by picking pairs of proteins known to be localized in separate subcellular compartments [26], resulting in a total of 1838501 pairs.

### 3.3 Function Prediction

The major advantage of the method presented in Section 3.1 is that the $p$-values obtained from the mixture model provide us with a metric of support or a reliability measure for the PPI data set. However, the limitation of our technique is that it can only be applied to pairs of genes with annotations. In order to overcome this limitation, it is sensible to first perform function prediction to predict the functional annotation of unannotated genes. As mentioned earlier, many computational approaches have been developed for this task [7]. However, the prediction methods are also prone to high false positives. Schwikowski *et al.* proposed the *Majority-Voting* (MV) algorithm for predicting the functions of an unannotated gene $u$ by an optimization method using the following objective function:

$$\alpha_u = \arg\max_{\alpha} \sum_{v \in N(u), \alpha_v \in A(v)} \delta(\alpha_v, \alpha), \tag{10}$$

where $N(u)$ is the set of neighbors of $u$, $A(u)$ is the set of annotations associated with gene $u$, $\alpha_i$ is the annotation for gene $i$, $\delta(x, y)$ is a function that equals 1 if $x = y$, and 0 otherwise [8]. In other words, gene $u$ is annotated with the term $\alpha$ associated with the largest number of its neighbors. The main weakness of

this conventional majority voting algorithm is that it weights all its neighbors equally, and is prone to errors because of the high degree of false positives in the PPI data set. Using the metric of support proposed in section 3.1, we propose a modified *"Reliable" Majority-Voting* (RMV) algorithm which assigns a functional annotation to an unannotated gene $u$ based on the following objective function, which is a weighted version of Equation (10):

$$\alpha_u = \arg\max_{\alpha} \sum_{v \in N(u), \alpha_v \in A(v)} w_{v,u} \delta(\alpha_v, \alpha), \tag{11}$$

where $w_{v,u}$ is the reliability of the interaction between genes $v$ and $u$, that is, $w_{v,u} = sim(A(v), \{\alpha\})$.

Another weakness of the conventional MV algorithm is that it only allows exact matches of annotations and will reject even approximate matches of annotations. Here we propose the *Weighted Reliable Majority-Voting* (WRMV) method, a modification of RMV, with the following objective function

$$\alpha_u = \arg\max_{\alpha} \sum_{v \in N(u)} w_{v,u} \big( \max_{\alpha_v \in A(v)} sim(\alpha_v, \alpha) \big), \tag{12}$$

where $sim(x, y)$ is a function that calculates the similarity between the GO terms $x$ and $y$.

Note that the algorithms proposed above only predict one functional annotation term for an uncharacterized gene. But they can be adapted to predict $k$ functional annotation terms for any uncharacterized gene by picking the $k$ best values of $\alpha$ in each case.

The example in Figure 4 illustrates the necessity of considering both the metric of support for the PPI data and the relationships between GO terms during function prediction. Assume we need to predict functions for a protein $u$, whose neighbors in the interaction network include proteins $v_1$, $v_2$, $v_3$, $v_4$, $v_5$, and $v_6$. As shown in Figure 4, suppose proteins $v_1$ and $v_2$ are annotated with GO term $t_2$, $v_3$ and $v_4$ with GO term $t_4$, and $v_5$ and $v_6$ with GO term $t_5$. According to the MV algorithm, protein $u$ will be assigned all the GO terms $t_2$, $t_4$, and $t_5$, since each of the three terms has equal votes (2 in this case). However, as can be seen from Figure 4, GO term $t_5$ is more specific than GO terms $t_2$ and $t_4$. So GO term $t_5$ should be the most favored as an annotation for protein $u$, assuming that all the PPI data are equally reliable.

Note that the metric of support can also be used to improve other approaches besides the MV algorithm. In this chapter, we have employed only local approaches, because as argued by Murali *et al.*, methods based on global optimization do not perform better than local approaches based on majority-voting algorithm [38].

### 3.4 Evaluating the Function Prediction

Several measures are possible in order to evaluate the function prediction methods proposed in Section 3.3. For the traditional cross-validation technique, the

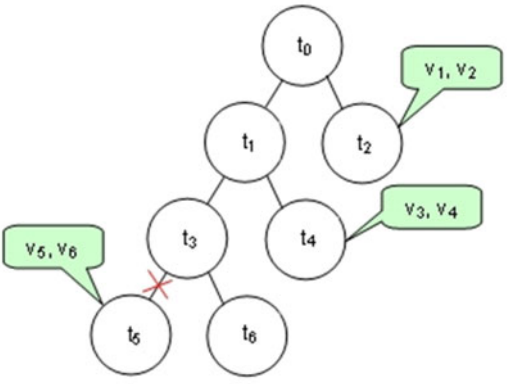



**Fig. 4.** An example showing the hierarchy of GO terms associated with a set of genes. GO term $t_2$ is associated with genes $v_1$ and $v_2$; GO term $t_4$ is associated with genes $v_3$ and $v_4$; GO term $t_5$ is associated with genes $v_5$ and $v_6$.

simplest method to perform an evaluation is to use *precision* and *recall*, defined as follows:

$$Precision = \frac{\sum_i k_i}{\sum_i m_i}, \quad Recall = \frac{\sum_i k_i}{\sum_i n_i}, \tag{13}$$

where $n_i$ is the number of known functions for the protein $i$, $m_i$ is the number of predicted functions for the protein $i$ when hiding its known annotations, and $k_i$ is the number of matches between known and predicted functions for protein $i$. The conventional method to count the number of matches between the annotated and predicted functions only considers the exact overlap between predicted and known functions, ignoring the structure and relationship between functional attributes. Once again,using the example illustrated in Figure 4, assume that the correct functional annotation of a protein $u$ is GO term $t_4$, while term $t_1$ is the only function predicted for it. Then both recall and precision would be reported to be 0 according to the conventional method. However, it overlooks the fact that GO term $t_4$ is quite close to the term $t_1$. Here we introduce a new definition for precision and recall. For a known protein, suppose the known annotated functional terms are $\{t_{o1}, t_{o2}, ..., t_{on}\}$, and the predicted terms are $\{t_{p1}, t_{p2}, ..., t_{pm}\}$. We define the success of the prediction for function $t_{oi}$ as

$$RecallSucess(t_{oi}) = \max_j sim(t_{oi}, t_{pj}),$$

and the success of the predicted function $t_{pj}$ as

$$PrecisionSuccess(t_{pj}) = \max_i sim(t_{oi}, t_{pj}).$$

We define the new precision and recall measures as follows:

$$Precision = \frac{\sum_j PrecisionSuccess(t_{pj})}{\sum_j sim(t_{pj}, t_{pj})}, \tag{14}$$

$$Recall = \frac{\sum_i RecallSucess(t_{o_i})}{\sum_i sim(t_{o_i}, t_{o_i})}. \tag{15}$$

Note that Equation (14) and Equation (15) are straightforward generalization of Equation (13), with similarity measures used instead of set overlap.

### 3.5 Experimental Results

**Results on using the Mixture Model.** The similarity between genes based on the *Biological Process* categorization of the GO hierarchy was calculated using Equation (1) and Equation (2). The method was separately applied to the *BioGRID* data set, where we confined ourselves to PPI data that had non-negative, integral confirmation number $k$ (for some user-specified number $k$). Interacting pairs of proteins from *BioGRID* data set were grouped based on their confirmation number. We hypothesized that each of these groups generated above contains two subgroups, one representing pairs of proteins that are experimentally observed to interact and that have high support from their GO annotations, and the other representing pairs that are experimentally observed interact with low support from their GO annotations.

As shown in Figure 5, a histogram of the (logarithm of) similarity measure (using the *Resnik* method for similarity of GO terms) was plotted for pairs of genes with a specified degree of confirmation from the PPI data set. In order to visualize the whole histogram, we have arbitrarily chosen $\log(0) = \log(0.01) \approx -4.61$. Based on our earlier assumptions, we conjectured that each of these histograms can be modeled as a mixture of two normal distributions (since the original is a mixture two log-normal distribution), one for the Group 1, and the other for the Group 2.

All the plots in Figure 5 have three well-separated subgroups. Note that the leftmost subgroup corresponds to those pairs of genes for which at least one has the GO terms associated with the root of the GO hierarchy; the subgroup in the middle corresponds to those pairs of genes at least one of which is associated with a node close to the root of the GO hierarchy. The reason for the existence of these two subgroups is that there are some PPI data sets containing genes with very non-specific functional annotations. As we can see from Figure 5, the larger the confirmation number, the less pronounced are these two groups. Thus, for the two leftmost subgroups, similarity of genes based on GO annotation cannot be used to differentiate signal from noise. (Thus function prediction for these genes are necessary and is an important focus of this chapter.) However, for PPI data containing genes with specific functions (i.e., the rightmost group in the plots of Figure 5), similarity of genes in this group was fitted to a mixture model as described in Section 3.1. In fact, a fit of the rightmost group with two

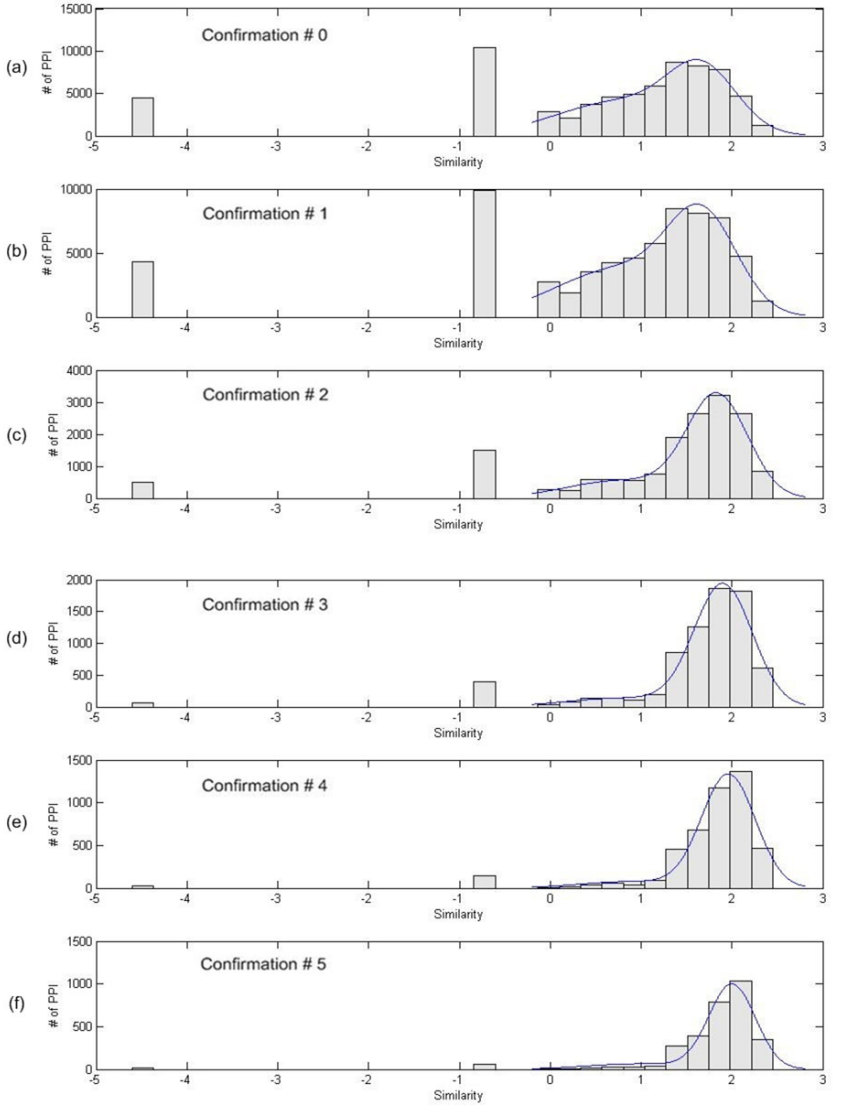



**Fig. 5.** Distribution of similarity of yeast genes based on the Resnik method. This was achieved by using (a) 0 or more independent confirmations (i.e., the entire PPI data set), (b) 1 or more independent confirmations, (c) 2 or more independent confirmations, (d) 3 or more independent confirmations, (e) 4 or more independent confirmations, and (f) 5 or more independent confirmations.

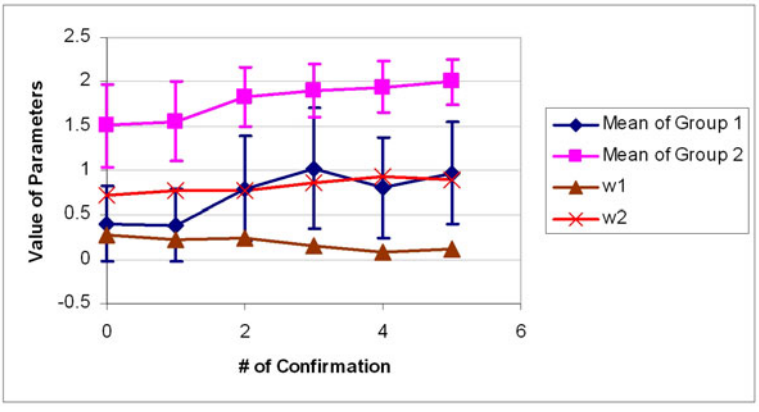



**Fig. 6.** Parameters for the density function for yeast PPI data. This was achieved by fitting $p(x) = w_1 p_1(x) + w_2 p_2(x)$ for the metric of support for yeast PPI data with different confirmation numbers. $w_1$ and $w_2$ are the weights for $p_1$ and $p_2$, respectively. Group 1 corresponds to noise, and Group 2 to signal.

normal distributions is also shown in the plots of Figure 5. The fit is excellent (with R-squared value more than 98 percent for the data set with confirmation number 1 or more). The details are shown in Figure 6. We are particularly interested in the fit of the data set with confirmation 1 and above. The estimated parameters are $\mu_1 = 0.3815$, $\sigma_1 = 0.4011$, $\mu_2 = 1.552$, $\sigma_2 = 0.4541$, $w_1 = 0.23$, and $w_2 = 0.77$. From the fit, we can calculate a value $s = 0.9498$ such that for any $x > s$, $p(x \in \text{Group 2}) > p(x \in \text{Group 1})$. This is the threshold we used to differentiate the two groups. The further away the point is from $s$, the greater the confidence. Furthermore, the confidence can be measured by computing the $p$-value since the parameters of the distribution are known. Further investigation of these two groups reveal that protein pairs in Group 2 contain proteins that have been well annotated (associating with GO terms that have levels larger or equal to 3). The components of Group 1 are more complicated. It consists of the interactions between two poorly annotated genes, the interactions between a well annotated gene and a poorly annotated gene, and the interactions between two well annotated genes.

The results of further experiments performed on the PPI data sets from the human proteome [36] also displayed similar results (Figures 7 and 8).

To test the power of our estimation, we applied it to the gold-standard data set. In particular, for each pair of genes in the gold-standard data set, we calculated the similarity between the genes in that pair and compared it to the threshold value $s = 0.9498$. If the similarity was larger than $s$, we labeled it as Group 2, otherwise, as Group 1. We then calculated the percentage of pairs of proteins in Group 2 and Group 1 in the gold-standard positive and negative data sets.

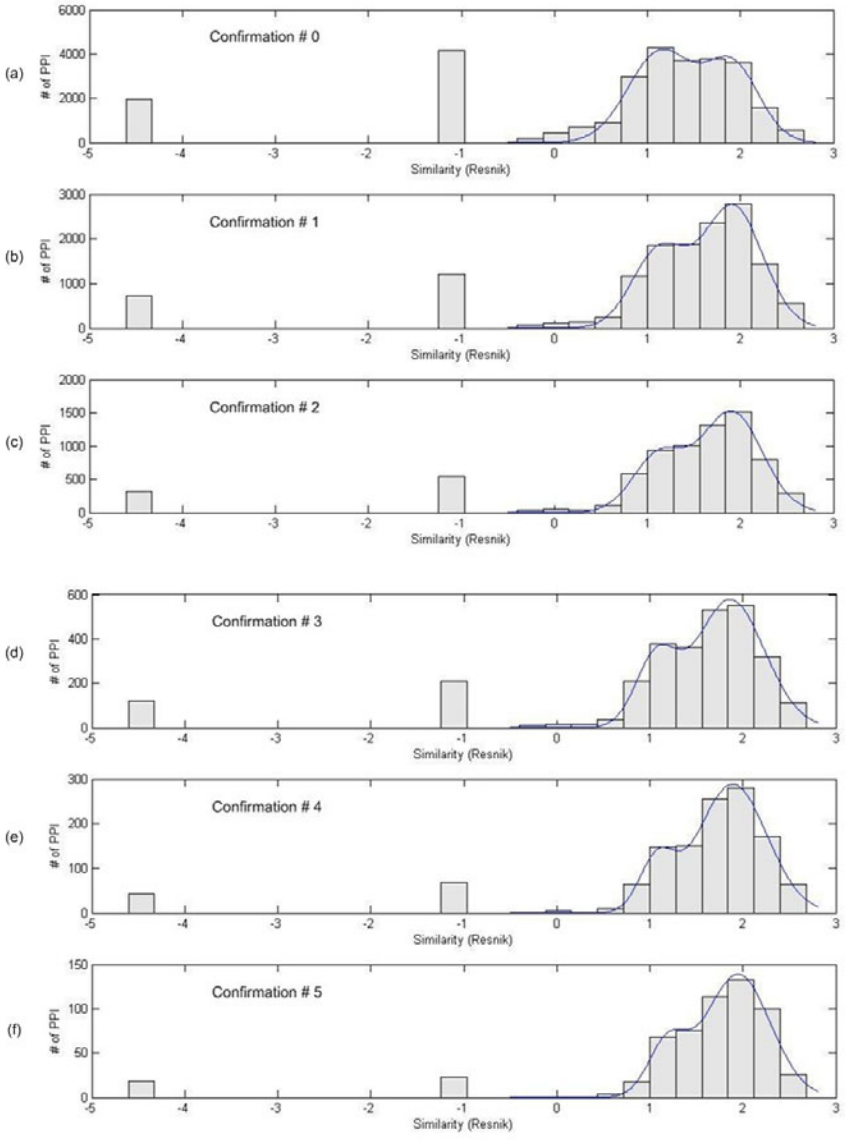



**Fig. 7.** Distribution of similarity of human genes based on the Resnik method. This was achieved by using: (a) 0 or more independent confirmations (i.e., the entire PPI data set), (b) 1 or more independent confirmations, (c) 2 or more independent confirmations, (d) 3 or more independent confirmations, (e) 4 or more independent confirmations, and (f) 5 or more independent confirmations.

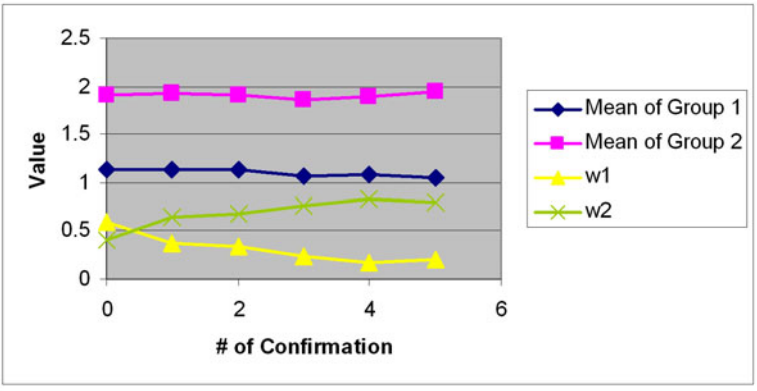



**Fig. 8.** Parameters for the density function for human PPI data. This was achieved by fitting $p(x) = w_1 p_1(x) + w_2 p_2(x)$ for the metric of support for human PPI data with different confirmation numbers. $w_1$ and $w_2$ are the weights for $p_1$ and $p_2$, respectively. Group 1 corresponds to noise, and Group 2 to signal.

**Table 1.** Mixture model on gold-standard data set

| | total PPI pairs | subgroup PPI pairs | percentage |
|---|---|---|---|
| GSPD[a] | 7727 | 7696[1] | 99.61 |
| GSND[b] | 1838501 | 1526467[2] | 83.03 |

[a] Golden Standard Positive Data set.
[b] Golden Standard Negative Data set.
[1] Number of PPI pairs in Group 2.
[2] Number of PPI pairs in Group 1.

As shown in Table 1, majority of the pairs in the gold-standard positive data (GSPD) set were labeled correctly as Group 2 (99.61%), and most of the pairs in the gold-standard negative data set (GSND) were correctly labeled as Group 1 (83.03%). These high percentage values provide further support for our mixture-model based technique. It is worth pointing out that the GSPD set is clearly more reliable than the GSND set as described in section 3.2.

One possible objection to the application in this chapter is that the results of the mixture model is an artifact of functional bias in the PPI data set. To address this objection, we applied the mixture model to PPI data after separating out the data from the three main different high-throughput methods, i.e., yeast two-hybrid systems, mass spectrometry, and synthetic lethality experiments. As reported by Mering *et al.* [32], the overlap of PPI data detected by the different methods is small, and each technique produces a unique distribution of interactions with respect to functional categories of interacting proteins. In other words, each method tends to discover different types of interactions. For example, the yeast two-hybrid system largely fails to discover interactions between proteins involved in translation; mass spectrometry method predicts relatively few interactions for proteins involved in transport and sensing.

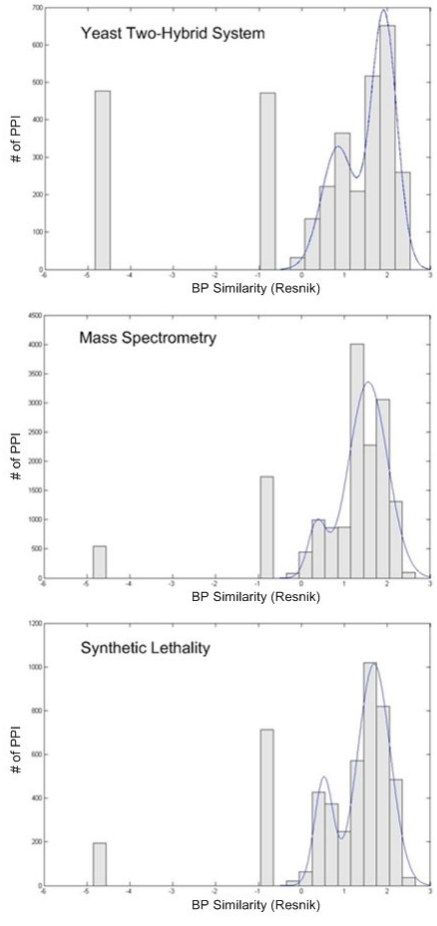



**Fig. 9.** Distribution of similarity of pairs of genes based on the method by Resnik for PPI data generated by different methods. The methods include high-throughput methods yeast two-hybrid systems (top), mass spectrometry (middle), and Synthetic Lethality (bottom).

In summary, if the PPI data set has a functional bias, then the PPI data produced by individual methods should have an even greater functional bias, with each one biased toward different functional categories.

Despite the unique functional bias of each method, the mixture model when applied to the PPI data from the individual methods showed the same bimodal mixture distribution (Figure 9), indicating that the mixture model is tolerant to severe functional bias and is therefore very likely to be a true reflection of inherent features of the PPI data.

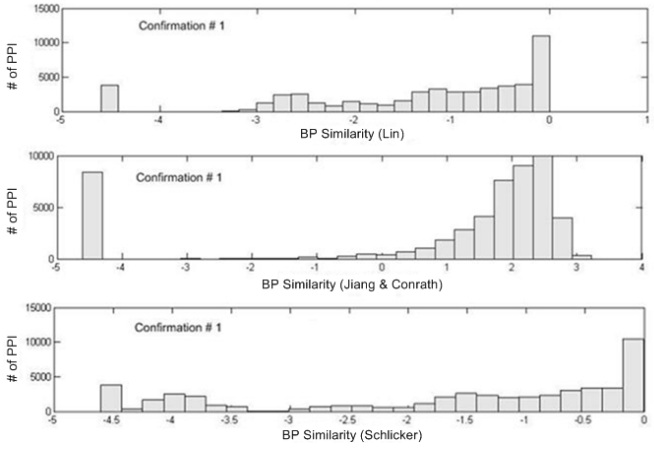



**Fig. 10.** Distribution of similarity of yeast genes based on method Lin, Jiang-Conrath, and Schlicker for yeast PPI data with confirmation number of 1 and more (Confirmation # 1)

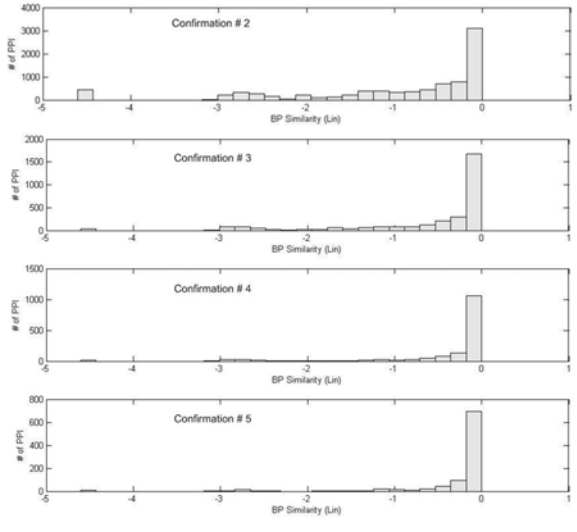



**Fig. 11.** Distribution of similarity of yeast genes based on method Jiang-Conrath. This was achieved by using PPI data with confirmation number of 2 and more (Confirmation # 2), 3 and more (Confirmation # 3), 4 and more (Confirmation # 4), and 5 and more (Confirmation # 5)

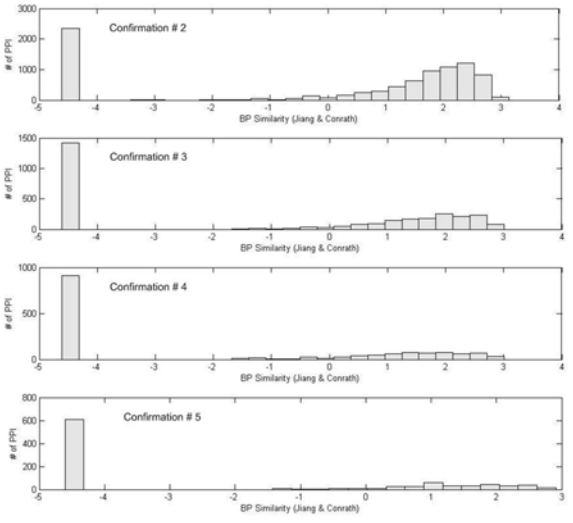



**Fig. 12.** Distribution of similarity of yeast genes based on method Jiang-Conrath. This was achieved by using PPI data with confirmation number of 2 and more (Confirmation # 2), 3 and more (Confirmation # 3), 4 and more (Confirmation # 4), and 5 and more (Confirmation # 5)

In order to justify our choice of the Resnik similarity measure, we also applied the Lin (Equation (4)), Jiang-Conrath (Equation (3)), and Schlicker (Equation (5)) methods to the PPI data set with confirmation number 1 or more. The results shown in Figures 10 confirm our theoretically-supported claim that the Lin and Jiang-Conrath methods are inappropriate for similarity computation.

As shown in Figure 10, the histogram of similarity values between genes calculated by Lin's formula has a peak at the rightmost end. Additionally, the rest of the histogram fails to display a bimodal distribution, which is necessary to flush out the false positives. Furthermore, the peaks increase with increasing confirmation number (Figures 11, 12, and 13), which is contrary to our intuition. The histograms of distance measures between genes calculated by the Jiang-Conrath's method (middle plot in Figures 10) produces a peak at its leftmost end with a unimodal distribution for the remaining, thus showing that the mixture model is unlikely to produce meaningful results. Schlicker's method was devised to combine Lin's and Resnik's methods. However, its performance was similar to that of Lin's method (see Figure 10).

We also applied these methods to the same PPI data set, but with higher confirmation numbers (Figures 11, 12, and 13). Since those data sets are likely to have fewer false positives, it is no surprise that the histograms were even less useful for discriminatory purpose.

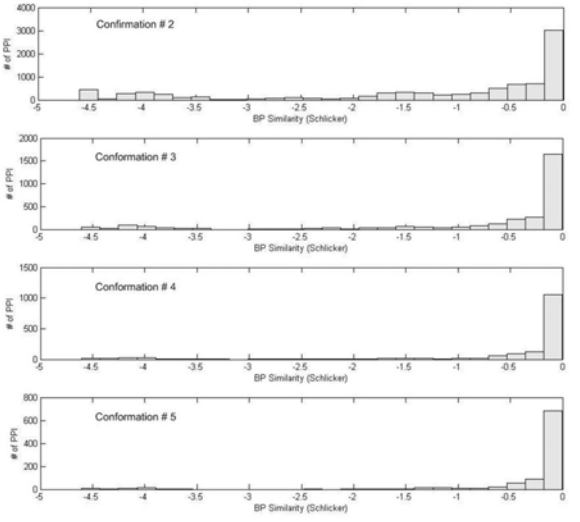



**Fig. 13.** Distribution of similarity of yeast genes based on method Schlicker. This was achieved by using PPI data with confirmation number of 2 and more (Confirmation # 2), 3 and more (Confirmation # 3), 4 and more (Confirmation # 4), and 5 and more (Confirmation # 5)

Finally, we tried our methods on the other two GO categorizations, i.e., *cellular component* and *molecular function*. Since those categorizations are less comprehensive with a large number of unannotated genes, similarity calculations based on the them did not adequately reflect the reliability of PPI data (Figures 14).

**Function Prediction.** Five different function prediction approaches based on neighborhood-counting were compared: three introduced in section 3.3 called MV (see Equation(10)), RMV (see Equation(11)), and WRMV (see Equation(12)), and two introduced in section 1.1 called Chi-Square based method (CS) developed by Hishigaki *et al.* [9] and FS-Weighted Averaging method (WA) developed by Hua *et al.* [10].

The precision and recall for each approach was calculated on the *BioGRID* PPI data set using five-fold cross validation. First, a conventional evaluation method was employed, which consisted of computing precision and recall as a simple count of the predictions for the gold-standard positive and negative sets (see Equation (13)). The results are shown in Figure 15. Then, an improved method, as described in Equations (14) and (15) was used, and the results are shown in Figure 16. In order to see the effectiveness of the new evaluation metric, the precision-recall curves of the three function prediction methods (RMV, WRMV and WA) using new and conventional evaluation metrics were compared by combining the related curves from Figures 15 and 16 and displayed in Figure 17.

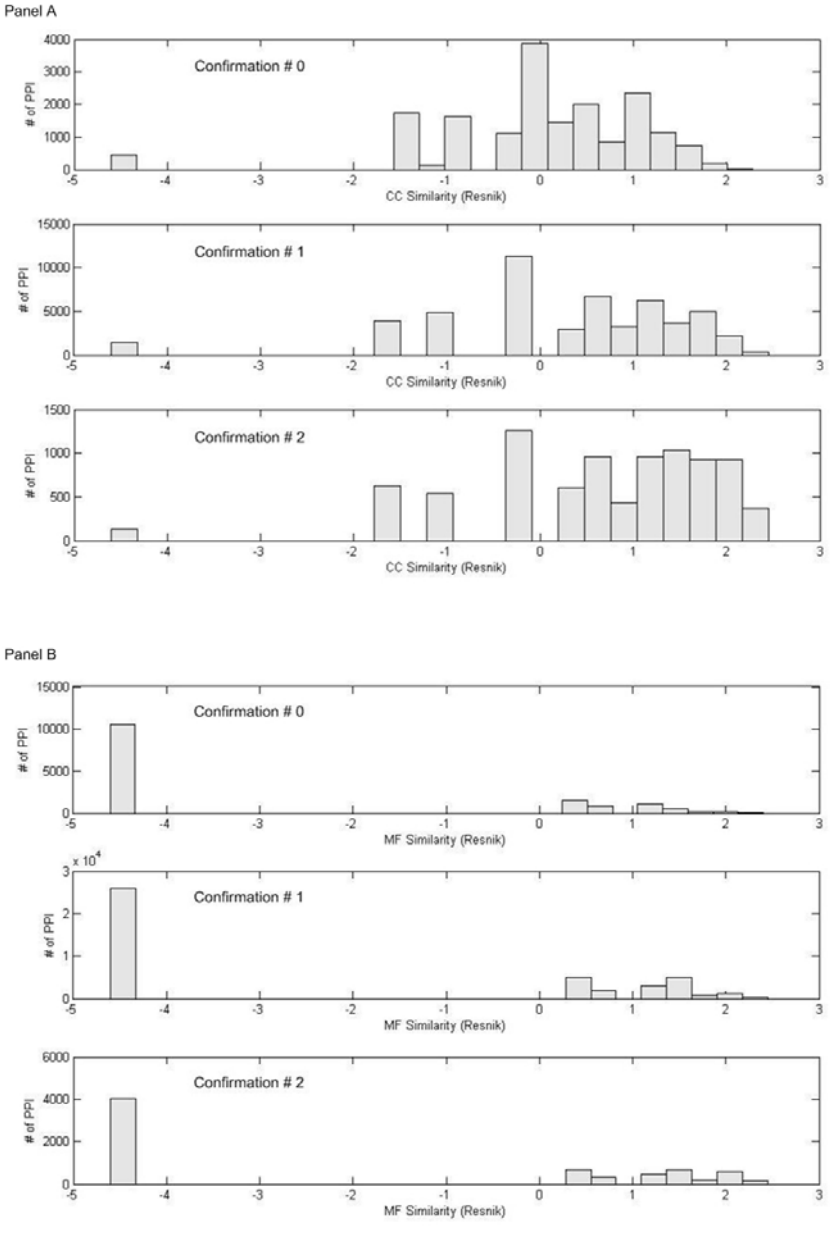



**Fig. 14.** Distribution of similarity of yeast genes based on the Resnik method using PPI data with confirmation number of 0 and more (Confirmation # 0), 1 and more (Confirmation # 1), and 2 and more (Confirmation # 2). Panel A: based on cellular component GO hierarchy; Panel B: based on molecular function GO hierarchy

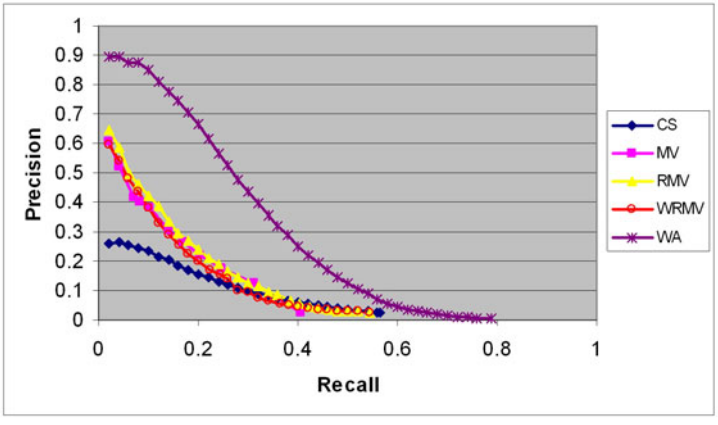



**Fig. 15.** Precision-recall analysis of five function prediction methods. This was achieved by using the conventional evaluation metric as described in Eq.(13) for 1) Chi-Square method (CS), 2) Majority-Voting method (MV), 3) Reliable Majority-Voting method (RMV), 4) Weighted Reliable Majority-Voting (WRMV), and 5) FS-Weighted Averaging method (WA)

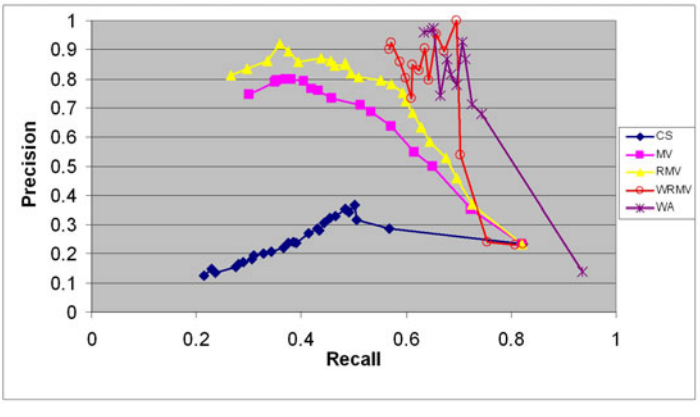



**Fig. 16.** Precision-recall analysis of five function prediction methods using new evaluation metric as described in Eq.(14) and Eq.(15) for 1) Chi-Square method (CS), 2) Majority-Voting method (MV), 3) Reliable Majority-Voting method (RMV), 4) Weighted Reliable Majority-Voting method (WRMV), and 5) FS-Weighted Averaging method (WA)

### 3.6 Discussion

As shown in Figure 15, when conventional evaluation methods were applied to calculate the precision and recall, the FS-Weighted Averaging (WA) method performed the best, and there was no significant difference among the other three

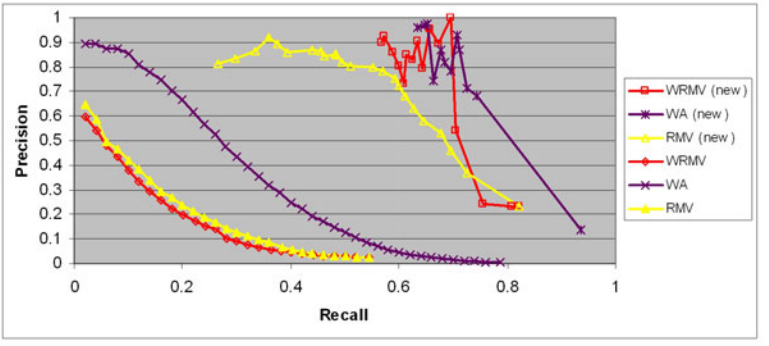



**Fig. 17.** Comparison of precision-recall analysis of three Majority-Voting function prediction methods using new evaluation metric as described in Eq.(14) and Eq.(15) for 1) Weighted Reliable Majority-Voting method (WRMV new), 2) FS-Weighted Averaging method, (WA new), and 3) Reliable Majority-Voting method (RMV new), and conventional metric as described in Eq.(13) for 4) Weighted Reliable Majority-Voting method (WRMV), 5) FS-Weighted Averaging method, (WA), and 6) Reliable Majority-Voting method (RMV)

methods (MV, RMV, and WRMV). However, when the new evaluation method (as described in Equation (14) and Equation (15)) was applied, both WA and WRMV performed well (see Figure 16). Among the three versions of Majority-Voting methods (MV, RMV, and WRMV), Weighted Reliable Majority-Voting method performed the best, and the conventional Majority-Voting method performed the worst.

The proposed new evaluation method has two advantages over the conventional one. First, the new evaluation method provides wider precision and recall coverage, that is, at the same precision (recall) value, the recall (precision) calculated by the new method is larger than that calculated by the old one. This is due to the strict definition of conventional precision and recall, while ignoring the fact that some pairs of true and predicted annotations are very similar to each other. Second, the new evaluation method has enhanced power to compare the performance of function prediction methods. For example, the precision-recall curves of the function prediction methods RMV and WRMV diverge based on the new evaluation metric, but are roughly indistinguishable based on the conventional metric.

Function predictions based on PPI data were performed using two sources of data: GO annotation data and BioGRID PPI data. Previous research on this topic focused on the interaction network inferred from PPI data, while ignoring the topology of the hierarchy representing the annotations. In some cases, only a fraction of the terms were used. Thus the resulting predictions were not comparable. For PPI data, quantitatively assessment of confidence for pairs of proteins becomes a pressing need.

# 4 A Functional Network of Yeast Genes Using Gene Ontology Information

The Gene Ontology (GO) project has integrated information from multiple data sources to annotate genes to specific biological process [35]. Generating gene networks using GO annotations is a novel and alternative way in which heterogeneous data sources can be efficiently integrated. In this Section, we present a novel approach to automatically generate a functional network of yeast genes using Gene Ontology (GO) annotations. An information theory-based **s**emantic **s**imilarity (SS) was calculated between pairs of genes, details of which were presented in Section 2 and embodied in Equations (1). This SS score was then used to predict linkages between genes, to generate a functional network. An alternative approach has been proposed using a measure called **l**og **l**ikelihood **s**core (LLS) [30]. The functional networks predicted using the SS and LLS measures were compared. Unlike the SS score, the LLS score was calculated from multiple data sources directly. We discuss our experiments on generating reliable functional gene networks.

## 4.1 Data Sets

*Saccharomyces Cerevisiae Gene Set and Gene Network.* The yeast gene network is based on the verified 5,794 protein encoding open reading frames (ORFs) of the yeast genome (version dated March 2005) downloaded from the *Saccharomyces cerevisiae* Genome Database (SGD). All computations described here are based on this gene set. YeastNet version 2 was downloaded from http://www.yeastnet.org/ [31]. Our resulting yeast gene network was mapped to YeastNet version 2 for the purpose of comparison and analysis.

*Benchmark Sets.* In order to derive the log likelihood score (LLS) from multiple data sources, GO annotations, which were downloaded from the *Saccharomyces cerevisiae* Genome Database (SGD) (version dated March 2005), were used as a major reference set for benchmarking [31]. The term "protein biosynthesis" was excluded by Lee et al. because it was assigned to so many genes that it significantly biased the benchmarking. Our semantic similarity calculation between genes was also based on this GO annotation system, but we did not remove any GO terms.

We also used the Munich Information Center for Protein Sequence (MIPS) protein function annotation set [37]. We used 24 major categories from the top level of this data set.

Yeast essential ORFs were downloaded from *Saccharomyces* Genome Deletion Project web page (http://www-sequence.stanford.edu/group/yeast_deletion_project/deletions3.html) [39].

## 4.2 Constructing a Functional Gene Network

In this section, we propose the use of semantic similarity between genes calculated based on gene ontology, in order to construct a functional gene network.

We then review a previously described method to calculate the log likelihood score (LLS) of pairs of genes, which provides yet another basis for constructing functional gene networks [30, 31].

The functional gene network described here is represented as a weighted graph with the node representing a gene or protein, the edge representing the functional relationships between two nodes, and the weight of the edge representing how similar the two nodes are. The larger the weight is, the closer the two nodes are functionally related. The key step of constructing a functional gene network is to estimate the weight of the edges in the network. In this section, we first introduce a method to calculate SS between genes based on GO information. We then review the method to calculate the LSS of pairs of genes [30, 31]. Both SS and LLS are used to estimate the weight of the edges of functional gene network.

### 4.3 Using Semantic Similarity (SS)

Computing the SS value between genes was described earlier in in Section 2. The functional gene network using SS scores consists all yeast genes. Any pair of genes is functionally linked by the SS score between them.

**Using the Log Likelihood Score (LLS).** The SS value between genes using GO information can be used to infer functional associations of genes. Such functional linkages between genes can also be inferred from multiple genomic and proteomic data. As mentioned above, many approaches have been developed in this area [26–31, 33, 34, 40–44]. Lee *et al.* developed a unified scoring scheme for linkages based on a Bayesian statistics approach [30, 31]. Each source of data is evaluated for its ability to reconstruct known gene pair relationships by measuring the likelihood that pairs of genes are functionally linked conditioned on the evidence. This is calculated as a LLS score:

$$LLS = \ln \frac{P(L|D)/P(\neg L|D)}{P(L)/P(\neg L)} \tag{16}$$

where $P(L|D)$ and $P(\neg L|D)$ are the frequencies of gene linkages observed in the given data (D) sharing (L) and not sharing ($\neg L$) function annotation, respectively, whereas $P(L)$ and $P(\neg L)$ represent the prior expectations (*i.e.*, the total frequency of linkages between all annotated yeast genes sharing and not sharing function annotations, respectively). LLS scores greater than zero indicate that the experiment tends to functionally link genes, with higher scores indicating more confident linkages. In order to decide whether pairs of genes are functionally linked or not, GO annotations based on the "biological process" hierarchy were used as a reference set. The "biological process" GO annotation contains 14 different levels of GO terms. Lee *et al.* used terms belonging to levels 2 through 10 [31]. They considered a pair of genes as being functionally linked if they shared an annotation from the set of GO terms between level 2 through 10, and not linked if they did not share any term.

Note that the LLS score was calculated for each data source, rather than for each gene pair. All possible gene pairs from the same data source received the same LLS score as calculated using Equation (16)(referred to as *single* LLS). For the gene pairs appearing in multiple data sources, a weighted sum method was employed to integrate multiple LLS scores into one [31] (referred to as *integrated* LLS). Thus the functional gene network generated by LLS scores consists all the genes from multiple data sources, and linkages between pairs of genes weighted by corresponding LLS scores (single LLS or integrated LLS).

### 4.4 Evaluating the Functional Gene Network

Functional networks were generated using: (a) SS scores, (b) LLS scores from individual data sources, and (c) LLS scores from integrated multi-source data. As discussed in in Section 2, many different SS scoring schemes have been proposed. For this work, we used all four schemes (Resnik, Jiang and Conrath, Lin, and Schlicker) to compute the SS scores. Functional networks generated using SS scores and LLS scores from integrated multi-source data were assessed by comparing them with an independent reference set of functional gene linkages from the MIPS database [37]. Gene pair linkages were assessed by computing recall and precision against the reference set.

Another metric to evaluate the quality of a gene network is the correlation between a gene's tendency to be essential (referred to as "lethality") and its centrality (referred to as "centrality") in a network [31, 39, 45]. The so called centrality-lethality rule was observed by Jeong *et al.*, who demonstrated that high-degree nodes in a protein interaction network of *S. cerevisiae* contain more essential proteins than would be expected by chance [45]. A gene is *essential* if lack of it will cause the death of a cell. The *centrality* of a gene is measured as the number of interactions in which the gene participates. The basic idea is that if a gene's function is more essential to a cell, it should have more functional linkages. Thus a high correlation between lethality and centrality suggests that the gene network is biologically relevant and therefore of good quality.

### 4.5 Experimental Results

In this section, we first present the results of the correlation analysis between SS and integrated LLS. Then networks derived by different methods will be evaluated and compared. Finally, we will illustrate the effectiveness of our functional network computed using SS by considering two examples of functional modules.

**Correlation between SS and LLS.** Both SS and integrated LLS were used to capture the functional linkages of genes. If two genes are truly functionally linked, one would expect both SS and LSS values to be high. We compared SS and LLS values to investigate the degree of agreement between these two metrics. When performing the analysis on individual genes, a great dispersion in the SS (calculated by Resnik method) and LLS was observed. This resulted in a low Pearson correlation coefficient (about 0.36) when all gene pairs were-considered (see Figure 18 at interval size zero). Such a dispersion reflects the

intrinsic complexity of biological measures. Only a broad correlation can be expected because of the noise in the measurements.

In order to ascertain underlying data trends, we then applied a strategy suggested by Sevilla *et al.*, which averages the log likelihood score over uniform SS intervals [46]. The idea behind such a strategy is as follows: for a given SS interval, the correlation coefficient will follow a certain statistical distribution. The average of the correlation coefficients within a SS interval gives an estimate of the mean of the statistical distribution of correlations.

Correlation results depend on the size of the intervals chosen. Large interval sizes (hence fewer intervals) lead to higher correlations. Correlations between the integrated LLS and SS computed using four different metrics (Resnik, Jiang and Conrath, Lin, and Schlicker) were computed. Figure 18 shows the Pearson correlation coefficients between SS and LLS for different interval sizes. The correlation between SS and LLS performed best when SS was calculated using the Resnik method (Figure 18). The results confirm our analysis from Section 4.3. Using this same method, increasing the interval size from 0 to 5 improved the correlation coefficient from 0.36 to 0.61 (when SS was calculated using Resnik method). The correlation coefficients also illustrate the intrinsic differences between SS and LLS.

As shown in Figure 18, SS scores calculated by Resnik method have higher correlation coefficients with integrated LLS scores. Thus, in following experiments, we only used the SS scores calculated by the Resnik method.

The functional network generated using SS (referred to as *SG*) was compared to those generated by the method of Lee *et al.* [31]. Eight types of functional genomic, proteomic, and comparative genomic data sets were used to construct the gene network by Lee *et al.*. These eight types of data resulted in nine functional

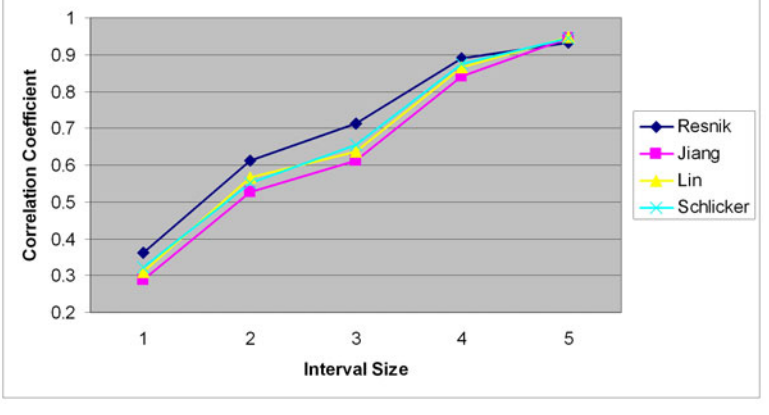



**Fig. 18.** Correlation between integrated log likelihood score (LLS) and GO semantic similarity (SS) computed using four different metrics (Resnik, Jiang and Conrath, Lin, and Schlicker). LLS was averaged over uniform SS intervals with various interval sizes. Correlation coefficients were then computed for the different interval sizes to compare LLS and SS.

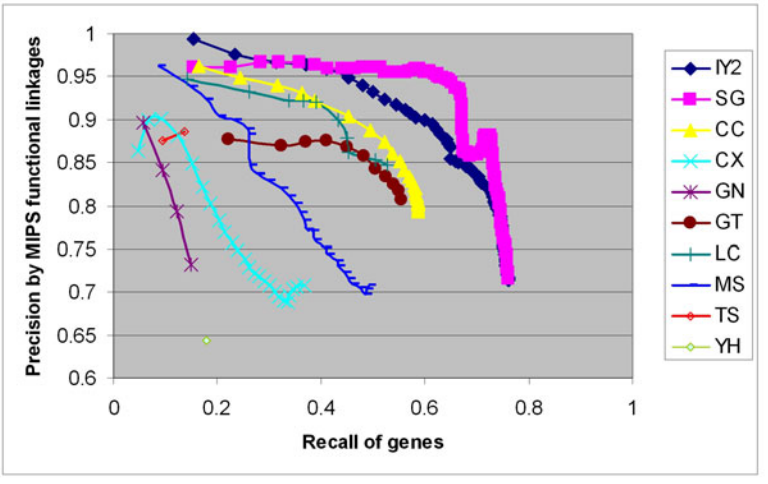



**Fig. 19.** Comparison of gene networks. Precision and recall of yeast genes are calculated using the unbiased MIPS functional linkage reference set as described by Lee *et al.* [31]. Precision is measured using reference linkages derived from MIPS functional annotation, masking the term "protein synthesis", and recall is calculated for total yeast genes. Gene pairs in each set were ranked by LLS or SS score, the cumulative precision and recall were calculated for each successive bin of 1,000 gene pairs.

gene networks: eight networks generated using single LLS score derived from each data type and one generated by using integrated LLS derived from all eight data types. The nine functional gene networks included those generated by using LLS scores derived from the following data sources: (1) Co-citation of literature data (the network was referred to as CC), (2) Co-expression of microarray data (the network was referred to as CX), (3) Gene neighborhoods of bacterial and archaeal orthologs (the network was referred to as GN), (4) Yeast genetic interactions (the network was referred to as GT), (5) Literature curated yeast protein interactions (the network was referred to as LC), (6) Protein complexes from affinity purification/mass spectrometry (the network was referred to as MS), (7) Protein interactions inferred from tertiary structures of complexes (the network was referred to as TS), (8) High-throughput yeast 2-hybrid assays (the network was referred to as YH), and (9) all data sets (the network was referred to as IY2). The networks generated by SS (the network was referred to as SG), along with the networks generated by various data sets described above, were compared using a training set derived from the MIPS protein function annotations by calculating the recall and precision of the MIPS reference linkage (Figure 19). The SG and IY2 networks showed high gene coverage and high precision, and surpassed that of any network constructed using the single individual data set in terms of precision at a given coverage. The SG network outperformed IY2 network at considerable gene coverage range (40% to 70% coverage) in terms of precision. This indicates that using only GO information to generate a functional gene network is a useful alternative to an approach that needs to integrate multiple data sets.

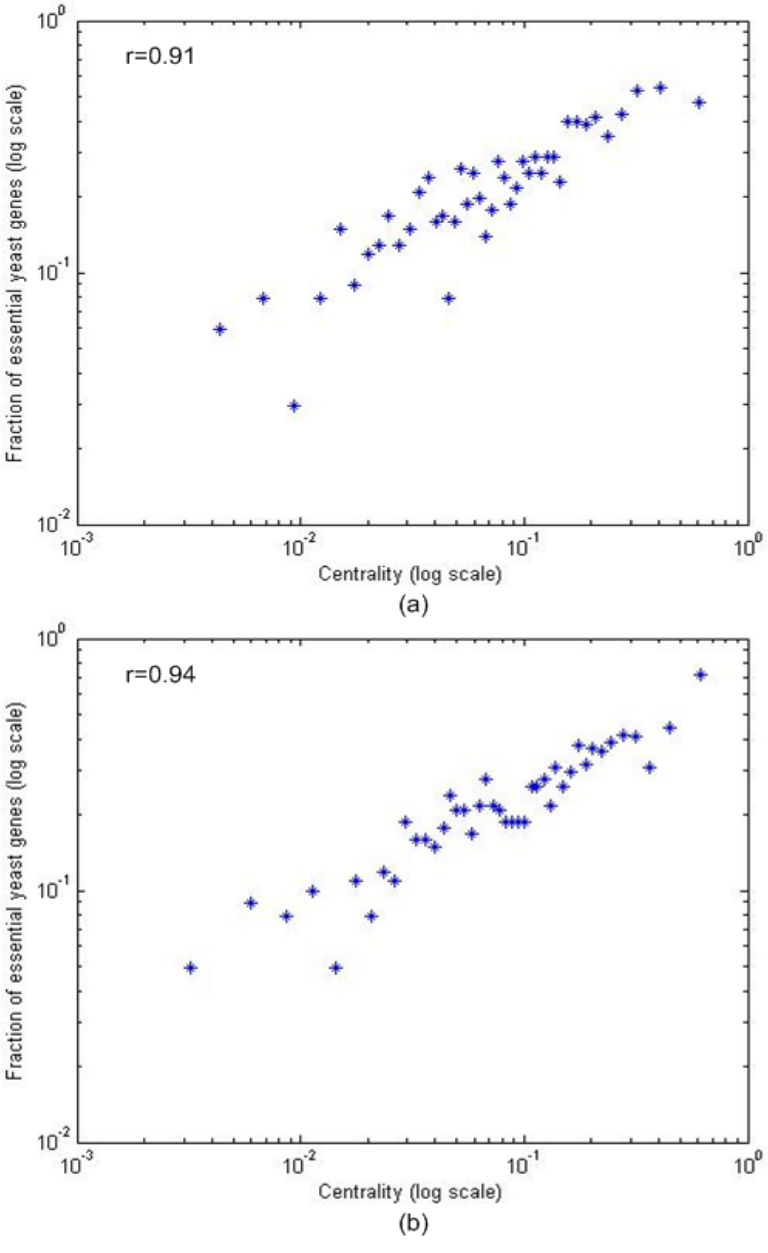



**Fig. 20.** Correlation between gene centrality and lethality. Each plot presents the correlation between centrality and the lethality of the genes for a given network. (a) shows the plot for the gene network derived from integrated LLS score, and (b) shows the plot for the gene network derived from SS score. The correlation coefficient ($r$) is measured as the nonparametric Spearman correlation coefficient.

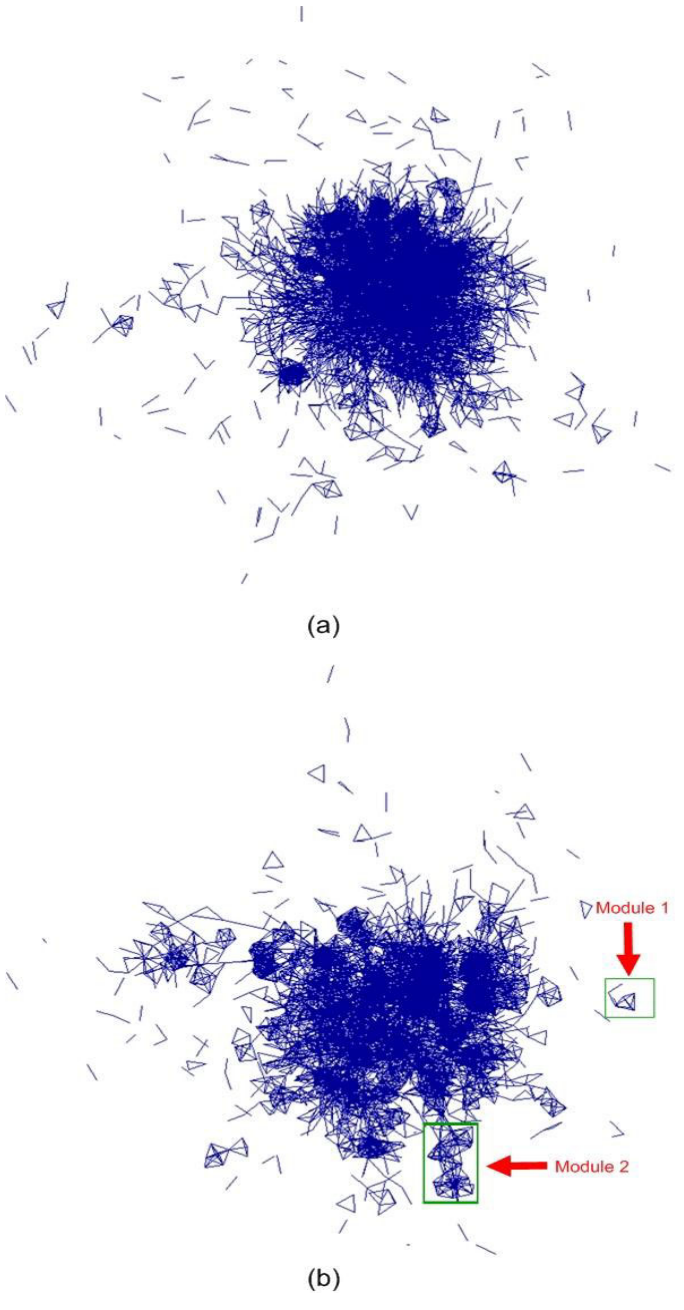



**Fig. 21.** A visualized comparison of the network derived from integrated LLS score (a) (plot of the top-scoring 10,000 linkages ranked by LLS score) with the network derived from SS score (b) (plot of the top-scoring 10,000 linkages ranked by SS score). Two functional modules are marked.

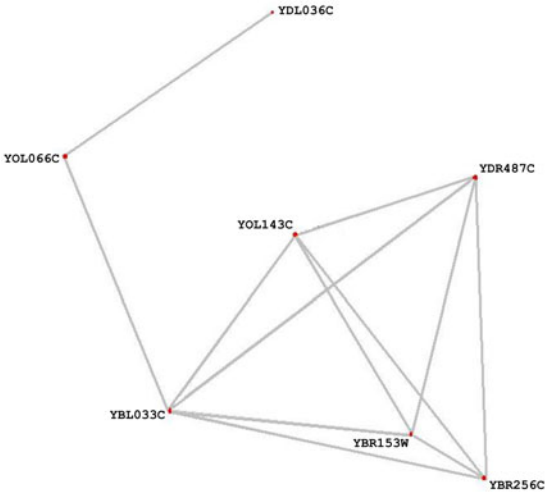



**Fig. 22.** A function module of genes involved in riboflavin metabolic process in yeast. Five of six genes involved in this pathway in yeast formed a clique.

**Comparison between Functional Networks.** The correlation analysis between network centrality and lethality of genes was performed. Network centrality is calculated as the number of linkages per gene normalized by the maximum observed value; and lethality of genes is calculated as the fraction of essential genes for each successive bin of 100 genes ranked by decreasing network centrality. Figure 20 shows that both SG and IY2 networks have a high correlation between lethality and centrality, indicating the high quality of these two functional networks. Although these two networks are roughly comparable in their overall quality and properties, correlation of the SS-drived network is slightly stronger than that derived from integrated LSS (0.94 and 0.91 respectively). Visualized networks also show substantially increased "clumping" of genes in the network derived from LSS (Figure 21).

**Case study: Functional Module Predicted from the SS Gene Network.** The resulting network of genes is highly complex (Figure 21). In order to discover and more conveniently describe the organization of the genes, we searched for coherent modules of genes in the network. These modules may be obtained using a unsupervised clustering algorithm [7]. The result is that genes are divided into groups (clusters or modules) according to the parameters chosen for clustering algorithm. The chosen parameters will affect the number of groups along with the group size. In our network (computed using SS), several gene groups were coherent enough to be visualized directly from the network topology (Figure 21). Here we present two examples of such easily identified functional modules.

One such module was identified as containing genes involved in yeast riboflavin metabolic pathway. Annotations showed that there were six genes associated with the GO term GO:0009231 (riboflavin metabolic process): *YBL033C*,

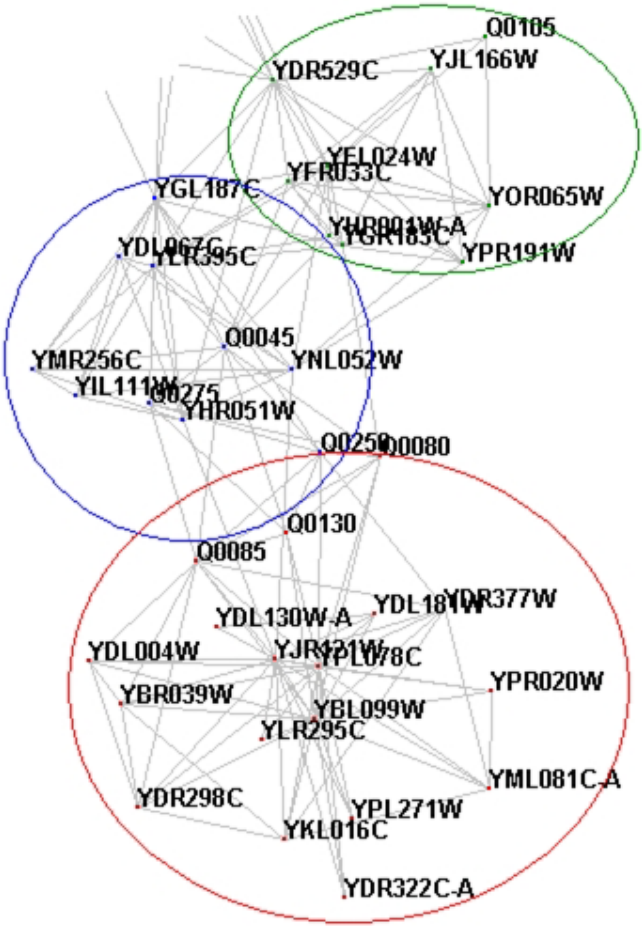



**Fig. 23.** Functional modules illustrate the hierarchy and relationship of GO terms

*YOL143C*, *YDR487C*, *YBR153W*, *YBR256C*, and *YOL066C*. These six genes were found in a coherent group (Figure 22; also see "Module 1" in Figure 21). Five of these genes formed a clique that was almost isolated from other genes in the network. The seventh gene from this module, YDL036C, shares the GO term "tRNA pseudoundine synthesis" with YOL066C and is thus included in the module. This indicates a strong correlation between genes specifically involved in riboflavin metabolism.

Another example is shown in Figure 23 (also marked as "Module 2" in Figure 21). This is a "super-module", and can be divided into three sub-modules, all of which are involved in ATP synthesis through oxidative phosphorylation. In oxidative phosphorylation, ATP is synthesized using energy from the passage of electrons through the electron-transport chain in the mitochondria. The ATP-generating reaction is catalyzed by the ATP synthase enzyme, which is

located on the inner mitochondrial membrane. The sub-module farthest away from the majority of genes in the network (red circle in Figure 23) contains genes encoding components of the ATP synthase complex. The central sub-module (blue circle in Figure 23) contains the genes encoding structural proteins comprising cytochrome C oxidase, which catalyzes the terminal step in the electron transport chain: the generation of water from $O_2$. The sub-module closest to the rest of the gene network (green circle in Figure 23) contains genes encoding the cytochrome bc1 complex which passes electrons from ubiquinol to cytochrome C. Although more systematic clustering analysis needs to be performed to further explore the relationship between these three sub-modules and other genes involved in oxidative phosphorylation in our network, this rudimentary analysis lends credibility to the validity of our network.

### 4.6 Discussion

The SS score had a broad correlation with the LLS score. However, the SS score also had many distinctive features. The network predicted by the SS scores outperformed those generated by the integrated LLS scores or the individual LLS scores in the following aspects: automatic removal of a functional bias in network training reference sets improved precision and recall across the network, and resulted in higher correlation between a gene's lethality and centrality in the network. We also showed that the resulting network can be applied to generate coherent function modules and their associations. We conclude that determination of semantic similarity between genes based on GO information can be used to generate a functional network of yeast genes that is comparable or better than those that are directly based on integrated heterogeneous genomic and proteomic data.

Both the SS and LLS score can be used to generate whole genome functional gene networks; however, the SS score has obvious advantages over the LLS score. First, calculating LLS scores requires a large number of data points from as many sources as possible. Multi-source data is not easy to obtain and may contain many missing values. This diminishes the merit of multi-source data. Second, calculating LLS is a supervised learning approach and requires reference training sets. The quality of such training sets affects the supervised learning process adversely. Therefore, reference training sets must be carefully chosen and preprocessed (for example, by removing systematic functional bias) to optimize results. Reference training sets were employed by Lee *et al.* in their computation, but using only a subset of the GO annotations (performance was poor when the entire annotation system was used). Third, integrating LLS from multi-source data requires the estimation of the correlation between different data types. Although some algorithms have been proposed, this still remains a challenge. In contrast, using the SS score method circumvents almost all of the above problems. SS score is calculated using only one data source – the GO annotation hierarchy. GO annotations have already integrated many sources of data and are hierarchically structured. SS scores can be applied to sets of annotated genes in a genome and do not have the adverse effect of training set bias. For example,

Lee *et al.* pointed out that the frequency distribution of annotated yeast genes in the GO "biological process" subcategory is heavily biased toward the single term "protein biosynthesis" (GO:0006412) [31]. This term alone is responsible for 27% of the total reference genes. This dominant term was "masked" in their research in order to remove what appeared to be a systematic bias. However, since semantic similarity score is calculated based on information content of a term (*i.e.*, the higher the frequency of a term, the lower its information content), such terms with high frequency have little impact on the scoring system.

One disadvantage of calculating the SS score between genes using GO information is that many unannotated genes will be excluded, and thus their potential gene linkages will not be included in the network. This problem can be solved by first predicting the functional annotations for unannotated genes (as outlined in Section 3), and then applying our approach to generate the network. Dealing with unannotated genes in the network is also a problem for the LLS approach since unannnotated genes may have little or no experimental support. Another issue is that LLS calculations require a functional annotation reference set to benchmark the assigned linkage between genes. Functional linkages involving unannotated genes will not be benchmarked. In that sense, LLS approach shares the same disadvantage as the SS approach. However one big advantage of LLS is that it is more flexible and can integrate most newly available data sets.

## 5 Conclusions

In this chapter, we have mainly focused on introducing a semantic metric using GO information, and the application of such a metric in function prediction for uncharacterized proteins and functional gene network generation. Although the fact that proteins having similar function annotations are more likely to have interactions has been confirmed in the literature, we provide a quantitative measure to estimate the similarity, and to uncover the relationship between the metric and the support of PPI data. In the application of generating functional gene network, we conclude that the semantic score between genes using GO information is able to generate a comparable or improved functional network of yeast genes as compared to those that have integrated heterogeneous genomic and proteomic data directly. Experimental results show that predicting linkage between genes by calculating pairwise semantic score using an information theory-based approach can reduce the functional bias in a reference training set, and thus improve the network quality without information loss.

In both applications, although only gene ontology information was used, data from multiple sources were involved indirectly, considering that GO annotations have been generated after having integrated information from multiple data sources. GO annotations can be viewed as a way in which unstructured multiple data sources are integrated into a single structured data source. Further investigations and comparisons are needed to reveal the relationship between widely used semantic metric (as used in this chapter) and other new proposed semantic scores.

## References


1. Ashburner, M., Ball, C., Blake, J., Botstein, D., Butler, H., Cherry, J., Davis, A., Dolinski, K., Dwight, S., Eppig, J., Harris, M., Hill, D., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J., Richardson, J., Ringwald, M., Rubin, G., Sherlock, G.: Gene ontology: tool for the unification of biology. the gene ontology consortium. Nat. Genet. 25(1), 25–29 (2000)
2. Resnik, P.: Using information content to evaluate semantic similarity. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence, pp. 448–453 (1995)
3. Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. In: Proceedings of International Conference on Research in Computational Linguistics (1997)
4. Lin, D.: An information-theoretic definition of similarity. In: Proceedings of the 15th International Conference on Machine Learning (1998)
5. Schlicker, A., Domingues, F.S., Rahnenfuhrer, J., Lengauer, T.: A new measure for functional similarity of gene products based on gene ontology. BMC Bioinformatics 7, 302–317 (2006)
6. Lord, P.W., Stevens, R.D., Brass, A., Goble, C.A.: Semantic similarity measures as tools for exploring the gene ontology. In: Pac. Symp. Biocomput., pp. 601–612 (2003)
7. Sharan, R., Ulitsky, I., Shamir, R.: Network-based prediction of protein function. Molecular Systems Biology 3(88), 1–13 (2007)
8. Schwikowski, B., Uetz, P., Fields, S.: A network of protein-protein interactions in yeast. Nat. Biotechnol. 18, 1257–1261 (2000)
9. Hishigaki, H., Nakai, K., Ono, T., Tanigami, A., Takagi, T.: Assessment of prediction accuracy of protein function from protein-protein interaction data. Yeast 18, 523–531 (2001)
10. Chua, H.N., Sung, W.K., Wong, L.: Exploiting indirect neighbours and topological weight to predict protein function from proteinprotein interactions. Bioinformatics 22, 1623–1630 (2006)
11. Letovsky, S., Kasif, S.: Predicting protein function from protein/protein interaction data: a probabilistic approach. Bioinformatics 204(suppl. 1), i197–i204 (2003)
12. Deng, M., Tu, Z., Sun, F., Chen, T.: Mapping gene ontology to proteins based on protein–protein interaction data. Bioinformatics 20(6), 895–902 (2004)
13. Vazquez, A., Flammini, A., Maritan, A., Vespignani, A.: Global protein function prediction from protein-protein interaction networks. Nat. Biotechnol. 21, 697–700 (2003)
14. Karaoz, U., Murali, T.M., Letovsky, S., Zheng, Y., Ding, C., Cantor, C.R., Kasif, S.: Whole-genome annotation by using evidence integration in functional-linkage networks. Proc. Natl. Acad. Sci. USA 101, 2888–2893 (2004)
15. Nabieva, E., Jim, K., Agarwal, A., Chazelle, B., Singh, M.: Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. Bioinformatics 21(suppl. 1), i302–i310 (2005)
16. Joshi, T., Chen, Y., Becker, J.M., Alexandrov, N., Xu, D.: Genome-scale gene function prediction using multiple sources of high-throughput data in yeast saccharomyces cerevisiae. OMICS 8(4), 322–333 (2004)
17. Lee, H., Tu, Z., Deng, M., Sun, F., Chen, T.: Diffusion kernel-based logistic regression models for protein function prediction. OMICS 10(1), 40–55 (2006)

18. Lanckriet, G.R., De Bie, T., Cristianini, N., Jordan, M.I., Noble, W.S.: A statistical framework for genomic data fusion. Bioinformatics 20(16), 2626–2635 (2004)
19. Tsuda, K., Shin, H., Schölkopf, B.: Fast protein classification with multiple networks. Bioinformatics 21(suppl. 2) (2005)
20. Bader, G.D., Hogue, C.W.: An automated method for finding molecular complexes in large protein interaction networks. BMC Bioinformatics 4(1) (2003)
21. Sharan, R., Ideker, T., Kelley, B., Shamir, R., Karp, R.M.: Identification of protein complexes by comparative analysis of yeast and bacterial protein interaction data. J. Comput. Biol. 12(6), 835–846 (2005)
22. Arnau, V., Mars, S., Marin, I.: Iterative cluster analysis of protein interaction data. Bioinformatics 21(3), 364–378 (2005)
23. Segal, E., Wang, H., Koller, D.: Discovering molecular pathways from protein interaction and gene expression data. Bioinformatics 19(suppl. 1), i264–i271 (2003)
24. Kelley, R., Ideker, T.: Systematic interpretation of genetic interactions using protein networks. Nature Biotechnology 23(5), 561–566 (2005)
25. Wu, Y., Lonardi, S.: A linear-time algorithm for predicting functional annotations from proteinprotein interaction networks. In: Proceedings of the Workshop on Data Mining in Bioinformatics (BIOKDD 2007), pp. 35–41 (2007)
26. Jansen, R., Yu, H., Greenbaum, D., Kluger, Y., Krogan, N.J., Chung, S., Emili, A., Snyder, M., Greenblatt, J.F., Gerstein, M.: A Bayesian networks approach for predicting protein-protein interactions from genomic data. Science 302, 449–453 (2003)
27. Zhang, L.V., Wong, S.L., King, O.D., Roth, F.P.: Predicting co-complexed protein pairs using genomic and proteomic data integration. BMC Bioinformatics 5 (April 2004)
28. Ben-Hur, A., Noble, W.S.: Kernel methods for predicting protein-protein interactions. Bioinformatics 21(suppl. 1) (June 2005)
29. Qi, Y., Bar-Joseph, Z., Klein-Seetharaman, J.: Evaluation of different biological data and computational classification methods for use in protein interaction prediction. PROTEINS: Structure, Function, and Bioinformatics 3, 490–500 (2006)
30. Lee, I., Date, S.V., Adai, A.T., Marcotte, E.M.: A probabilistic functional network of yeast genes. Science 306, 1555–1558 (2004)
31. Lee, I., Li, Z., Marcotte, E.M.: An improved, bias-reduced probabilistic functional gene network of baker's yeast, saccharomyces cerevisiae. PLoS ONE, e988 (2007)
32. von Mering, C., Krause, R., Snel, B., Cornell, M., Oliver, S.G., Fields, S., Bork, P.: Comparative assessment of large-scale data sets of protein-protein interactions. Nature 417, 399–403 (2002)
33. Yu, J., Fotouhi, F.: Computational approaches for predicting protein-protein interactions: A survey. J. Med. Syst. 30(1), 39–44 (2006)
34. Bader, J.S.: Greedily building protein networks with confidence. Bioinformatics 19(15), 1869–1874 (2003)
35. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M., Sherlock, G.: Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat. Genet. 25, 25–29 (2000)
36. Stark, C., Breitkreutz, B.J., Reguly, T., Boucher, L., Breitkreutz, A., Tyers, M.: BioGRID: a general repository for interaction datasets. Nucleic Acids Res. 34 (January 2006)

37. Mewes, H., Gruber, F., Geier, C., Haase, B., Kaps, D., Lemcke, A., Mannhaupt, K., Pfeiffer, G., Schuller, F.: MIPS: a database for genomes and protein sequences. Nucleic Acids Res. 30(1), 31–34 (2002)
38. Murali, T., Wu, C., Kasif, S.: The art of gene function prediction. Nat. Biotechnol. 24(12), 1474–1475 (2006)
39. Giaever, G., Chu, A., Ni, L., Connelly, C., Riles, L., et al.: Functional profiling of the saccharomyces cerevisiae genome. Nature 418(6896), 387–391 (2002)
40. Myers, C.L., Robson, D., Wible, A., Hibbs, M.A., Chiriac, C., Theesfeld, C.L., Dolinski, K., Troyanskaya, O.G.: Discovery of biological networks from diverse functional genomic data. Genome Biology 6, R114 (2005)
41. Rhodes, D.R., Tomlins, S.A., Varambally, S., Mahavisno, V., Barrette, T., Kalyana-Sundaram, S., Ghosh, D., Pandey, A., Chinnaiyan, A.M.: Probabilistic model of the human protein-protein interaction network. Nature Biotechnology 23(8), 951–959 (2005)
42. Pan, X., Ye, P., Yuan, D.S., Wang, X., Bader, J.S., Boeke, J.D.: A DNA integrity network in the yeast Saccharomyces cerevisiae. Cell 124, 1069–1081 (2006)
43. Zhong, W., Sternberg, P.W.: Genome-wide prediction of c. elegans genetic interactions. Science 311, 1481–1484 (2006)
44. Huang, H., Zhang, L.V., Roth, F.P., Bader, J.S.: Probabilistic paths for protein complex inference, pp. 14–28 (2006)
45. Jeong, H., Mason, S.P., Barabasi, A.L., Oltvai, Z.N.: Lethality and centrality in protein networks. Nature 411, 41–42 (2001)
46. Sevilla, J.L., Segura, V., Podhorski, A., Guruceaga, E., Mato, J.M., Martinez-Cruz, L.A., Corrales, F.J., Rubio, A.: Correlation between gene expression and go semantic similarity. IEEE/ACM Trans. Comput. Biol. Bioinformatics 2(4), 330–338 (2005)

# Chapter 8
# Mining Multiple Biological Data for Reconstructing Signal Transduction Networks

Thanh-Phuong Nguyen and Tu-Bao Ho

[1] The Microsoft Research - University of Trento Centre for Computational and Systems Biology, Piazza Manci 17, 38123, Povo, Trento, Italy
nguyen@cosbi.eu
[2] School of Knowledge Science, Japan Advanced Institute of Science and Technology, 1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan
bao@jaist.ac.jp

**Abstract.** Signaling transduction networks (STNs) are the key means by which a cell converts an external signal (e.g. stimulus) into an appropriate cellular response (e.g. cellular rhythms of animals and plants). The essence of STN is underlain in some signaling features scattered in various data sources and biological components overlapping among STN. The integration of those signaling features presents a challenge. Most of previous works based on PPIs for STN did not take the signaling properties of signaling molecules and components overlapping among STN into account. This paper describes an effective computational method that can exploit three biological facts of STN applied to human: protein-protein interaction networks, signaling features and sharing components. To this end, we introduce a soft-clustering method for doing the task by exploiting integrated multiple data, especially signaling features, i.e., protein-protein interactions, signaling domains, domain-domain interactions, and protein functions. The gained results demonstrated that the method was promising to discover new STN and solve other related problems in computational and systems biology from large-scale protein interaction networks. Other interesting results of the early work on yeast STN are additionally presented to show the advantages of using signaling domain-domain interactions.

## 1 Introduction

The way how an organism can survive is the continually adjusting its internal state to changes in the environment. To track environmental changes, the organism must communicate effectively with their surroundings. These may be in the form of chemicals, such as hormones or nutrients, or may take another form, such as light, heat, or sound. A signal itself rarely causes a simple, direct chemical change inside the cell. Instead, the signal is transduced through a multi-step chain, or changed in form. Signal transduction systems are especially important in multicellular organisms, because of the need to coordinate the activities of hundreds to trillions of cells [1]. Signal transduction network refers to the entire set of pathways and interactions by which environmental signals are received and responded to by single cells [53]. It is unsurprising that many components

D.E. Holmes, L.C. Jain (Eds.): Data Mining: Found. & Intell. Paradigms, ISRL 25, pp. 163–185.
springerlink.com 

of these signal transduction circuits are oncogenes or tumour suppressors, emphasizing the importance of understanding signaling in normal tissues and targeting aberrant signaling in diseases [44].

Traditionally, the discovery of molecular components of signaling networks in yeast and mammals has relied upon the use of gene knockouts and epistasis analysis. Although these methods have been highly effective in generating detailed descriptions of specific linear signaling pathways, our knowledge of complex signaling networks and their interactions remains incomplete. New computational methods that capture molecular details from high-throughput genomic data in an automated fashion are desirable and can help direct the established techniques of molecular biology and genetics [6, 49].

Signal transduction networks (STN) are chiefly based on interactions between proteins, which are intrinsic to almost all cellular functions and biological processes [12, 15, 22, 23]. The study of protein interactions is fundamental to understanding the complex mechanisms underlying signal transduction networks. Protein-protein interaction problem have attracted a lot of research for last ten years, both experimental methods [10, 28, 54, 59] and computational method as well [11, 13, 17, 29, 38, 43, 47]. In addition, the enormous amount of protein-protein interaction (PPI) data has been nowadays generated and published much more than ever [37, 41] such as DIP [51], MISP [45], i2d [14], MINP [16], BIND [3], STRING [39], etc. Hence, the PPI-based approach is greatly appealing for studying STN.

When exploring PPI data to reconstructing STN, we face two problems of complexity. The first one derives from the large number of molecules and multiple types of interactions between them. In addition to the size of the signaling machinery, a second layer of complexity inter-connectivity of signaling biochemistry is apparent from the fact that signaling proteins often contain multiple functional domains, thus enabling each to interact with numerous downstream targets [18]. Therefore, it has become emerging to develop effective data mining methodologies to extract, process, integrate and discover useful knowledge from the PPI network data accompanying with other proteomic and genomic data. These methodologies should be robust to manipulate the huge number of proteins involving in the STN and also flexible to combine other signaling features. The retrieved knowledge is expected to better understand the system behavior of signaling networks, and to predict higher order functions that can be validated by experiments.

The objective of this paper is to present the study of STN based on PPI. First, we briefly introduce some background of STN and PPI network. Then the work on combining multiple data to reconstruct human STN is described. In addition, we provide some results of Yeast STN reconstruction by using signaling features. The future work and the summarization are lastly given.

## 2 Background

In this section, we present some biological background of the signal transduction network and then the protein-protein interaction network.

### 2.1 Signal Transduction Network

Signal transduction networks are the key means for the communication between cells. Those networks consist of extracellular and intracellular signaling molecules. Some of

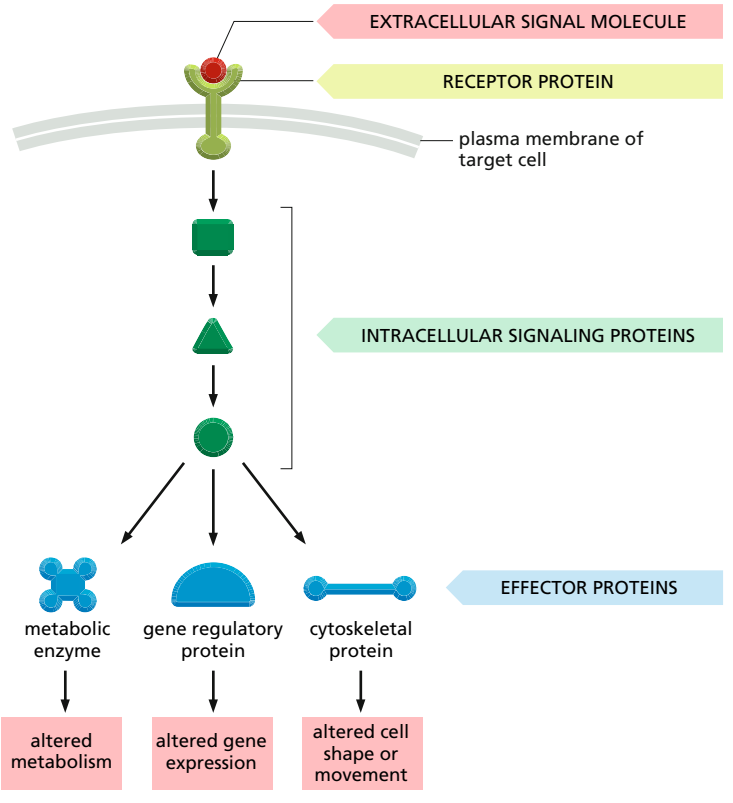



**Fig. 1.** A simple intracellular signaling pathway activated by an extracellular signal molecule. The signal molecule usually binds to a receptor protein that is embedded in the plasma membrane of the target cell and activates one or more intracellular signaling pathways mediated by a series of signaling proteins. Finally, one or more of the intracellular signaling proteins alters the activity of effector proteins and thereby the behavior of the cell. Adopted in Chapter 15: Mechanisms of Cell Communication in the book Molecular Biology of the Cell (textbook) [1].

extracellular ones operate over long distances, signaling to cells far away; others signal only to immediate neighbors. Most cells in multicellular organisms both emit and receive signals. Once receptor proteins bind the signal molecules, one or more intracellular signaling pathways are activated. These relay chains of molecules mainly intracellular signaling proteins process the signal inside the receiving cell and distribute it to the appropriate intracellular targets. These targets are generally effector proteins, which are altered when the signaling pathway is activated and implement the appropriate change of cell behavior. Depending on the signal and the nature and state of the receiving cell, these effectors can be gene regulatory proteins, ion channels, components of a metabolic pathway, or parts of the cytoskeleton among other things [1]. An simple example of an intracellular signaling pathway is depicted in Figure 1. Because of the wide-range of function, signal transduction networks play a pivotal role in almost of

fundamental cellular processes including cell proliferation, metabolism, differentiation, and survival [37].

Additionally, an intracellular signaling cascade can no longer be viewed as a linear pathway that relays and amplifies information. It is known that the cell uses these pathways as a way of integrating multiple inputs to shape a uniquely defined output. Hence, the interactions of different pathways and the dynamic modulation of the activities of the components within signaling pathways can create a multitude of biological outputs. The cell appears to use these complex networks of interacting pathways and regulatory feedback mechanisms to co-coordinately regulate multiple functions. These outputs allow the cell to respond to and adapt to an ever-changing environment [40].

## 2.2 Protein-Protein Interaction

Protein-protein interactions are specific interactions between two or more proteins. Indeed, protein-protein interactions are at the core of the entire interactomics system of any living cell.

### 2.2.1 Biological Characteristics of the Protein-Protein Interactions

The followings are the summary of general characteristics of protein-protein interactions [60].

*Classification:* Protein-protein interactions can be arbitrarily classified based on the proteins involved (structural or functional groups) or based on their physical properties (weak and transient, non-obligate vs. strong and permanent). Protein interactions are usually mediated by defined domains, hence interactions can also be classified based on the underlying domains.

*Universality:* Protein-protein interactions affect almost all processes in a cell: structural proteins need to interact in order to shape organelles and the whole cell, molecular machines such as ribosomes or RNA polymerases are hold together by protein-protein interactions, and the same is true for multi-subunit channels or receptors in membranes [2].

*Specificity:* Distinguishes such interactions from random collisions that happen by Brownian motion in the aqeous solutions inside and outside of cells. Note that many proteins are known to interact although it remains unclear whether certain interactions have any physiological relevance.

*Number of interactions:* It is estimated that even simple single-celled organisms, such as yeast have their roughly 6000 proteins interact by at least 3 interactions per protein, i.e. a total of 20,000 interactions or more. By extrapolation, there may be 100,000 interactions in the human body.

*Protein-protein interactions and protein complexes:* Most protein-protein interactions are detected as interacting pairs or as components of protein complexes. Such complexes may contain dozens or even hundreds of protein subunits (ribosomes, spliceosomes, etc.). It has even been proposed that all proteins in a given cell are connected in a huge network in which certain protein interactions are forming and dissociating constantly.

#### 2.2.2 Topological Characteristics of the Protein-Protein Interaction Network

The followings are some topological characteristics of the protein-protein interaction networks [36].

*Scale-free network:* Protein-protein interactions have the features of a scale-free network, meaning that their degree distribution approximates a power law, $P(k) \sim k^{\gamma}$. In scale-free networks, most proteins participate in only a few interactions, while a few (termed "hubs") participate in dozens of interactions.

*Small-world effect:* Protein-protein interaction networks have a characteristic property known as the "small world effect", which states that any two nodes can be connected via a short path of a few links. Although the small-world effect is a property of random networks, the path length in scale-free networks is much shorter than that predicted by the small-world effect. Therefore, scale-free networks are "ultra-small". This short path length indicates that local perturbations in metabolite concentrations could permeate an entire network very quickly.

*Disassortativity:* In protein-protein interaction networks, highly-connected nodes (hubs) seldom directly link to each other. This differs from the assortative nature of social networks, in which well-connected people tend to have direct connections to each other. By contrast, all biological and technological networks have the property of disassortativity, in which highly-connected nodes are infrequently linked each other.

## 3 Constructing Signal Transduction Networks Using Multiple Data

In this section, we present our proposed method to construct the STN from the human PPI networks and multiple databases using soft-clustering. The related work were summarized in Section 3.1. In the next section (Section 3.2), we described our framework for doing the task of constructing STN. The evaluation of the experiments is showed in Section 3.4. Experimental results and some discussions are presented in Section 3.4.2.

### 3.1 Related Work

Constructing STN based on PPI is an area of much ongoing research. A statistical model, based on representing proteins as collections of domains or motifs, which predicts unknown molecular interactions within these biological networks, was proposed by Gomez *et al.* [25]. Using Markov chain Monte Carlo method, they then modeled the signal transduction networks (STN) in terms of domains in upstream and downstream protein interactions. Steffen *et al.* developed a computational method for generating static models of STN which utilizes PPI maps generated from large-scale two-hybrid screens and expression profiles from DNA microarrays [57]. Liu *et al.* applied a score function that integrated protein-protein interaction data and microarray gene expression data to predict the order of signaling pathway components [37]. Concerning protein modification time-course data, Allen *et al.* applied a method of computational algebra to modeling of signaling networks [4]. Another work by Fukuda *et al.* is to represent the model of signal transduction pathways based on a compound graph structure. Their

method is designed to capture directly the structure of pathways that biologists bear in mind or that are described in the articles [20]. One of the most recent work is to search for the optimal subnetworks from PPI according to some cost functions [62]. Korcsm *et al.* have presented a signaling resource, SignaLink, compiled by applying uniform manual curation rules and data structures across eight major, biochemically defined signaling pathways in three metazoans. The curation method allowed a systematic comparison of pathway sizes and cross-talks [32]. Other work on the cross-talk of signaling pathway used network theory to find out the pathway interactions through connector proteins as the key means to transduce signals between pathways [27]. Li *et al.* built the global pathway crosstalk network by combining pathway and protein interaction data based on the shortest path profiles only [35].

Although the previous work achieved many results, there are still some biological characteristics of STN that they did not take much into account. First, it is known that the deeper level underlying the PPI to transmit signals in cells are functional domains, so-called signaling domains, and their interactions [46], [18]. Data of those significant signaling features are structured, complexly relational, and sparse in different data sources. In order to construct STN effectively, those data is needed to be appropriately integrated. Second, STN indeed have many overlapping components including proteins and their interactions [40]. This work aims to solve those two intricate problems of STN to better construct STN from PPI networks. To this end, we introduce an effective computational method to construct STN that (1) exploits integrated multiple signaling features of STN from heterogeneous sources, i.e., protein-protein interactions, signaling domains, domain-domain interactions, and protein functions, and (2) detects overlapping components using soft-clustering. Additionally, in previous work clustered objects were often individual proteins, but our method handled clustered objects as the functional or physical protein interactions because these interactions are the means to transmit signals in cells.

We evaluated the proposed method using human protein interaction network published in the database Reactome. Five complex biological processes were tested to demonstrate the performance. The clustered results are well-matched with these five processes. To the best of our knowledge, this work is the first one that computationally solves the STN problem for *Homo Sapiens*. The preliminary results open a prospect to study other problems related to complex biological systems in *Homo Sapiens*.

### 3.2 Materials and Methods

The method does two main tasks. The first one is to extract and preprocess signaling feature data from various data sources. Those relational data in heterogeneous types are then weighted and normalized by the proposed functions. Based on data extracted in the first task, the second one is to combine weighted data and then cluster protein-protein interactions into STN using soft-clustering. Because the main data mining technique in this paper is the clustering, we first review the clustering problem in the PPI network analysis to provide more details of this study. The next two subsections, Subsection 3.3.4 and Subsection 3.3.5, describe two mentioned tasks of the data extraction and the STN reconstruction, respectively.

### 3.3 Clustering and Protein-Protein Interaction Networks

A cluster is a set of objects which share some common characteristics. Clustering is the process of grouping data objects into sets (clusters) which demonstrate greater similarity among objects in the same cluster than the ones in the different clusters. Clustering differs from classification; in the latter, objects are assigned to predefined classes, while clustering defines the classes themselves. Thus, clustering is an unsupervised classification problem, which means that it does not rely on training the data objects in predefined classes.

Clustering methods can be broadly divided into hierarchical and partitioning ones. In partitioning clustering, there are two categories of hard-clustering method and soft-clustering method. On the one hand, hard-clustering is based on classical set theory and assigns an instance to exactly one cluster, e.g., k-means, SOMs, etc. On the other hand, soft-clustering assigns an instance to several cluster and differentiate grade of representation (cluster membership), e.g., fuzzy c-means, HMMs, etc. [21].

In the traditional clustering approaches, a simple distance measure can often be used to reflect dissimilarity between two patterns, while other similarity measures can be used to characterize the conceptual similarity between patterns. However, most of protein-protein interactions are the binary ones without direction, the graph of PPI network is represented with proteins as nodes and interactions as edges. The relationship between two proteins is therefore a simple binary value: 1 if they interact, 0 if they do not. Because of this monotony, the definition of the distance between the two proteins becomes more difficult. Moreover, the reliable clustering of PPI networks is problematical due to a high rate of false positives and the huge volume of data.

Clustering approaches for PPI networks can be broadly classified into two categories, distance-based and graph-based. Distance-based clustering uses classic clustering techniques and focuses on the definition of the distance between proteins. Graph-based clustering includes mainly take into account the topology of the PPI network. Based on the structure of the network, the density of each subgraph is maximized or the cost of cut-off minimized while separating the graph. The following subsections will discuss each of these clustering approaches in more detail. In addition, we also give the presentation of soft clustering for PPI networks.

#### 3.3.1 Distance-Based Clustering

Generally, there are four distance-based clustering approaches applying for PPI networks [36]. The first category of approaches uses classic distance measurement methods, which offered a variety of coefficient formulas to compute the distance between proteins in PPI networks [50]. The second class of approaches defines a distance measure based on network distance, including the shortest path length, combined strength of paths of various lengths, and the average number of steps a Brownian particle takes to move from one vertex to another. The third approach type, exemplified by UVCLUSTER, defines a primary and a secondary distance to establish the strength of the connection between two elements in relationship to all the elements in the analyzed dataset [5]. The fourth is a similarity learning approach by incorporating some annotation data. Although these four categories of approaches each involve different methods for distance measurement, they all apply classic clustering approaches to the computed distance between proteins [36].

### 3.3.2 Graph-Based Clustering

A protein-protein interaction network is an undirected graph in which the weight of each edge between any two proteins is either 1 or 0. This section will explore graph-based clustering, another class of approaches to the process of clustering. Graph-based clustering techniques are explicitly presented in term of a graph, thus converting the process of clustering a data set into such graph-theoretical problems as finding a minimum cut or maximal subgraphs in the graph $G$ [36].

**a. Finding Dense Subgraphs.** The goal of this class of approaches is to identify the densest subgraphs within a graph; specific methods vary in the means used to assess the density of the subgraphs.

- ***Enumeration of Complete subgraphs.*** This approach is to identify all fully connected subgraphs (cliques) by complete enumeration. While this approach is simple, it has several drawbacks. The basic assumption underlying the method - that cliques must be fully internally connected - does not accurately reflect the real structure of protein complexes and modules. Dense subgraphs are not necessarily fully connected. In addition, many interactions in the protein network may fail to be detected experimentally, thus leaving no trace in the form of edges [55].
- ***Monte Carlo optimization.*** The use of a Monte Carlo approach allows smaller pieces of the cluster to be separately identified rather focusing exclusively on the whole cluster. Monte Carlo simulations are therefore well suited to recognizing highly dispersed cliques [55].
- ***Redundancies in PPI Network.*** This approach assumes that two proteins that share a significantly larger number of common neighbors than would arise randomly will have close functional associations. This method first ranks the statistical significance of forming shared partnerships for all protein pairs in the PPI network and then combines the pair of proteins with least significance. The p-value is used to rank the statistical significance of the relationship between two proteins. In the next step, the two proteins with smallest p-value are combined and are considered to be in the same cluster. This process is repeated until a threshold is reached [52].
- ***Molecular Complex Detection.*** Molecular complex detection (MCODE), proposed by Bader and Hogue, is an effective approach for detecting densely-connected regions in large protein-protein interaction networks. This method weights a vertex by local neighborhood density, chooses a few seeds with high weight, and isolates the dense regions according to given parameters. The MCODE algorithm operates in three steps: vertex weighting, complex prediction, and optional postprocessing to filter or add proteins to the resulting complexes according to certain connectivity criteria [8].

**b. Finding Minimum Cut.** Second category of graph-based clustering approaches generates clusters by trimming or cutting a series of edges to divide the graph into several unconnected subgraphs. Any edge which is removed should be the least important (minimum) in the graph, thus minimizing the informational cost of removing the edges. Here, the least important is based on the structure of the graph. It does not mean the interaction between these two proteins is not important.

- ***Highly Connected Subgraph (HCS) Algorithm.*** The highly-connected subgraph or HCS method is a graph-theoretic algorithm which separates a graph into several subgraphs using minimum cuts. The resulting subgraphs satisfy a specified density threshold. Despite its interest in density, this method differs from approaches discussed earlier which seek to identify the densest subgraphs. Rather, it exploits the inherent connectivity of the graph and cuts the most unimportant edges to find highly-connected subgraphs [26].
- ***Restricted Neighborhood Search Clustering Algorithm (RNSC).*** A cost-based local search algorithm based on the tabu search meta-heuristic was proposed [31]. In the algorithm, a clustering of a graph $G = (V,E)$ is defined as a partitioning of the node set $V$. The process begins with an initial random or user-input clustering and defines a cost function. Nodes are then randomly added to or removed from clusters to find a partition with minimum cost. The cost function is based on the number of invalid connections. An invalid connection incident with $v$ is a connection that exists between $v$ and a node in a different cluster, or, alternatively, a connection that does not exist between $v$ and a node $u$ in the same cluster as $v$.
- ***Super Paramagnetic Clustering (SPC).*** The super-paramagnetic clustering (SPC) method uses an analogy to the physical prop- erties of an inhomogenous ferromagnetic model to find tightly-connected clusters in a large graph [24].
- ***Markov Clustering.*** The Markov clustering (MCL) algorithm was designed specifically for application to simple and weighted graphs and was initially used in the field of computational graph clustering. The MCL algorithm finds cluster structures in graphs by a mathematical bootstrapping procedure. The MCL algorithm simulates random walks within a graph by the alternation of expansion and inflation operations [61].
- ***Line Graph Generation.***Line graph generation method generates the graph in which edges now represent proteins and nodes represent interactions [48]. First, the protein interaction network is transformed into a weighted network, where the weights attributed to each interaction reflect the degree of confidence attributed to that interaction. Next, the network connected by interactions is expressed as a network of interactions, which is known in graph theory as a line graph. These nodes are then linked by shared protein content. The scores for the original constituent interactions are then averaged and assigned to each edge. Finally, an algorithm for clustering by graph flow simulation, TribeMCL, is used to cluster the interaction network and then to reconvert the identified clusters from an interaction-interaction graph back to a protein-protein graph for subsequent validation and analysis.

### 3.3.3 Soft-Clustering for PPI Networks

Many proteins are believed to exhibit multiple functionalities in several biological process in general and STNs in particular. They do not participate in one process but intend to involve in some of them to perform different roles. Because soft clustering method is able to distinguish overlapping parts among clusters, it is potentially more sensible to reconstruct the biological processes. Some soft clustering methods are well applied to PPI networks.

The line graph generation is one of soft clustering techniques and has a number of attractive features [36]. It does not sacrifice informational content, because the original bidirectional network can be recovered at the end of the process. Furthermore, it takes into account the higher-order local neighborhood of interactions. Additionally, the graph it generates is more highly structured than the original graph. Finally, it produces an overlapping graph partitioning of the interaction network, implying that proteins may be present in multiple functional modules.

Ucar *et al.*'s work proposed a soft clustering method using hub-induced subgraphs [58]. Their approach consists of two stages. In the first stage, they refine the PPI graph to improve functional modularity, using hub-induced subgraphs. They employ the edge betweenness measure to identify dense regions within the neighborhoods. In the second stage, they cluster the refined graph using traditional algorithms. Their end goal is to isolate components with high degree of overlap with known functional modules. An additional advantage of the refinement process is its ability to perform soft clustering of hub proteins. Owing to this approach, they improved functional modularity in PPI network.

Other soft clustering for PPI is an ensemble framework [7]. They construct a variant of the PCA-agglo consensus algorithm to perform soft clustering of proteins, which allows proteins to belong to multiple clusters. The hard agglomerative algorithm places each protein into the most likely cluster to satisfy a clustering criterion. However, it is possible for a protein to belong to many clusters with varying degrees. The probability of a protein belonging to an alternate cluster can be expressed as a factor of its distance from the nodes in the cluster. If a protein has sufficiently strong interactions with the proteins that belong to a particular cluster, then it can be considered amenable to multiple memberships.

#### 3.3.4 Extracting Signaling Feature Data from Multi-data Sources

STN have a complex two-level signaling machinery. The first level of complexity in cellular signaling constructs from the large number of molecules and multiple types of interactions between them. The second layer of complexity of signaling biochemistry is apparent from the fact that signaling proteins often contain multiple functional domains, thus enabling each to interact with numerous downstream targets [18]. Considering these complexities, we extracted the following structured data of signaling features.

1. Protein-protein interactions (PPI): the upper level consists of the components as interfaces to transmit signals. PPI data were extracted from Reactome database[1].
2. Domain-domain interactions (DDI): the deeper level consists of the functional domains that perform as the basic elements in signal transduction. DDI data were extracted from iPfam database [2].
3. Signaling domain-domain interactions: the functional level consists of signaling domains (specific functional domains) that act as key factors to transduce signals inside STN. Signaling DDI data were extracted from SMART database[3] and referred in [46].

[1] www.reactome.org/

[2] www.sanger.ac.uk/Software/Pfam/iPfam/

[3] smart.embl-heidelberg.de/

**Table 1.** List of signaling features and their corresponding data sources

| Feature | Database | Description of database |
|---|---|---|
| Protein-protein interactions | Reactome database [30] | An online bioinformatics database of biology described in molecular terms. The largest set of entries refers to human biology |
| Domain-domain interactions | iPfam database [19] | A resource describing domain-domain interactions observed in PDB entries |
| Signaling domains | SMART database [34] and Pawson's dataset [46] | SMART allows the identification and annotation of genetically mobile domains and the analysis of domain architectures |
| Function of protein | Uniprot database [9] | The world's most comprehensive catalog of information on proteins |

Functions of proteins in STN were also extracted from Uniprot database[4] in terms of keywords.

The extracted data are in different types, e.g., the numerical type for number of PPI, interaction generality, number of signaling DDI or categorical type for protein functions. Those data have complex relations, such as a protein may have many interactions and then each interaction may have many DDI. In a domain interaction, the interacting partner may be a signaling domain or not. To exploit these relations, after extracting data from multi-data sources, we weighted and normalized these relational data by weight functions. Table 2 shows these proposed weight functions and the corresponding explanations.

- PPI weight function ($w_{ppi}$): The topological relation of proteins in the PPI network was extracted in terms of the numbers of interactions of each partner and the interaction generality.
- Signaling DDI weight function ($w_{Sddi}$): The relation between a PPI and their domains was exploited to study more deeply STN in terms the number of DDI and signaling DDI mediating this interactions.
- Keyword weight function ($w_{func}$): The relation of a PPI and protein functions was considered in terms of the keywords tagged in each partner and the keywords shared between them.

#### 3.3.5 Combining Signaling Feature Data to Construct STN using Soft-Clustering

After weighing signaling features, it is necessary to combine them all in a unified computational scheme to take advantage of those data. We integrated these data and represented them in forms of feature vectors. Each interaction has its own feature vector that has three elements corresponding to three features, $v_{ij} = \{w_{ppi}, w_{Sddi}, w_{func}\}$. Subsequently, we employed a soft-clustering algorithm to cluster the interactions based on their features vectors. Soft-clustering can construct STN and detect the overlapping components that can not be found by traditional hard-clustering. Note that we

[4] www.uniprot.org/

**Table 2.** Signaling features and their weight functions

| Weight functions | Notations and explanation |
|---|---|
| $w_{ppi}(p_{ij}) = \frac{g_{ij}^2}{n_i * n_j}$ | $g_{ij}$: Interaction generality, the number of proteins that interact with just two interacting partners, $p_i$ and $p_j$.<br>$n_i$: The number of protein-protein interactions of the protein $p_i$. |
| $w_{Sddi}(p_{ij}) = \frac{n_{Sddi}+1}{n_{ddi}+1}$ | $n_{Sddi}$: The number of signaling domain-domain interactions shared between two interacting proteins.<br>$n_{Sddi}$: The number of domain-domain interactions shared between two interacting proteins. |
| $w_{func}(p_{ij}) = \frac{k_{ij}^2}{k_i * k_j}$ | $k_{ij}$: The number of sharing keywords $k_{ij}$ of two interacting partners, $p_i$ and $p_j$.<br>$k_i$: The number of keywords of the protein $p_i$. |

used Mfuzz software package [33] to implement fuzzy c-means (FCM) clustering algorithm in our experiments. Fuzzy c-means (FCM) clustering algorithm is a popular soft-clustering algorithm. It is based on the iterative optimization of an objective function to minimize the variation of objects within clusters. First, it generates accessible internal cluster structures, i.e. it indicates how well corresponding clusters represent genes/proteins. Second, the overall relation between clusters, and thus a global clustering structure, can be defined. Additionally, soft clustering is more noise robust and a priori pre-filtering of genes can be avoided. This prevents the exclusion of biologically relevant genes from the data analysis.

Figure 1 summarizes the key idea of our method that does (1) extracting and weighing signaling features and (2) integrating and soft-clustering them into STN. Given a large protein-protein interaction network $\mathfrak{N}$, the outputs of our method are STN, which are the subgraphs of edges as protein interactions and node as proteins. Step 1 is to obtain the binary interactions from the protein-protein interaction network $\mathfrak{N}$. From Step 2 to Step 5 is to do the first task, extracting and then weighing signaling data features by functions shown in Table 2. These steps were done for all binary interactions to exploit the relations between PPI and signaling features. Step 6 and Step 7 are to perform the second task, combining weighted feature data, representing them in forms of feature vectors $v_{ij} = \{w_{ppi}, w_{Sddi}, w_{func}\}$ and lastly doing soft-clustering into STN $\mathscr{S}$. STN $\mathscr{S}$ are returned in Step 8.

### 3.4 Evaluation

To evaluate the performance of the method, we consider a complex PPI network to detect STN out of other biological processes. The tested PPI network does not contain only signaling processes but also other biological processes functioned inside the network as the nature in cells. The clustered results should reflect these complicated phenomenon, well construct signaling processes and find overlapping components. We extracted five heterogeneous processes in Reactome database and the experimental results demonstrated that our method effectively constructed signaling processes from the PPI network.

**Algorithm 1.** The proposed method to construct STN from PPI networks using soft-clustering and multi-signaling feature data.

**Input:**
Protein-protein network $\mathfrak{N}$.
Set of features $\mathscr{F} \subset \{f_{ppi}, f_{Sddi}, f_{func}\}$.
**Output:**
Set of signal transduction networks $\mathscr{S}$.

1: Extract binary interactions $\{p_{ij}\}$ from the protein-protein network $\mathfrak{N}$. $\mathscr{P} := \{p_{ij}\}$.
2: For each interaction $p_{ij} \subset \mathscr{P}$
3: Extract and formalize data for the PPI data feature $f_{ppi}$
Calculate the number of interactions $n_i$, $n_j$ of each interacting partner $p_i$ and $p_j$, respectively.
Calculate the interacting generality $g_{ij}$ of interaction $p_{ij}$.
Weigh the feature $f_{ppi}$ by the numbers $n_i$, $n_j$, and $g_{ij}$.
4: Extract and formalize data for the signal DDI feature $f_{Sddi}$
Calculate the number of sharing domain-domain interactions $n_{ddi}$ of two interacting partners, $p_i$ and $p_j$.
Calculate the number of sharing signaling domain-domain interactions $n_{Sddi}$ of two interacting partners, $p_i$ and $p_j$.
Weigh the feature $f_{Sddi}$ by the numbers $n_{ddi}$, $n_{Sddi}$.
5: Extract and formalize data for the function data feature $f_{func}$
Calculate the number of keywords $k_i$, $k_j$ of each interacting partner $p_i$ and $p_j$, respectively.
Calculate the number of sharing keywords $k_{ij}$ of two interacting partners, $p_i$ and $p_j$.
Weigh the feature $f_{func}$ by the numbers $k_i$, $k_j$, and $k_{ij}$.
6: Combine and represent the all features in the feature vectors $v_{ij} = \{f_{ppi}, f_{Sddi}, f_{func}\}$.
7: Apply a soft-clustering algorithm with the set of feature vectors $\{v_{ij}\}$ to cluster interactions $p_{ij}$ into signal transduction networks $\mathscr{S}$.
8: **return** $\mathscr{S}$.

**Table 3.** Five tested biological processes and some related information

| Reactome annotation | Description | #Proteins | #Interactions |
|---|---|---|---|
| REACT_1069 | Post-translational protein modification | 40 | 23 |
| REACT_1892 | Elongation arrest and recovery | 31 | 68 |
| REACT_498 | Signaling by Insulin receptor | 39 | 44 |
| REACT_769 | Pausing and recovery of elongation | 31 | 68 |
| REACT_9417 | Signaling by EGFR | 40 | 25 |

### 3.4.1 Experiments for Human STN Construction

The Reactome database consists of 68 *Homo sapiens* biological processes of 2,461 proteins. They also published 6,188 protein interactions, among those there are 6,162 interactions participating in biological processes. Investigating known biological processes in Reactome database, there are 636 proteins partaking in at least 2 different processes, 400 proteins in at least 3 processes, 119 proteins in 5 processes. These facts prove that there exists lot of proteins and their interactions overlapping among these processes.

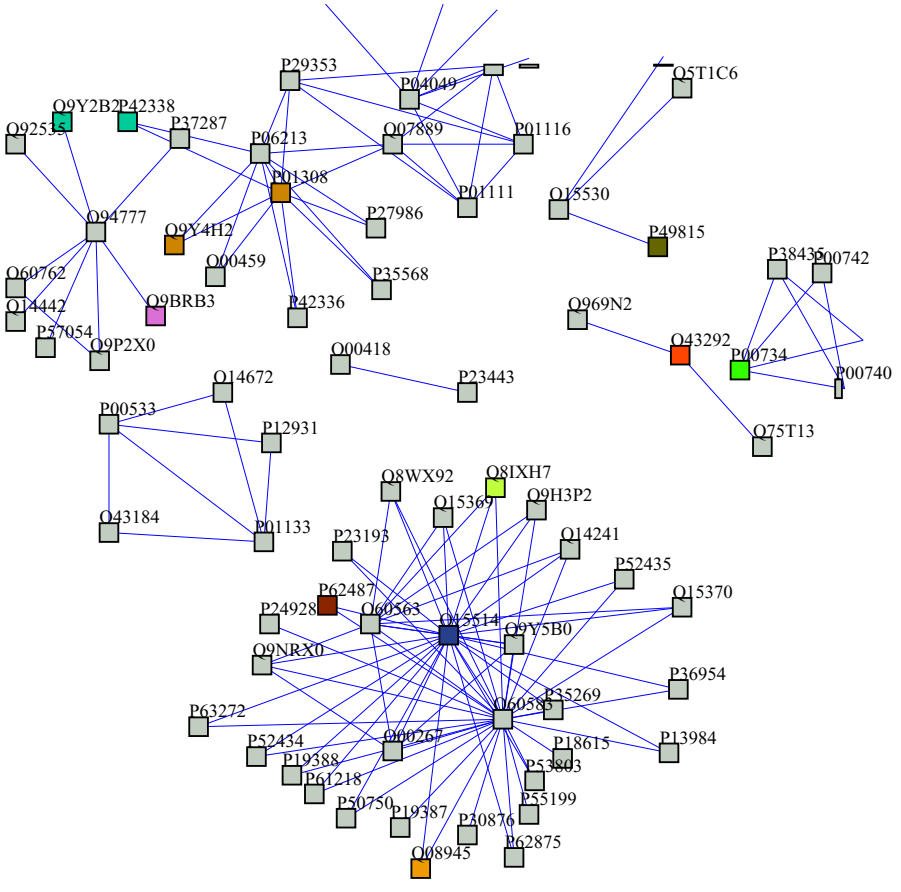



**Fig. 2.** Protein interaction networks of the five testing processes

In our experiments, we extracted a group of five biological processes which have from 30 to 50 proteins and include signaling networks. Table 3 shows some information related to these five processes. Totally, this group consists of 145 distinct interactions of 140 distinct proteins. Among these processes, there are overlapping interactions and proteins. Figure 2 illustrates the interaction network of five processes.

Proteins taking part in these processes are extracted and looked for their interactions in the Reactome interactions set. We strictly extracted only the interactions that have both interacting partners joining in processes because the method considers the proteins but more importantly their interactions. The extracted interactions and their signaling features were then input in the soft-clustering algorithm.

In this work, we applied Mfuzz software package to run fuzzy c-means (FCM) clustering algorithm. It is based on the iterative optimization of an objective function to minimize the variation of objects within clusters [33]. As a result, fuzzy c-means produces gradual membership values $\mu_{ij}$ of an interaction $i$ between 0 and 1 indicating the degree of membership of this interaction for cluster $j$. This strongly contrasts with

hard-clustering, e.g., the commonly used k-means clustering that generates only membership values $\mu_{ij}$ of either 0 or 1. Mfuzz is constructed as an R package implementing soft clustering tools. The additional package Mfuzzgui provides a convenient TclTk-based graphical user interface.

Concerning the parameters of Mfuzz, the number of clusters was 5 (because we are considering 5 processes) and the so-called fuzzification parameter $\mu_{ij}$ was chosen 0.035 (because the testing data is not noisy).

### 3.4.2 Experimental Results and Discussion for Human STN Construction

Actually, two processes REACT_1892 and REACT_498 share the same set of proteins and the same interactions as well. Also, two signaling processes, REACT_9417 and REACT_498 have 16 common interactions. Nevertheless, the process 'post-translational protein modification' is separated with the rest processes. In such complex case, the method should construct STN effectively and detect the overlaps among STN.

The threshold to output clusters is 0.1. The threshold means that if the membership of an interaction $i$ with a cluster $j$ $\mu_{ij} \geq 0.1$, this interaction highly correlates with the cluster $j$ and it will be clustered to cluster $j$. Five clusters are outcomes and then matched with 5 processes. The results are shown in Table 4.

**Table 4.** Clustered results for five tested biological processes

| Process | True positive[1] | False negative[2] | False positive[3] | #Overlap_Int[4] |
|---|---|---|---|---|
| REACT_1069 | 0.565 | 0.174 | 0.435 | 3/0 |
| REACT_1892 | 1.000 | 0.103 | 0.000 | 70/68 |
| REACT_498 | 0.818 | 0.068 | 0.182 | 17/16 |
| REACT_769 | 1.000 | 0.103 | 0.000 | 70/68 |
| REACT_9417 | 0.960 | 0.120 | 0.040 | 17/16 |

1 True positive: the number of true interactions clustered/the number of interactions of the fact process.
2 Faulse negative: the number of interactions missed in fact processes/the number of interactions of the fact process.
3 False positive : the number of false interactions clustered/the number of interactions of the fact process.
4 #Overlap_Int: the number of overlapping interactions among the clusters/the number of overlapping interactions among the fact processes.

Table 4 shows that we can construct signal transduction networks with the small error and can detect the nearly exact number of overlapping interactions. The combination of signaling feature data distinguished signaling processed from other biological processes and soft-clustering detected the overlapping components. When we checked the overlapping interactions among the clusters, there were exact 16 interactions that are shared in two signaling processes 'signaling by Insulin receptor' and 'signaling by EGFR'. In addition, the same interaction set of the process 'elongation arrest recovery' and the process 'pausing and recovery of elongation' are found in their clusters. In fact, REACT_1069 does not overlap other processes but the results return three overlapping interactions, i.e., one with REACT_1892 and REACT_769 and two with REACT_498 and REACT_9417.

Analyzing the case of interaction (P00734, P00734) shared among REACT_1069, REACT_498 and REACT_9417, we found some interesting findings. Protein P00734 (Prothrombin) functions in blood homeostasis, inflammation and wound healing and joins in biological process as cell surface receptor linked signal transduction (have GO term GO:0007166). In Reactome database, interaction (P00734, P00734) does not happen in the processes REACT_498 and REACT_9417, however according to the function of P00734, it probably partakes in one or two signaling processes REACT_498 and REACT_9417.

Although the experiment carried out a case study of five biological processes; the proposed method is flexible to be applied to the larger scale of human interaction networks. In the intricate relations of many biological processes, the proposed method can well construct signal transduction networks.

We proposed a general framework to construct STN from multiple signaling feature data using soft-clustering. The experiments with various parameters and other soft-clustering algorithms (not only FCM algorithm in Mfuzz) should be tested.

## 4 Some Results of Yeast STN Reconstruction

In addition to the work on human STN, we also carried out the work on yeast STN. This work consist of two parts: (1) signaling DDI prediction using ILP and (2) MARK yeast reconstruction.

This work concentrates on study STN for *Saccharomyces cerevisiae* – a budding yeast. The objective of this work is twofold. One objective is to present a method of predicting signaling domain-domain interactions (signaling DDI) using inductive logic programming (ILP), and the other is to present a method of discovering signal transduction networks (STN) using signaling DDI.

For signaling DDI prediction, we first examine five most informative genome databases, and extract more than twenty four thousand possible and necessary ground facts on signaling protein domains. We then employ inductive logic programming (ILP) to infer efficiently signaling DDI. Sensitivity (88%) and accuracy (83%) obtained from 10-fold cross validation show that our method is useful for predicting signaling domain interactions. Studying yeast MAPK pathways, we predicted some new signaling DDI that

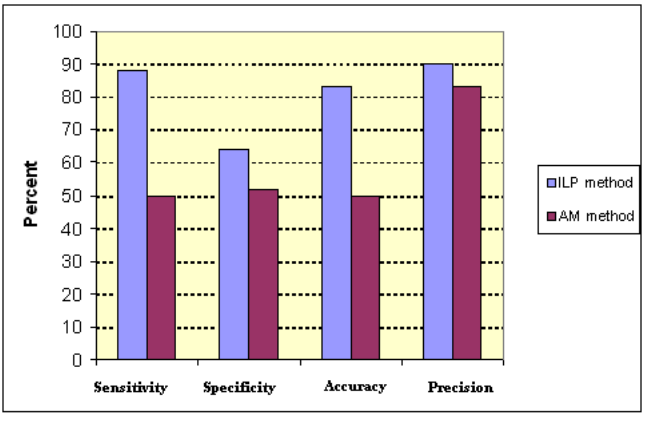



**Fig. 3.** Performance of ILP method (*minpos* = 3 and *noise* = 0) compared with AM methods for signaling DDI prediction

do not exist in the well-known InterDom database. Assuming all proteins in STN are known, we preliminarily build up signal transduction networks between these proteins based on their signaling domain interaction networks. We can mostly reconstruct the STN of yeast MAPK pathways from the inferred signaling domain interactions with coverage of 85%.

Figure 3 shows the results for signaling domain-domain interactions. Our experimental results obtained higher sensitivity, specificity, accuracy and precision compared with AM method [56].

From predicted (signaling) domain interaction networks, we raise the question of how completely they cover the STN, and how to reconstruct STN using signaling DDI. Our motivation was to propose a computational approach to discover more reliable and stable STN using signaling DDI. When studying yeast MAPK pathways, the results of our work are considerable.

All extracted domains of proteins in MAPK pathways are inputs (testing examples) in our proposed predictor using ILP method [42]. With 32 proteins appearing in MAPK pathways, we extracted 29 different protein domains, and some of them are shared among proteins. Some domains are determined to be signaling domains, such as domain *pf00069* belonging to many proteins, for example, *ste11_yeast, fus3_yeast* or

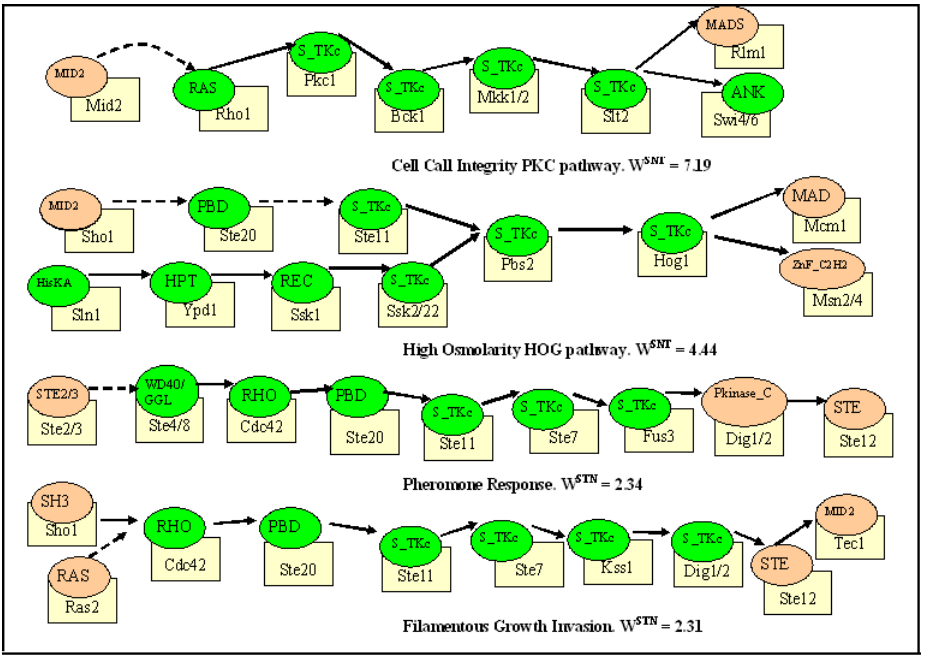



**Fig. 4.** MAPK signal transduction pathways in yeast covered by signaling DDI networks. The rectangles denote proteins, the ellipses illustrate their domains and the signaling domains are depicted in dark. The signaling DDI are the lines with arrows, the missing interactions are dashed lines with arrows.

**Table 5.** Results of predicted signaling DDI in the yeast MAPK pathways

| The yeast MARK pathways pathways | Percentage of signaling DDI predicted | #CYGD PPI covered | #DIP PPI covered |
|---|---|---|---|
| Cell Wall Integrity PKC | 88% | 39 | 47 |
| Pheromone Response | 88% | 41 | 42 |
| Filamentous Growth | 88% | 40 | 38 |
| Invasion High Osmolarity HOG | 80% | 40 | 53 |

*pbs_2*, etc., and some of them are not signaling domains, such as *TEA* or *MID2*. Figure 4 shows yeast MAPK (mitogen-activated protein kinase) covered by signaling domain interactions. MAPK pathways involve pheromone response, filamentous growth, and maintenance of cell wall integrity pathways. Table 5 shows the results of predicted signaling DDI when reconstructing STN for the yeast MAPK pathways. Moreover, among predicted signaling DDI for yeast MAPK pathways, there are some DDI which are newly discovered, when compared with the InterDom database. For example, our predicted DDI *(pf00071,pf00768), (pf00768,pf00069), (pf00433,pf02200)* do not exist in the InterDom database.

Evaluating signaling domain interactions predicted from the testing set of MAPK domains, 88% of protein relations in the Cell Wall Integrity PKC pathway, the Pheromone Response pathway, and the Filamentous Growth pathway are covered, and the Invasion High Osmolarity HOG pathway has coverage of 80%. Outstandingly, lots of domain interactions are found in which their corresponding proteins interacted in DIP (Database of Interacting Proteins) [5] and/or in CYGD (Comprehensive Yeast Genome Database)footnotehttp://mips.gsf.de/genre/proj/yeast/, for example, seven signaling domain interactions in the Cell Wall Integrity PKC pathway belong to 39 protein-protein interactions in CYGD database, and also belong to 47 protein-protein interactions in DIP. For estimating the reliability of STN, the reliability score $W^{STN}$ (see in [42]) was calculated for yeast MAPK pathways. The reliability score of the Cell Wall Integrity PKC pathway is the highest with $W^{STN} = 7.19$.

The work is the first work that took effort to predict signaling DDI. The results on yeast STN confirmed the role of signaling domain-domain interactions and it

## 5 Outlook

The previous sections presented our work testing the example of five biological processes for human or a single pathway - MARK pathway for yeast. However, the methods are easy to be applied to the large-scale protein interaction networks. In the intricate relationships with various processes, the proposed method can well detect signal transduction networks. The preliminary results encourage the further studies on biological complex systems.

1. Consider the whole interaction networks or some functional sub-networks, it is interesting to not only reconstruct the known signal transduction networks but also

[5] http://dip.doe-mbi.ucla.edu/

model the new ones. The components (proteins and their interactions) that are shared among these networks to perform various functions in different biological processes can be further functionally investigated.
2. Given starting nodes (e.g., membrane proteins) and ending nodes (e.g, transcription factors), the proposed method can specify the signal transduction networks and then discover complete signaling pathways.
3. In human disease study, human protein interaction networks, signal transduction pathways and diseases closely associate with each other. Signaling network dysfunction can result in abnormal cellular transformation or differentiation, often producing a physiological disease outcome. The potential work on identification of disease-related subnetworks are significant and can be carried out through the constructed signal transduction networks.
4. Some other data mining methods in relational learning and statistical learning can be applied to supplement the work.
5. Our proposed methods are flexible to integrate other useful biological features and apply to other organisms.

## 6 Summary

In this paper, we have presented a the study on mining multiple data to reconstruct STN. The soft-clustering method was used to construct signal transduction networks from protein-protein networks. Many structured data of signaling features were extracted, integrated and exploited. The experimental results demonstrated that our proposed method can construct STN effectively. The overlapping parts among STN were well detected. As proposing the general framework to construct signal transduction networks from protein interaction networks using soft-clustering, the method should be more carefully tested with various parameters and other algorithms (not only FCM algorithm in Mfuzz). Other computational measures also need calculated to better demonstrate efficiency of the method. Yet, the experimental results show that the proposed method is promising to construct signal transduction networks from protein-protein interaction networks. On the other hand, the work on the yeast STN proposed an alternative way to study in deep the mechanism of STNs in terms of signaling domain interactions. These work are expected to provide insights of the cell signaling that will be useful for studying systems biology.

**Acknowledgement.** We would like to respectfully thank to Professor Kenji Satou for all helpful advices and discussion. Also, we highly appreciate all comments with Dr. Dang-Hung Tran and Dr. Jose Clemente.

## References

1. Molecular Biology of the Cell. Garland Science (2007)
2. Alberts, B.: Molecular biology of the cell. Garland Science (2002)

3. Alfarano, C., Andrade, C.E., Anthony, K., Bahroos, N., Bajec, M., Bantoft, K., Betel, D., Bobechko, B., Boutilier, K., Burgess, E., Buzadzija, K., Cavero, R., D'Abreo, C., Donaldson, I., Dorairajoo, D., Dumontier, M.J., Dumontier, M.R., Earles, V., Farrall, R., Feldman, H., Garderman, E., Gong, Y., Gonzaga, R., Grytsan, V., Gryz, E., Gu, V., Haldorsen, E., Halupa, A., Haw, R., Hrvojic, A., Hurrell, L., Isserlin, R., Jack, F., Juma, F., Khan, A., Kon, T., Konopinsky, S., Le, V., Lee, E., Ling, S., Magidin, M., Moniakis, J., Montojo, J., Moore, S., Muskat, B., Ng, I., Paraiso, J.P., Parker, B., Pintilie, G., Pirone, R., Salama, J.J., Sgro, S., Shan, T., Shu, Y., Siew, J., Skinner, D., Snyder, K., Stasiuk, R., Strumpf, D., Tuekam, B., Tao, S., Wang, Z., White, M., Willis, R., Wolting, C., Wong, S., Wrong, A., Xin, C., Yao, R., Yates, B., Zhang, S., Zheng, K., Pawson, T., Ouellette, B.F.F., Hogue, C.W.V.: The Biomolecular Interaction Network Database and related tools 2005 update. Nucleic Acids Research 33(suppl. 1), D418–D424 (2005), http://nar.oxfordjournals.org/content/33/suppl_1/D418.abstract, doi :10.1093/nar/gki051
4. Allen, E.E., Fetrow, J.S., Daniel, L.W., Thomas, S.J., John, D.J.: Algebraic dependency models of protein signal transduction networks from time-series data. Journal of Theoretical Biology 238(2), 317–330 (2006)
5. Arnau, V., Mars, S., Marin, I.: Iterative Cluster Analysis of Protein Interaction Data. Bioinformatics 21(3), 364–378 (2005), http://bioinformatics.oxfordjournals.org/cgi/content/abstract/21/3/364
6. Asthagiri, A.R., Lauffenburger, D.A.: Bioengineering models of cell signaling. Annual Review of Biomedical Engineering 2(1), 31–53 (2000), http://arjournals.annualreviews.org/doi/abs/10.1146/annurev.bioeng.2.1.31, doi:10.1146/annurev.bioeng.2.1.31
7. Asur, S., Ucar, D., Parthasarathy, S.: An ensemble framework for clustering protein protein interaction networks. Bioinformatics 23(13), i29–i40 (2007), doi:10.1093/bioinformatics/btm212
8. Bader, G.D., Hogue, C.W.: An automated method for finding molecular complexes in large protein interaction networks. BMC Bioinformatics 4(1), 2 (2003), http://dx.doi.org/10.1186/1471-2105-4-2, doi:10.1186/1471-2105-4-2
9. Bairoch, A., Apweiler, R., Wu, C., Barker, W., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M., Martin, M., Natale, D., O'Donovan, C., Redaschi, N., Yeh, L.: The universal protein resource (uniprot). Nucleic Acids Research 33, D154–D159 (2005)
10. Bauer, A., Kuster, B.: Affinity purification-mass spectrometry: Powerful tools for the characterization of protein complexes. Eur. J. Biochem. 270(4), 570–578 (2003)
11. Ben-Hur, A., Noble, W.S.: Kernel methods for predicting protein-protein interactions. Bioinformatics 21(suppl.1), i38–i46 (2005), http://bioinformatics.oxfordjournals.org/cgi/content/abstract/21/suppl1/i38, doi: 10.1093/bioinformatics/bti1016
12. Bhalla, U.S.: Understanding complex signaling networks through models and metaphors. Progress in Biophysics and Molecular Biology 81(1), 45–65 (2003), http://www.sciencedirect.com/science/article/B6TBN-47C7506-3/2/2267fd452dc127061f9236c3d42067f0, doi:10.1093/bioinformatics/bti1016
13. Bock, J.R., Gough, D.A.: Predicting protein-protein interactions from primary structure. Bioinformatics 17(5), 455–460 (2001)
14. Brown, K., Jurisica, I.: Unequal evolutionary conservation of human protein interactions in interologous networks. Genome Biology 8(5), R95 (2007), doi:10.1186/gb-2007-8-5-r95

15. Cannataro, M., Guzzi, P.H., Veltri, P.: Protein-to-protein interactions: Technologies, databases, and algorithms. ACM Comput. Surv. 1:1-1:36 (2010), http://doi.acm.org/10.1145/1824795.1824796, doi: http://doi.acm.org/10.1145/1824795.1824796
16. Chatr-aryamontri, A., Ceol, A., Palazzi, L.M., Nardelli, G., Schneider, M.V., Castagnoli, L., Cesareni, G.: MINT: the Molecular INTeraction database. Nucl. Acids Res. 35(suppl.1), D572–D574 (2007), doi: 10.1093/nar/gkl950
17. Chen, X., Liu, M.: Prediction of protein-protein interactions using random decision forest framework. Bioinformatics 21(24), 4394–4400 (2005), doi:10.1093/bioinformatics/bti721
18. Eungdamrong, N.J., Iyenga, R.: Modeling cell signaling networks. Biology of the Cell 96(5), 355–362 (2004)
19. Finn, R.D., Marshall, M., Bateman, A.: iPfam: visualization of protein protein interactions in PDB at domain and amino acid resolutions. Bioinformatics 21(3), 410–412 (2005), http://bioinformatics.oxfordjournals.org/content/21/3/410.abstract, doi: 10.1093/bioinformatics/bti011
20. Fukuda, K., Takagi, T.: Knowledge representation of signal transduction pathways. Bioinformatics 17(9), 829–837 (2001), doi:10.1093/bioinformatics/17.9.829
21. Futschik, M., Carlisle, B.: Noise-robust soft clustering of gene expression time-course data. J. Bioinform. Comput. Biol. 3(4), 965–988 (2005)
22. Gagneur, J., Casari, G.: From molecular networks to qualitative cell behavior. FEBS Letters 579(8), 1861–1871 (2005), http://www.sciencedirect.com/science/article/B6T36-4FG2TYJ-5/2/904b1a2f8f6bc73b06ab00e9e4bfe2f8, doi: 10.1016/j.febslet.2005.02.007; System Biology
23. Gagneur, J., Krause, R., Bouwmeester, T., Casari, G.: Modular decomposition of protein-protein interaction networks. Genome Biol. 5(8) (2004), http://dx.doi.org/10.1186/gb-2004-5-8-r57
24. Getz, G., Levine, E., Domany, E.: Coupled two-way clustering analysis of gene microarray data. Proceedings of the National Academy of Sciences of the United States of America 97(22), 12079–12084 (2000), http://www.pnas.org/content/97/22/12079.abstract
25. Gomez, S.M., Lo, S., Rzhetsky, A.: Probabilistic Prediction of Unknown Metabolic and Signal-Transduction Networks. Genetics 159(3), 1291–1298 (2001)
26. Hartuv, E., Shamir, R.: A clustering algorithm based on graph connectivity. Inf. Process. Lett. 76, 175–181 (2000), http://portal.acm.org/citation.cfm?id=364456.364469, doi:10.1016/S0020-0190(00)00142-3
27. Ihekwaba, A.E., Nguyen, P.T., Priami, C.: Elucidation of functional consequences of signalling pathway interactions. BMC Bioinformatics 10(370)
28. Ito, T., Chiba, T., Ozawa, R., Yoshida, M., Hattori, M., Sakaki, Y.: A comprehensive two-hybrid analysis to explore the yeast protein interactome. Proc. Natl. Acad. Sci. USA 98, 4569–4574 (2001)
29. Jansen, R., Yu, H., Greenbaum, D., Kluger, Y., Krogan, N.J., Chung, S., Emili, A., Snyder, M., Greenblatt, J.F., Gerstein, M.: A Bayesian Networks Approach for Predicting Protein-Protein Interactions from Genomic Data. Science 302(5644), 449–453 (2003), http://www.sciencemag.org/cgi/content/abstract/302/5644/449, doi:10.1126/science.1087361
30. Joshi-Tope, G., Gillespie, M., Vastrik, I., D'Eustachio, P., Schmidt, E., de Bono, B., Jassal, B., Gopinath, G., Wu, G., Matthews, L., Lewis, S., Birney, E., Stein, L.: Reactome: a knowledgebase of biological pathways. Nucl. Acids Res. 33(suppl.1), D428–D432 (2005), doi:10.1093/nar/gki072

31. King, A.D., Prulj, N., Jurisica, I.: Protein complex prediction via cost-based clustering. Bioinformatics 20(17), 3013–3020 (2004), http://bioinformatics.oxfordjournals.org/content/20/17/3013.abstract, doi:10.1093/bioinformatics/bth351
32. Korcsmaros, T., Farkas, I.J., ad Petra Rovo, M.S.S., Fazekas, D., Spiro, Z., Bode, C., Lenti, K., Vellai, T., Csermely, P.: Uniformly curated signaling pathways reveal tissue-specific cross-talks and support drug target discovery. Bioinformatics 26(16), 2042–2050 (2010), http://bioinformatics.oxfordjournals.org/content/26/16/2042.abstract, doi:10.1093/bioinformatics/btq310
33. Kumar, L., Futschik, M.: Mfuzz: A software package for soft clustering of microarray data. Bioinformation 2(1), 5–7 (2007)
34. Letunic, I., Doerks, T., Bork, P.: SMART 6: recent updates and new developments. Nucleic Acids Research 37(suppl. 1), D229–D232 (2009), http://nar.oxfordjournals.org/content/37/suppl_/D229.abstract, doi: 10.1093/nar/gkn808
35. Li, Y., Agarwal, P., Rajagopalan, D.: A global pathway crosstalk network. Bioinformatics 24(12), 1442–1447 (2008), doi: 10.1093/bioinformatics/btn200
36. Lin, C., Cho, Y., Hwang, W., Pei, P., Zhang, A.: Clustering methods in protein-protein interaction network. In: Knowledge Discovery in Bioinformatics: Techniques, Methods and Application (2006)
37. Liu, Y., Zhao, H.: A computational approach for ordering signal transduction pathway components from genomics and proteomics data. BMC Bioinformatics 5(158) (2004), http://dx.doi.org/10.1186/1471-2105-5-158, doi:10.1186/1471-2105-5-158
38. Matthews, L.R., Vaglio, P., Reboul, J., et al.: Identification of potential interaction networks using sequence-based searches for conserved protein-protein interactions or 'interologs'. Genome Res. 11(12), 2120–2126 (2001)
39. von Mering, C., Huynen, M., Jaeggi, D., Schmidt, S., Bork, P., Snel, B.: STRING: a database of predicted functional associations between proteins. Nucleic Acids Research 31(1), 258–261 (2003), http://nar.oxfordjournals.org/content/31/1/258.abstract, doi:10.1093/nar/gkg034
40. Neves, S.R., Iyengar, R.: Modeling Signaling Networks. Sci. STKE 2005(281), tw157 (2005), http://stke.sciencemag.org/cgi/content/abstract/sigtrans;2005/281/tw157, doi:10.1126/stke.2812005tw157
41. Ng, S.K., Tan, S.H.: Discovering protein-protein interactions. Journal of Bioinformatics and Computational Biology 1(4), 711–741 (2003)
42. Nguyen, T., Ho, T.: Discovering signal transduction networks using signaling domain-domain interactions. Genome Informatics 17(2), 35–45 (2006)
43. Nguyen, T., Ho, T.: An Integrative Domain-Based Approach to Predicting Protein-Protein Interactions. Journal of Bioinformatics and Computational Biology 6 (2008)
44. Nicolau, M., Tibshirani, R., Brresen-Dale, A.L., Jeffrey, S.S.: Disease-specific genomic analysis: identifying the signature of pathologic biology. Bioinformatics 23(8), 957–965 (2007), http://bioinformatics.oxfordjournals.org/content/23/8/957.abstract, doi:10.1093/bioinformatics/btm033
45. Pagel, P., Kovac, S., Oesterheld, M., Brauner, B., Dunger-Kaltenbach, I., Frishman, G., Montrone, C., Mark, P., Stumpflen, V., Mewes, H.W., Ruepp, A., Frishman, D.: The MIPS mammalian protein-protein interaction database. Bioinformatics 21(6), 832–834 (2005), http://bioinformatics.oxfordjournals.org/cgi/content/abstract/21/6/832, doi: 10.1093/bioinformatics/bti115
46. Pawson, T., Raina, M., Nash, N.: Interaction domains: from simple binding events to complex cellular behavior. FEBS Letters 513(1), 2–10 (2002)

47. Pellegrini, M., Marcotte, E.M., Thompson, M.J., et al.: Assining protein functions by comparative genome analysis: Protein phylogenetic profiles. Proc. Natl. Acad. Sci. USA 96(8), 4285–4288 (1999)
48. Pereira-Leal, J.B., Enright, A.J., Ouzounis, C.A.: Detection of functional modules from protein interaction networks. Proteins: Structure, Function, and Bioinformatics 54(1), 49–57 (2004), http://dx.doi.org/10.1002/prot.10505, doi:10.1002/prot.10505
49. Priami, C.: Algorithmic systems biology. Commun. ACM 52, 80–88 (2009)
50. Rives, A.W., Galitski, T.: Modular organization of cellular networks, vol. 100(3), pp. 1128–1133 (2003), http://www.pnas.org/content/100/3/1128.abstract, doi:10.1073/pnas.0237338100
51. Salwinski, L., Miller, C.S., Smith, A.J., Pettit, F.K., Bowie, J.U., Eisenberg, D.: Dip: The database of interacting proteins: 2004 update. Nucleic Acids Research 32, 449–451 (2004)
52. Samanta, M.P., Liang, S.: Predicting protein functions from redundancies in large-scale protein interaction networks. Proceedings of the National Academy of Sciences of the United States of America 100(22), 12,579–12,583 (2003), doi:10.1073/pnas.2132527100
53. Scott, J.D., Pawson, T.: Cell communication: The inside story. Scientific American (2000)
54. Smith, G.P.: Filamentous fusion phage: Novel expression vectors that display cloned antigens on the virion surface. Science 228(4705), 1315–1317 (1985)
55. Spirin, V., Mirny, L.A.: Protein complexes and functional modules in molecular networks. In: Proceedings of the National Academy of Sciences of the United States of America, vol. 100(21), 21, 123–12,128 (2003), 32324100, http://www.pnas.org/content/100/21/12123.abstract, doi:10.1073/pnas.20
56. Sprinzak, E., Margalit, H.: Correlated sequence-signatures as markers of protein-protein interaction. Journal of Molecular Biology 311(4), 681–692 (2001)
57. Steffen, M., Petti, A., Aach, J., D'haeseleer, P., Church, G.: Automated modelling of signal transduction networks. BMC Bioinformatics 3(34) (2002)
58. Ucar, D., Asur, S., Catalyurek, U.V., Parthasarathy, S.: Improving functional modularity in protein-protein interactions graphs using hub-induced subgraphs. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) PKDD 2006. LNCS (LNAI), vol. 4213, pp. 371–382. Springer, Heidelberg (2006), doi: 10.1093/bioinformatics/btm212
59. Uetz, P., Giot, L., Cagney, G., Mansfield, T.A., Judson, R.S., Knight, J.R., Lockshon, D., Narayan, V., Srinivasan, M., Pochart, P., Qureshi-Emili, A., Li, Y., Godwin, B., Conover, D., Kalbfleisch, T., Vijayadamodar, G., Yang, M., Johnston, M., Fields, S., Rothberg, J.M.: A comprehensive analysis of protein-protein interactions in saccharomyces cerevisiae. Nature 403(6770), 623–627 (2000), http://dx.doi.org/10.1038/35001009, doi:10.1038/35001009
60. Uetz, P., Vollert, C.: Protein-Protein Interactions. Encyclopedic Reference of Genomics and Proteomics in Molecular Medicine 17 (2006)
61. Van Dongen, S.: A new cluster algorithm for graphs. Tech. Rep. Technical Report INS-R0010, Center for Mathematics and Computer Science (CWI), Amsterdam (2000)
62. Zhao, X., Wang, R., Chen, L., Aihara, K.: Automatic modeling of signal pathways from protein-protein interaction networks. In: The Sixth Asia Pacific Bioinformatics Conference, pp. 287–296 (2008)

# Chapter 9
# Mining Epistatic Interactions from High-Dimensional Data Sets

Xia Jiang[1], Shyam Visweswaran[1], and Richard E. Neapolitan[2]

[1] Department of Biomedical Informatics, University of Pittsburgh
[2] Department of Computer Science, Northeastern Illinois University

**Abstract.** Genetic epidemiologists strive to determine the genetic profile of diseases. Two or more genes can interact to have a causal effect on disease even when little or no such effect can be observed statistically for one or even both of the genes individually. This is in contrast to Mendelian diseases like cystic fibrosis, which are associated with variation at a single genetic locus. This gene-gene interaction is called epistasis. To uncover this dark matter of genetic risk it would be pivotal to be able to discover epistatic relationships from data. The recent availability of high-dimensional data sets affords us unprecedented opportunity to make headway in accomplishing this. However, there are two central barriers to successfully identifying genetic interactions using such data sets. First, it is difficult to detect epistatic interactions statistically using parametric statistical methods such as logistic regression due to the sparseness of the data and the non-linearity of the relationships. Second, the number of candidate models in a high-dimensional data set is forbiddingly large. This paper describes recent research addressing these two barriers. To address the first barrier, the primary author and colleagues developed a specialized Bayesian network model for representing the relationship between features and disease, and a Bayesian network scoring criterion tailored to this model. This research is summarized in Section 2. To address the second barrier the primary author and colleagues developed an enhancement of Greedy Equivalent Search. This research is discussed in Section 3. Background is provided in Section 1.

## 1 Introduction

Genetic epidemiologists strive to determine the genetic profile of diseases. For example, the $\epsilon 4$ allele of the APOE gene has been established as a risk factor for late-onset Alzheimer's disease (Coon et al., 2007; Papassotiropoulos et al., 2006; Corder et al., 1993). However, often genes do not affect phenotype according to the simple rules developed by Mendel (Bateson, 1909). Rather two or more genes can interact to have a causal effect on phenotype even when little or no such effect can be observed statistically for one or even both of the genes individually. For example, (Rieman et al., 2007) found that the GAB2 gene seems to be statistically relevant to Alzheimer's disease when the $\epsilon_4$ allele of the

D.E. Holmes, L.C. Jain (Eds.): Data Mining: Found. & Intell. Paradigms, ISRL 25, pp. 187–209.
springerlink.com 

APOE gene is present, but GAB2 by itself exhibits no statistical relevance to the disease. This is in contrast to Mendelian diseases like cystic fibrosis, which are associated with variation at a single genetic locus.

This gene-gene interaction is called *epistasis*. Much of the genetic risk of many common diseases remains unknown and is believed to be due to epistasis. This is referred to as the *dark matter* of genetic risk (Galvin et al., 2010). To uncover this dark matter of genetic risk it would be pivotal to be able to discover epistatic relationships from data.

The *dimension* of a data set is the number of attributes in the data set. The recent availability of high-dimensional data sets affords us unprecedented opportunity to learn the etiology of disease from data. For example, the advent of high-throughput technologies has enabled *genome-wide association studies* (*GWAS* or *GWA* studies) (Wang et al., 1998; Matsuzaki et al., 2004), which involve sampling in cases and controls around 500,000 genetic loci. The government has invested heavily in studies that produce these high-dimensional data sets, and the initial results of these studies have been gratifying in that they have suggested a number of previously unsuspected etiologic pathways (Manolio, 2009). However, analysis of these data sets has not yet yielded the level of disease-associated feature discoveries originally anticipated (Wade, 2010). This could well be do to the difficulty with discovering epistatic interactions using such data sets.

There are two central barriers to successfully identifying genetic interactions using high-dimensional sets. First, it is difficult to detect epistatic interactions statistically using parametric statistical methods such as logistic regression due to the sparseness of the data and the non-linearity of the relationships (Velez et al., 2007). So we need to develop efficacious methods for evaluating candidate epistatic models. Second, the number of candidate models in a high-dimensional data set is forbiddingly large. For example, if we only examined all 1, 2, 3, and 4 loci models when there are 100,000 loci, we would need to examine about $4.17 \times 10^{18}$ models. Since we do not have the computational power to investigate so many models, we need efficient algorithms that enable us to only investigate the most promising ones.

This paper describes recent research addressing these two barriers. Since the research described here pertains to data sets containing many different possible causal features including both genetic and environmental factors, we will refer to the risk factors simply as *features*. The phenotype need not be disease status (e.g. it could be height or math ability); however, for the sake of focus we will use disease terminology throughout. To address the first barrier just identified, the primary author and colleagues developed a specialized Bayesian network model for representing the relationship between features and disease, and a Bayesian network scoring criterion tailored to this model (Jiang et al., 2010a). This research is summarized in Section 3. To address the second barrier the primary author and colleagues developed an enhancement of Greedy Equivalent Search (Chickering, 2003) called Multiple Beam Search (Jiang et al., 2010b). This research is discussed in Section 4. First, we provide some background.

## 2 Background

We review epistasis, GWAS, a well-known epistatic learning method called multifactor dimensionality reduction, and Bayesian networks.

### 2.1 Epistasis

Biologically, *epistasis* refers to gene-gene interaction when the action of one gene is modified by one or several other genes. Statistically, epistasis refers to interaction between genetic variants at multiple loci in which the net effect on disease from the combination of genotypes at the different loci is not accurately predicted by a combination of the individual genotype effects. In general, the individual loci may exhibit no marginal effects.

**Example 1.** *Suppose we have two loci $G_1$ and $G_2$, disease $D$, and the alleles of $G_1$ are $A$ and $a$, whereas those of $G_2$ are $B$ and $b$. Suppose further that we have the probabilities (relative frequencies in the population) in the following table:*

| | $AA$ (.25) | $Aa$ (.5) | $aa$ (.25) |
|---|---|---|---|
| $BB$ (.25) | 0.0 | 0.1 | 0.0 |
| $Bb$ (.5) | 0.1 | 0.0 | 0.1 |
| $bb$ (.25) | 0.0 | 0.1 | 0.0 |

*The entries in the table denote, for example, that*

$$P(D = yes|G_1 = Aa, G_2 = BB) = 0.1.$$

*The heading $AA$ (.25) means 25% of the individuals in the population have genotype $AA$. We also assume that $G_1$ and $G_2$ mix independently in the population (no linkage). We then have the following (we do not show the random variables $G_i$ for brevity):*

$$\begin{aligned} P(D = yes)|AA) &= P(D = yes|AA, BB)P(BB) + \\ &\quad P(D = yes|AA, Bb)P(Bb) + P(D = yes|AA, bb)P(bb) \\ &= 0.0 \times .25 + 0.1 \times 0.5 + 0.0 \times .25 = .05 \end{aligned}$$

$$\begin{aligned} P(D = yes)|Aa) &= P(D = yes|Aa, BB)P(BB) + \\ &\quad P(D = yes|Aa, Bb)P(Bb) + P(D = yes|Aa, bb)P(bb) \\ &= 0.1 \times .25 + 0.0 \times 0.5 + 0.1 \times .25 = .05 \end{aligned}$$

$$\begin{aligned} P(D = yes)|aa) &= P(D = yes|aa, BB)P(BB) + \\ &\quad P(D = yes|aa, Bb)P(Bb) + P(D = yes|aa, bb)P(bb) \\ &= 0.0 \times .25 + 0.1 \times 0.5 + 0.0 \times .25 = .05 \end{aligned}$$

*So if we look at $G_1$ alone no statistical correlation with D will be observed. The same is true if we look at $G_2$ alone. However, as can be seen from the above table, the combinations AABb, AaBB, Aabb, and aaBb make disease D more probable.*

It is believed that epistasis may play an important role in susceptibility to many common diseases (Galvin et al, 2010). For example, Ritchie et al. (2001) found a statistically significant high-order interaction among four polymorphisms from three estrogen pathway genes (COMT, CYP1B1, and CYP1A1) relative to sporadic breast cancer, when no marginal effect was observed for any of the genes.

### 2.2 Detecting Epistasis

It is difficult to detect epistatic relationships statistically using parametric statistical methods such as logistic regression due to the sparseness of the data and the non-linearity of the relationships (Velez et al., 2007). As a result, non-parametric methods based on machine-learning have been developed. Such methods include combinatorial methods, set association analysis, genetic programming, neural networks and random forests (Heidema et al., 2006).

Combinatorial methods search over all possible combinations of loci to find combinations that are predictive of the phenotype. The combinatorial method *multifactor dimensionality reduction* (*MDR*) (Hahn et al., 2005) combines two or more variables into a single variable (hence leading to dimensionality reduction); this changes the representation space of the data and facilitates the detection of nonlinear interactions among the variables. MDR has been successfully applied in detecting epistatic interactions in complex human diseases such as sporadic breast cancer, cardiovascular disease, and type II diabetes (Ritchie et al., 2001; Coffey et al., 2004; Cho et al., 2004).

A combinatorial method must focus on a relatively small number of loci to be tractable. For example, if we examined all 1, 2, 3, and 4 loci subsets of 100,000 loci, we would need to examine about $4.17 \times 10^{18}$ subsets. The successes of MDR were achieved by identifying relatively few relevant loci up front. For example, in the sporadic breast cancer discovery the focus was on five genes known to produce enzymes in the metabolism of estrogens.

### 2.3 High-Dimensional Data Sets

To uncover the dark matter barrier of genetic risk it would be pivotal to be able to discover epistatic relationships from data without identifying relevant loci up front. The recent availability of high-dimensional data sets affords us unprecedented opportunity to accomplish this. For example, the advent of high-throughput technologies has enabled *genome-wide association studies* (*GWAS* or *GWA* studies) (Wang et al., 1998; Matsuzaki et al., 2004). A *single nucleotide polymorphism* (*SNP*) is a DNA sequence variation occurring when a single nucleotide in the genome differs between members of a species. Usually, the less frequent allele must be present in 1% or more of the population for a site to

qualify as a SNP (Brooks, 1999). A GWAS involves sampling in an individual around 500,000 representative SNPs that capture much of the variation across the entire genome. The initial results of these studies has been gratifying. Over 150 risk loci have been identified in studies of more than 60 common diseases and traits. These associations have suggested previously unsuspected etiologic pathways (Manolio, 2009). For example, a GWAS in (Reiman et al, 2007) identified the GAB2 gene as a risk factor for Alzheimer's disease, and a GWAS in (Hunter et al., 2007) found four SNPs in intron 2 of FGFR2 highly associated with breast cancer. These initial GWAS successes were achieved by analyzing the association of each locus individually with the disease. Their success notwithstanding, analysis of high-dimensional data sets has not yet yielded the level of disease-associated feature discoveries originally anticipated (Wade, 2010). To realize the full potential of a GWAS and perhaps approach maximizing what we can discover from them, we need to analyze the effects of multiple loci on a disease (i.e. epistasis).

Realizing this, recently researchers have worked on developing methods for simultaneously analyzing the effects of multiple loci using high-dimensional data sets. *Lasso* is a shrinkage and selection method for linear regression (Tibshirani, 1996). It has been used successfully in problems where the number of predictors far exceeds the number of observations (Chen et al., 1998). So, researchers applied lasso to analyzing the multiple effects of loci on disease based on GWAS data (Wu et al., 2009; Wu et al. 2010). There are two difficulties with this procedure. First, as mentioned earlier, regression has difficulty handling a nonlinear epistatic relationship. Second, loci interactions with no marginal effects will not be detected at all unless we include terms for pairwise interactions. Wu et al. (2010) do this, but now we are faced with the combinatorial explosion problem discussed above. Another strategy involves using permutation tests (Zhang et al., 2009). These methods use standard statistic analysis to investigate different ensembles of two-loci analyses. Other methods include the use of ReliefF (Moore and White, 2007; Epstein and Haake, 2008), random forests (Meng et al. 2007), predictive rule inference (Wan et al., 2010), a variational Bayes algorithm (Longston et al. 2010), a Bayesian marker partition algorithm (Zhang and Liu, 2007), the Bayesian graphical method developed in (Verzilli et al., 2006), and the Markov blanket-base method discussed in (Han et al., 2009). The Bayesian graphical method does approximate model averaging using *Markov chain Monte Carlo* (*MCMC*) and is unlikely to scale up. The Markov blanket-based method uses a $G^2$ test and forward search investigating one locus at a time. Such as search would miss a loci-loci interaction that had no marginal effects.

### 2.4 Barriers to Learning Epistasis

As mentioned earlier, there are two central barriers to successfully identifying potential interactions in the etiology of disease using high-dimensional sets. First, we need to find an efficacious way to evaluate candidate models that can identify relationships like epistasis which exhibit little or no marginal effects. That is, even if we had the computational power to investigate all subsets of a

high-dimensional set, we would want to evaluate each subset using a method that has been shown to learn epistatic relationships well. Second, since we do not have the computational power to investigate all subsets, we need efficient algorithms that enable us to only investigate the most promising subsets. Not surprisingly, Evans et al. (2006) showed that as the marginal effects of two loci approach zero, the power to detect a two-loci interactions approaches zero unless we exhaustively investigate all two-loci interactions. None of the methods discussed above circumvent or solve this problem. Their evaluations were performed using high-dimensional data sets in which there were marginal effects.

As mentioned earlier, recent research by the primary author and colleagues addressing these barriers is discussed in Section 3 and Section 4. First, we provide further review.

### 2.5 MDR

*Multifactor dimensionality analysis* (*MDR*) (Hahn et al., 2003) is a well-known method for detecting epistasis. In Section 3 we include MDR in a comparison of the performance of two methods. So we review MDR using an example taken from (Velez et al., 2007).

**Example 2.** *Suppose we are investigating whether* $SNP_1$ *and* $SNP_2$ *are correlated with disease* $D$*. Suppose further that we obtain the data depicted in Figure 1 (a). The number of individuals with the disease appears on the left in each cell, whereas the number without it appears on the right. For example, of those individuals who have* $SNP_1 = 0$*,* 49 *have the disease and* 44 *do not have the disease. We see from Figure2 (a) that neither SNP by itself seems correlated with the disease.*

*Using MDR we investigate whether the SNPs together are correlated with the disease. First, we take the cross product of all values of the SNPs as shown on the left in Figure 1 (b). For each combination of values of the SNPs we determine how many individuals have the disease and how many do not, as also shown in that figure. Let* $\#D$ *be the number who have the disease and* $\#noD$ *be the number who do not have the disease. For a given SNP combination we call the combination high-risk (HR) if* $\frac{\#D}{\#noD} > T$*, where* $T$ *is a threshold. Ordinarily,* $T = 1$ *if in total we have the same number of individuals with the disease as without the disease. Using this threshold, the number of high-risk individual for each SNP combination is the number labeled HR on the left in Figure 1 (b).*

*Next we create a new binary variable* $SNP_1 \times SNP_2$ *whose value is* $HR$ *if the SNP combination is one of the high-risk SNPs and whose value is* $LR$ *(low-risk) if the SNP combination is one of the low-risk SNPs. We then compute the total number of individuals who have value* $HR$ *and have the disease, have value* $HR$ *and do not have the disease, have value* $LR$ *and have the disease, and have value* $LR$ *and do not have the disease. These totals appear on the right in Figure 1 (b). For example, consider those individuals who have value* $HR$ *and have the disease. We obtain the total as follows:*

$$46 + 49 + 46 + 59 = 200.$$

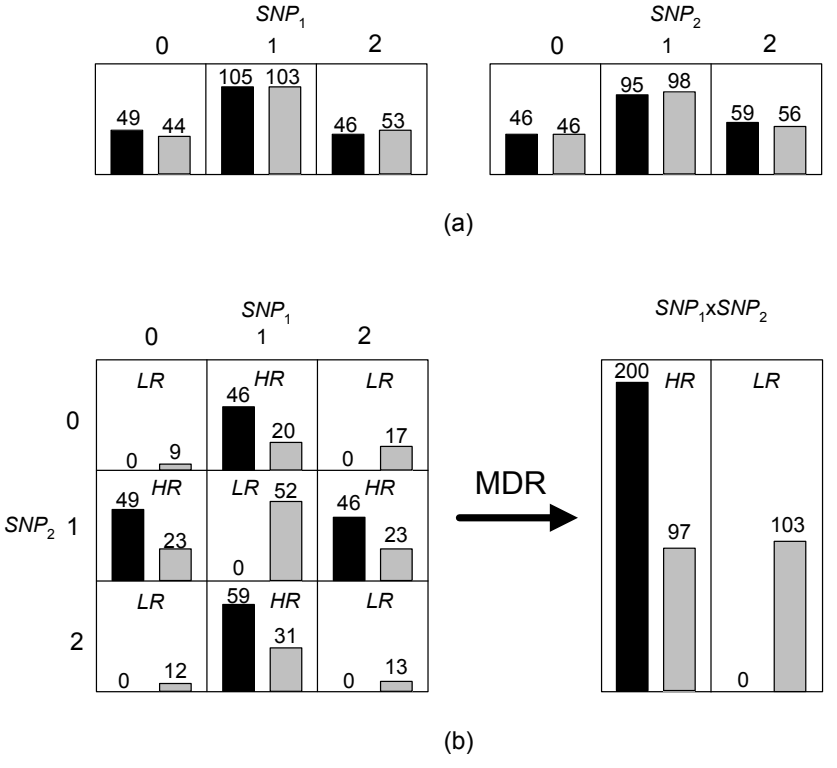



**Fig. 1.** The number of individuals with the disease appears on the left in each cell, whereas the number without it appears on the right. (a) shows the numbers for each value of each SNP individually. (b) shows the the number for the cross product of the values of the SNPs and the numbers for the binary-valued variable $SNP_1 \times SNP_2$ obtained using MDR. The values of this variable are $HR$ (high-risk) and $LR$ (low-risk).

*We see from the right part of Figure 1 (b) that the disease appears to be correlated with the cross product of the SNPs.*

We illustrated MDR for the case where we are investigating the correlation of a 2-SNP combination with a disease. Clearly, the method extends to three or more SNPs. If we are investigating $n$ SNPs and considering $k$-SNP combinations with a disease, we investigate all $\binom{n}{k}$ combinations and choose the combination that appears best according to some criterion. (Velez et al., 2007) use the following *classification error* as the criterion:

$$\frac{(\#\text{ individuals with HR and no disease}) + (\#\text{ individuals with LR and disease})}{\#\text{ individuals}}.$$

For example, for the SNPs illustrated in Figure 1 the classification error is

$$\frac{97+0}{400} = 0.2425.$$

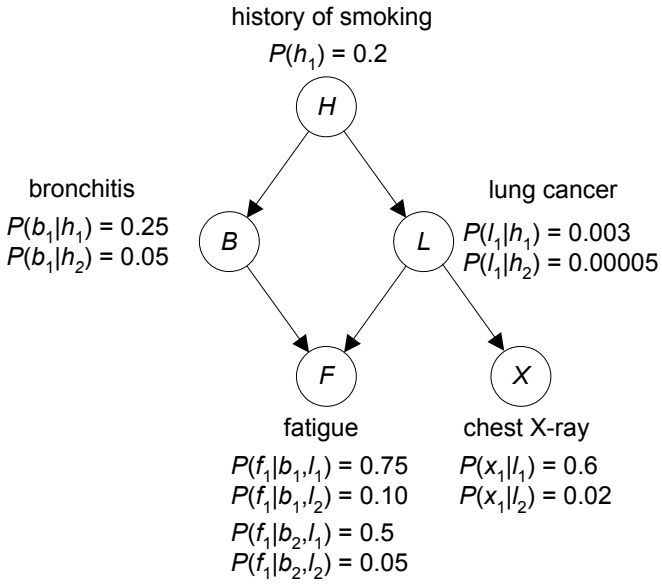



**Fig. 2.** A BN modeling lung disorders. This BN is intentionally simple to illustrate concepts; it is not intended to be clinically complete.

(Velez et al., 2007) determine a likely SNP combination using 10-fold cross validation. That is, they break the data set into 10 partitions, they use 9 partitions to learn a $k$-SNP combination (the one with the smallest classification error), and then using the learned SNP combination they determine the prediction error for the remaining partition. They repeat this procedure for all possible choices of the 9 learning partitions, and then they take the average. This process is repeated for all $1, 2, ..., k$ SNP combinations that are computationally feasible. A model is chosen for each number of SNPs. A final model is chosen from this set of models based on minimizing the average prediction error and maximizing cross-validation consistency.

If the number of individuals with the disease is not the same as the number of individuals without the disease, some adjustments need to be made. (Velez et al., 2007) discuss using *balanced accuracy* and an *adjusted threshold* to handle this situation.

### 2.6 Bayesian Networks

The epistasis discovery method presented in Sections 3 and 4 is based on Bayesian networks; so, we review them next.

Bayesian networks (Neapolitan, 2004; Koller and Friedman, 2009) are increasingly being used for modeling and knowledge discovery in many domains including bioinformatics (Neapolitan, 2009). A *Bayesian network* (*BN*) consists of a directed acyclic graph (DAG) $G$ whose nodes are random variables and the

conditional probability distribution of each node given its parents in $G$. Figure 2 shows a BN. In that BN $h_1$, for example, means an individual has a smoking history, whereas $h_2$ means the individual does not.

Using a Bayesian network inference algorithm we can compute the probability of nodes of interest based on the values of other nodes. For example, for the Bayesian network in Figure 2, we could compute the probability that a patient has bronchitis or lung cancer given that the patient has a smoking history and positive chest X-ray.

Methods have been developed for learning both the parameters in a BN and the structure (DAG) from data. The task of learning a unique DAG model from data is called *model selection*. In the *constraint-based approach* (Spirtes et al., 1993, 2000) to model selection, we try to learn a DAG from the conditional independencies that the data suggest are present in the generative probability distribution . In a *score-based approach* (Neapolitan, 2004), we assign a score to a DAG based on how well the DAG fits the data. A straightforward score, called the *Bayesian score*, is the probability of the given the DAG. This score for discrete random variables is as follows (Cooper and Herskovits, 1992):

$$\begin{aligned} score_{Bayes}(G : Data) &= P(Data|G) \\ &= \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\sum_{k=1}^{r_i} a_{ijk})}{\Gamma(\sum_{k=1}^{r_i} a_{ijk} + \sum_{k=1}^{r_i} s_{ijk})} \prod_{k=1}^{r_i} \frac{\Gamma(a_{ijk} + s_{ijk})}{\Gamma(a_{ijk})} \end{aligned} \quad (1)$$

where $r_i$ is the number of states of $X_i$, $q_i$ is the number of different instantiations of the parents of $X_i$, $a_{ijk}$ is the ascertained prior belief concerning the number of times $X_i$ took its $k$th value when the parents of $X_i$ had their $j$th instantiation, and $s_{ijk}$ is the number of times in the data that $X_i$ took its $k$th value when the parents of $X_i$ had their $j$th instantiation.

The Bayesian score assumes that our prior belief concerning each of the probability distributions in the network is represented by a Dirichlet distribution. When using the $Dir(\theta_X; a_1, a_2, \ldots, a_r)$ distribution to represent our belief concerning the unknown probability distribution $\theta_X$ of random variable $X$, due to cogent arguments such as the one in (Zabell, 1982), it has become standard to represent prior ignorance as to the value of $\theta_X$ by setting all parameter equal to 1. That is, $a_1 = a_2 = \cdots = a_r = 1$. The parameters $\{a_{ij1}, a_{ij2}, \ldots a_{ijr_i}\}$ in Equation 1 are Dirichlet parameters representing our belief about $\theta_{ij}$, which is the conditional probability distribution of $X_i$ given that the parents of $X$ are in their $j$th instantiation. To represent prior ignorance as to all conditional probabilities in the network, Cooper and Herskovits (1992) set $a_{ijk} = 1$ for all $i$, $j$, and $k$; they called this the K2 score.

Heckerman et al. (1995) noted a problem with setting all $a_{ijk}$ to 1, namely that equivalent DAGs could end up with different Bayesian scores. For example, the DAGs $X \rightarrow Y$ and $X \leftarrow Y$ could obtain different scores. Heckerman et al. (1995) proved that this does not happen if we use a *prior equivalent sample* size $\alpha$ in the DAG. When using a prior equivalent sample size we specify the same prior sample size $\alpha$ at each node. If we want to use a prior equivalent sample

size and represent a prior uniform distribution for each variable in the network, for all $i$, $j$, and $k$ we set

$$a_{ijk} = \frac{\alpha}{r_i q_i}.$$

When we determine the values of $a_{ijk}$ in this manner, we call the Bayesian score the *Bayesian Dirichlet equivalence uniform (BDeu)* score.

Another popular way of scoring is to use the *Minimum Description Length* (*MDL*) Principle (Rissanen, 1978), which is based on information theory and says that the best model of a collection of data is the one that minimizes the sum of the encoding lengths of the data and the model itself. To apply this principle to scoring DAGs, we must determine the number of bits needed to encode a DAG $G$ and the number of bits needed to encode the data given the DAG. Suzucki (1999) developed the following well-known MDL score:

$$score_{MDL}(G : Data) = \sum_{i=1}^{n} \frac{d_i}{2} \log_2 m - m \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} P(x_{ik}, pa_{ij}) \log_2 \frac{P(x_{ik}, pa_{ij})}{P(x_{ik})P(pa_{ij})}, \tag{2}$$

where $n$ is the number of nodes in $G$, $d_i$ is the number of parameters stored for the $i$th node in $G$, $m$ is the number of data items, $r_i$ is the number of states of $X_i$, $x_{ik}$ is the $k$th state of $X_i$, $q_i$ is the number of instantiations of the parents of $X_i$, $pa_{ij}$ is the $j$th instantiation of the parents of $X_i$ in $G$, and the probabilities are computed using the data. The first term is the number of bits required to encode the DAG model (called the *DAG penalty*), and the second term concerns the number of bits needed to encode the data given the model. Lam and Bacchus (1994) developed a similar MDL score.

Another score based on information theory is the *Minimum Message Length Score* (*MML*) (Korb and Nicholson, 2003).

If the number of variables is not small, the number of candidate DAGs is forbiddingly large. Furthermore, the BN structure learning problem has been shown to be NP-hard (Chickering, 1996). So heuristic algorithms have been developed to search over the space of DAGs during learning (Neapolitan, 2004). When the number of variables is large relative to the number of variables, many of the highest scoring DAGs can have similar scores (Heckerman, 1996). In this case approximate model averaging using MCMC may obtain better results than model selection (Hoeting et al., 1999).

## 3 Discovering Epistasis Using Bayesian Networks

First we describe a specialized Bayesian network model for representing epistatic interactions; then we discuss an MDL score tailored to this model; finally we provide experimental results evaluating the performance of this score.

### 3.1 A Bayesian Network Model for Epistatic Interactions

Consider all DAGs containing features and a phenotype $D$, where $D$ is a leaf. We are not representing the relationship between gene expression levels. Rather

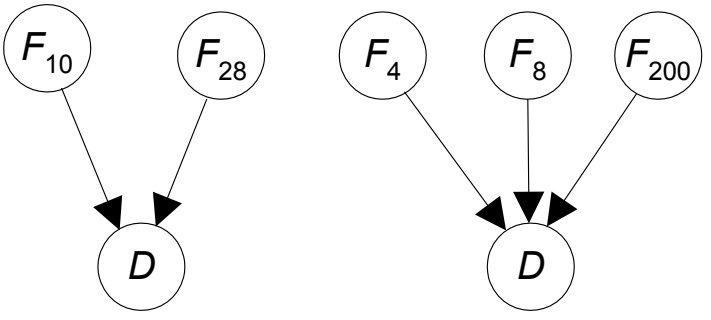



**Fig. 3.** Example DDAGs

we are representing the statistical dependence of the phenotype on alleles of the genes. So there is no need for edges between features, and we need only consider DAGs where the only edges are ones to $D$. Call such models *direct DAGs* (*DDAGs*). Figure 3 shows DDAGs. The size and complexity of the search space have been immensely reduced. Even so, there are still $2^n$ models where $n$ is the number of features. In practice a limit is put on the number of parents.

### 3.2 The BNMBL Score

Jiang et al. (2010a) developed an MDL score tailored to DDAGs. Each parameter in a DAG model is a fraction with precision $1/m$, where $m$ is the number of data items. So it takes $\log_2 m$ to store each parameter. However, as discussed in (Friedman and Yakhini, 1996), the high order bits are not very useful. So we can use only $\frac{1}{2}\log_2 m$ bits to store each parameter. In this way we arrive at the DAG penalty in Equation 2. Suppose now that $k$ SNPs have edges to $D$ in a given DDAG. That is, $D$ has $k$ parents. Since each SNP has three possible values, there are $3^k$ different values of these parents. The expected value of the number of data items that assume particular values of the parents is therefore $m/3^k$. If we approximate the precision for each parameter needed for $D$ by this expected value, our DAG penalty for a DDAG is

$$\frac{3^k}{2}\log_2\frac{m}{3^k} + \frac{2k}{2}\log_2 m. \tag{3}$$

When this DAG penalty is used in the MDL score, Jiang et al. (2010a) called the score the *Bayesian network minimum bit length* (*BNMBL*) score.

### 3.3 Experiments

Jiang et al. (2010a) compared the performance of BNMBL using both simulated and real data sets. Results obtained using each of the data sets are discussed next.

### 3.3.1 Simulated Data Sets

(Velez et al., 2007) developed a set of simulated data sets concerning epistatic models. These data sets, which are available at http://discovery.dartmouth.edu/epistatic_data/ VelezData.zip, were developed as follows. First, they created 70 different probabilistic relationships (models) in which 2 SNPs combined are correlated with the disease, but neither SNP is individually correlated. The relationships represented various degree of *penetrance, heritability*, and *minor allele frequency. Penetrance* is the probability that an individual will have the disease given that the individual has a genotype that is associated with the disease. *Heritability* is the proportion of the disease variation due to genetic factors. The *minor allele frequency* is the relative frequency of the less frequent allele in a locus associated with the disease. Supplementary Table 1 to (Velez et al., 2007) shows the details of the 70 models.

Data sets were then developed having a case-control ratio (ratio of individuals with the disease to those without the disease) of 1:1. To create one data set they fixed the model. Based on the model, they then developed data concerning the two SNPs that were related to the disease in the model, 18 other unrelated SNPs, and the disease. For each of the 70 models, 100 data sets were developed, making

**Table 1.** Columns MDR and BNMBL show the powers for MDR and BNMBL for models 55-59. The row labeled "Total" is the sum of the powers over the five models.

| Size of Data Set | Model | MDR | BNMBL |
|---|---|---|---|
| 200 | 55 | 3 | 7 |
| | 56 | 3 | 4 |
| | 57 | 3 | 5 |
| | 58 | 3 | 7 |
| | 59 | 3 | 3 |
| | **Total** (200) | **15** | **26** |
| 400 | 55 | 8 | 8 |
| | 56 | 7 | 9 |
| | 57 | 11 | 9 |
| | 58 | 15 | 27 |
| | 59 | 8 | 7 |
| | **Total** (400) | **49** | **60** |
| 800 | 55 | 26 | 30 |
| | 56 | 22 | 36 |
| | 57 | 25 | 29 |
| | 58 | 49 | 67 |
| | 59 | 18 | 24 |
| | **Total** (800) | **140** | **186** |
| 1600 | 55 | 66 | 81 |
| | 56 | 59 | 83 |
| | 57 | 68 | 81 |
| | 58 | 88 | 96 |
| | 59 | 49 | 63 |
| | **Total** (1600) | **330** | **404** |

a total of 7000 data sets. They followed this procedure for data set sizes of 200, 400, 800, and 1600. The task of a learning algorithm is to learn the 2-SNP model used to create each data set from that data set.

**Method.** The performances of MDR and BNMBL were compared using the simulated data sets just discussed. Jiang et al. (2010a) used MDR v. 1.25, which is available at www.epistasis.org, to run MDR, and developed their own implementation of BNMBL.

We say that a method correctly learns the model generating the data if it scores that model highest out of all $\binom{20}{2} = 190$ 2-SNP models. For a given model, let *power* to be the number of times the method correctly learned the model generating the data out of the 100 data sets generated for that model. The powers of MDR and BNMBL were compared using the data sets concerning the hardest-to-detect models and using all the data sets.

**Results.** Velez et al. (2007) showed that MDR has the lowest detection sensitivity for models 55-59 in Supplementary Table 1 to (Velez et al., 2007). These models have the weakest broad-sense heritability (0.01) and a minor allele frequency of 0.2. Table 1 shows the power for MDR and BNMBL for these 5 models. BNMBL outperformed MDR in 16 of the experiments involving the most difficult models, whereas MDR outperformed BNMBL only 2 times.

The first three columns of Table 2 shows the sums of the powers over all 70 models. The last two column shows $p$-values. These $p$-values were computed as follows. The Wilcoxon two-sample paired signed rank test was used to compare the powers of MDR and BNMBL over all 70 models. If we let the null hypothesis be that the medians of the power are equal, and the alternative hypothesis be that the median power of BNMBL is greater than that of MDR, then $p$ is the level at which we can reject the null hypothesis. We see from Table 2 that BNMBL significantly out-performed MDR for all data set sizes.

Looking again at Table 1, we see that when we have a relatively large amount of data (1600 data items), BNMBL correctly identified $404 - 330 = 74$ more difficult models than MDR. From Table 2 we see that when there are 1600 data items BNMBL correctly identified $6883 - 6792 = 91$ more total models than MDR. The majority of the improvement obtained by using BNMBL concerns the more difficult models. Arguably, real epistatic relationships are more often represented by such difficult models.

**Table 2.** The columns labeled MDR,and BNMBL show the sums of the powers over all 70 data sets for each of the methods. The other column shows p-values. See the text for their description.

| $n$ | **MDR** | **BNMBL** | $p$ |
|---|---|---|---|
| 200 | 4904 | 5016 | 0.009 |
| 400 | 5796 | 5909 | 0.004 |
| 800 | 6408 | 6517 | 0.003 |
| 1600 | 6792 | 6883 | 0.012 |

**Table 3.** Mean running times in seconds for MDR and BNMBL

| $n$ | **MDR** | **BNMBL** |
|---|---|---|
| 200 | 119.8 | 0.020 |
| 400 | 146.6 | 0.031 |
| 800 | 207.9 | 0.050 |
| 1600 | 241.7 | 0.097 |

Table 3 shows the mean running times in seconds obtained by averaging the running times over the data sets generated from all 70 genetic models. MDR is several orders of magnitude slower than BNMBL. The superior running time of BNMBL is due largely to its ability to use the entire data set for computing the score of each model, while MDR performs multi-fold cross-validation to score the models.

### 3.3.2 Real Data Set

It is well-known that the apoplipoprotein E (APOE) gene is associated with many cases of LOAD, which is characterized by dementia onset after age 60 (Coon et al., 2007; Papassotiropoulos et al., 2006; Corder et al., 1993). The APOE gene has three common variants $\varepsilon 2$, $\varepsilon 3$, and $\varepsilon 4$. The least risk is associated with the $\varepsilon 2$ allele, while each copy of the $\varepsilon 4$ allele increases risk.

Coon et al. (2007) performed a genome-wide association study, which investigated over 300,000 SNPs, using 1086 LOAD cases and controls to determine the odds ratio (OR) associated with genes relative to LOAD. Only SNP rs4420638 on chromosome 19, which is located 14 kilobase pairs distal to and in linkage disequilibrium with the APOE gene, significantly distinguished between LOAD cases and controls.

Reiman et al. (2007) investigated the association of these same SNPs separately in APOE $\varepsilon 4$ carriers and in APOE $\varepsilon 4$ noncarriers. A discovery cohort and two replication cohorts were used in the study. See (Reiman et al., 2007) for the

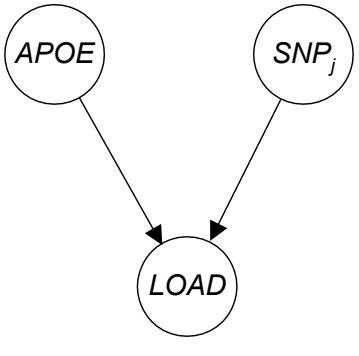



**Fig. 4.** A Bayesian networks in which the disease node (LOAD) has precisely two-parents, one being the APOE gene and the other being a SNP.

**Table 4.** The 28 highest scoring SNPs according to BNMBL. The column labeled GAB2 contains "yes" if the SNP is located on GAB2; the column labeled "Reiman" contains "yes" if the SNP is one of the 10 high scoring GAB2 SNPS discovered in (Reiman et al., 2007).

| | SNP | $Score_{BNMBL}$ | Chromosome | GAB2 | Reiman |
|---|---|---|---|---|---|
| 1 | rs2517509 | 0.13226 | 6 | | |
| 2 | rs1007837 | 0.13096 | 11 | yes | yes |
| 3 | rs12162084 | 0.13042 | 16 | | |
| 4 | rs7097398 | 0.13032 | 10 | | |
| 5 | rs901104 | 0.13019 | 11 | yes | yes |
| 6 | rs7115850 | 0.13018 | 11 | yes | yes |
| 7 | rs7817227 | 0.13009 | 8 | | |
| 8 | rs2122339 | 0.13002 | 4 | | |
| 9 | rs10793294 | 0.12997 | 11 | yes | yes |
| 10 | rs4291702 | 0.12992 | 11 | yes | yes |
| 11 | rs6784615 | 0.12986 | 3 | | |
| 12 | rs4945261 | 0.12963 | 11 | yes | yes |
| 13 | rs2373115 | 0.12956 | 11 | yes | yes |
| 14 | rs10754339 | 0.12932 | 1 | | |
| 15 | rs17126808 | 0.12932 | 8 | | |
| 16 | rs7581004 | 0.12929 | 2 | | |
| 17 | rs475093 | 0.12921 | 1 | | |
| 18 | rs2450130 | 0.12906 | 11 | yes | |
| 19 | rs898717 | 0.12888 | 10 | | |
| 20 | rs473367 | 0.12884 | 9 | | |
| 21 | rs8025054 | 0.12873 | 15 | | |
| 22 | rs2739771 | 0.12863 | 15 | | |
| 23 | rs826470 | 0.12862 | 5 | | |
| 24 | rs9645940 | 0.12853 | 13 | | |
| 25 | rs17330779 | 0.12847 | 7 | | |
| 26 | rs6833943 | 0.12830 | 4 | | |
| 27 | rs2510038 | 0.12824 | 11 | yes | yes |
| 28 | rs12472928 | 0.12818 | 2 | | |

details of the cohorts. Within the discovery subgroup consisting of APOE $\varepsilon 4$ carriers, 10 of the 25 SNPs exhibiting the greatest association with LOAD (contingency test $p$-value $9\times10^{-8}$ to $1\times10^{-7}$) were located in the GRB-associated binding protein 2 (GAB2) gene on chromosome 11q14.1. Associations with LOAD for 6 of these SNPs were confirmed in the two replication cohorts. Combined data from all three cohorts exhibited significant association between LOAD and all 10 GAB2 SNPs. These 10 SNPs were not significantly associated with LOAD in the APOE $\varepsilon 4$ noncarriers.

Reiman et al. (2007) also provided immunohistochemical validation for the relevance of GAB2 to the neuropathology of LOAD.

**Method.** Jiang et al. (2010a) investigated all 3-node Bayesian network models in which the disease node (LOAD) has precisely two-parents, one being the APOE gene and the other being one of the 312,260 SNPs investigated in (Reiman et al., 2007). Figure 4 shows one such a model. These models were scored with BNMBL using the combined data set consisting of all three cohorts described in (Reiman et al., 2007). The combined data set contains data on 1411 subjects. Of these subjects, 861 are LOAD cases, and 644 are APOE $\varepsilon 4$ carriers.

**Results.** Table 4 shows the 28 highest scoring SNPs. Since all DAG models have the same complexity, there is no need to include the term for DAG complexity in the score. So $Score_{BNMBL}$ consists only of term encoding the data without the minus sign, which means higher scores are better. We see that 7 of the top 13 SNPs were among the 10 SNPs discovered in (Reiman et al., 2007) and 9 of the top 27 SNPs are located in GAB2. The remaining high scoring SNPs are scattered among various chromosomes.

The results obtained using BNMBL substantiate the results in (Reiman et al., 2007), namely that GAB2 is associated with LOAD in APOE $\varepsilon 4$ carriers. This outcome demonstrates that BNMBL is a promising tool for learning real epistatic interactions.

An advantage of using BNMBL for knowledge discovery in this domain is that there is no need analyze the statistical relevance of a SNP separately under different conditions (e.g. first in all subjects, then in $\varepsilon 4$ carriers, and finally in $\varepsilon 4$ noncarriers). Rather we just score all relevant models using BNMBL.

## 4 Efficient Search

The second barrier to learning epistatic relationships from high-dimensional data sets is that we do not have the computational power to investigate very many subsets of the loci. So we need efficient algorithms that enable us to only investigate promising subsets.

*Greedy Equivalent Search* (*GES*) (Chickering, 2003) is an efficient Bayesian network learning algorithm that will learn the most concise DAG representing a probability distribution under the assumptions that the scoring criterion is consistent and that the probability distribution admits a faithful DAG representation and satisfies the composition property. See (Neapolitan, 2004) for a complete discussion of these assumptions and the algorithm. Briefly, the algorithm starts with the empty DAG and in sequence greedily adds the edge to the DAG that increases the score the most until no edge increases the score. Then in sequence it greedily deletes the edge from the DAG until no edge decreases the score. It is not hard to see that if there are $n$ variables, the worst-case time complexity of the algorithm is $\theta(n^2)$.

An initial strategy might be to try to learn the interacting SNPs by using the GES algorithm to search all DDAGs. However, a moment's reflection reveals that this could not in general work. Suppose we have an epistatic interaction between two SNPs and $D$ such that each SNP is marginally independent of $D$

and all other SNPs are also independent of $D$. Suppose further that we have a data set so large that the generative distribution is represented exactly in the data set. In this case the GES algorithm would learn the correct DAG if its assumptions were met. However, in the first step of the algorithm all SNPs will score the same because they are all independent of $D$, none of them will increase the score and the algorithm will halt.

The problem is the epistatic interactions do not satisfy the composition property which is necessary to the GES algorithm. Jiang et al. (2010b) ameliorated this problem by initially expanding each of the SNPs using greedy search rather than initially starting only with the one that increases the score the most. In this way, we will definitely investigate every 2-SNP combination. If an epistatic interaction is occurring, two of the SNPs involved in the interaction may score high. Once we identify these two, often we should also often find possible 3rd and 4th and so on SNPs involved in the interaction. The algorithm follows. In this algorithm by $score(A_i)$ we mean the score of the model that has edges from the SNPs in $A_i$ to $D$.

```
for each SNP SNP_i
    A_i = {SNP_i};
    do
        if adding any SNP to A_i increases score(A_i)
            add the SNP to A_i that increases score(A_i) the most;
    while adding some SNP to A_i increases score(A_i);
    do
        if deleting any SNP from A_i increases score(A_i)
            delete the SNP from A_i that increases score(A_i) the most;
    while deleting some SNP from A_i increases score(A_i);
endfor;
report k highest scoring sets A_i.
```

We call this algorithm *Multiple Beam Search* (*MBS*). It clearly requires $\theta(n^3)$ time in the worst case where $n$ is the number of SNPs. However, in practice if the data set is large, we would add at most $m$ SNPs in the first step, where $m$ is a parameter. So the time complexity would be $\theta(mn^2)$.

This technique would not work if there is a dependence between $k$ SNPs and $D$, but every proper subset of the $k$ SNPs is marginally independent of $D$. The MBS algorithm is effective for handling the situation in which we have $k$ SNPs interacting, each of them is marginally independent of the disease, and there is a dependence between the disease and at least one pair of the interacting SNPs. A reasonable conjecture is that many but certainly not all epistatic interactions satisfy this condition.

### 4.1 Experiments

Jiang el al. (2010b, 2010c) evaluated MBS using a simulated data set and two real GWAS data sets. The results are discussed next.

**Table 5.** Number of times the correct model scored highest out of 7000 data sets for MBS and Baycom

| Size of Data Set | MBS | BayCom |
|---|---|---|
| 200 | 4049 | 4049 |
| 400 | 5111 | 5111 |
| 800 | 5881 | 5881 |
| 1600 | 6463 | 6463 |

**Table 6.** Comparisons of average values of detection measures over 7000 data sets for MBS and Baycom

| | Spatial Recall | | Precision | | Overlap Coefficient | |
|---|---|---|---|---|---|---|
| Data Set Size | MBS | BayCom | MBS | BayCom | MBS | BayCom |
| 200 | 0.593 | 0.593 | 0.607 | 0.607 | 0.593 | 0.593 |
| 400 | 0.737 | 0.737 | 0.744 | 0.744 | 0.737 | 0.737 |
| 800 | 0.843 | 0.843 | 0.846 | 0.846 | 0.843 | 0.843 |
| 1600 | 0.925 | 0.925 | 0.926 | 0.926 | 0.925 | 0.025 |

#### 4.1.1 Simulated Data Sets

The simulated data sets developed in (Velez et al., 2007) (discussed in Section 3.3) were used in this evaluation which is taken from (Jiang et al., 2010b).

**Method.** The simulated data were analyzed using the following methods: 1) a Bayesian network combinatorial method, called *BayCom*, and which scores all 1-SNP, 2-SNP, 3-SNP, and 4-SNP DDAGs; 2) MBS with a maximum of $m = 4$ SNPs added in the first step. Candidate models were scored with the MML score mentioned in Section 2.6. This score has previously been used successfully in causal discovery (Korb and Nicholson, 2007).

**Results.** Table 5 shows the number of times the correct model scored highest over all 7000 data sets. Important detection measures include recall, precision, and the overlap coefficient. In the current context they are as follows. Let $S$ be the set of SNPs in the correct model and $T$ be the set of SNPs in the highest scoring model. Then (# returns the number of items in a set)

$$recall = \frac{\#(S \cap T)}{\#(S)},$$

$$precision = \frac{\#(S \cap T)}{\#(T)},$$

$$overlapcoefficient = \frac{\#(S \cap T)}{\#(SUT)}.$$

Table 6 shows the average values of these measures over all 7000 data sets. MBS performed as well as BayCom in terms of accuracy and the other measures. Table 7 shows the running times. MBS was up to 28 times faster than BayCom.

**Table 7.** Average running times in seconds over 7000 data sets

| Data Set Size | MBS | BayCom |
|---|---|---|
| 200 | 0.108 | 2.0 |
| 400 | 0.191 | 5.15 |
| 800 | 0.361 | 9.61 |
| 1600 | 0.629 | 18.0 |

**Table 8.** Occurrences of GAB2 and rs6094514 in high-scoring models when using MBS to analyze the LOAD data set

| # models in top 10 containing a GAB2 SNP | # models in top 100 containing a GAB2 SNP | # rs6094514 occurrences with GAB2 in top 10 | # rs6094514 occurrences with GAB2 in top 100 |
|---|---|---|---|
| 6 | 36 | 6 | 33 |

#### 4.1.2 Real Data Set

The real data set analyzed using MBS was the LOAD data set introduced in (Reiman et al., 2007) (See Section 3.3).

**Method.** (Jiang et al, 2010b) analyzed the LOAD data set as follows. Using all 1411 cases, they pre-processed the data by scoring all DDAG models in which APOE and one of the 312,316 SNPs are each parents of LOAD (See Figure 4). They then selected the SNPs from the highest-scoring 1000 models. Next MBS was run using the data set consisting of APOE and these 1000 SNPs. They did not constrain APOE to be in the discovered models. At most $m = 3$ nodes were added in the first step of MBS. There were $4.175 \times 1010$ models under consideration. Of course MBS scored far fewer models.

**Results.** The 1000 highest scoring models encountered in the MBS search were recorded. APOE appeared in every one of these models, and a GAB2 SNP appeared in the top two models. Columns one and two in Table 8 show the number of times a GAB2 SNP appeared respectively in the top 10 models and top 100 models. Of the 312,316 SNPs in the study, 16 are GAB2 SNPs. Seven of these 16 SNPs appeared in at least one of the 36 high-scoring models containing a GAB2 SNP.

All of these seven SNPs were among the 10 GAB2 SNPs identified in (Reiman et al., 2007). The probability of 36 or more of the top 100 models containing at least one of the 16 GAB2 SNPs by chance is $2.0806 \times 10^{-106}$. GAB2 SNPs never occurred together in a model. This pattern is plausible since each GAB2 SNP may represent the dependence between LOAD and GAB2, and therefore it could render LOAD independent of the other GAB2 SNPs. These results substantiate those in (Reiman et al., 2007), that GAB2 has an affect on LOAD. The results do not indicate whether GAB2 influences LOAD by interacting with APOE since APOE appears in every high-scoring model.

The run time was 4.1 hours. When 1, 2, and 3 SNP combinations involving only 200 SNPs in the LOAD data set were analyzed, the run time for BayCom was 1.04 hours. An extrapolation of this result indicates that it would take about 3.71 years to analyze all 1, 2, 3, and 4 combinations involving 1001 loci (1000 SNPs plus APOE).

An unexpected result was obtained. SNP rs6094514, which is an intron on the EYA2 gene on chromosome 20, often appeared along with GAB2 and LOAD. The third and fourth columns in Table 4 show the numbers of such occurrences respectively in the top 10 and top 100 models. Among the top 100 models, SNP rs6094514 only occurred once without GAB2. As it turns out, prior research has associated this SNP with LOAD. In a cross-platform comparison of outputs from four GWAS, Shi et al. (2010) found SNP rs6094515 to be associated with LOAD with a combined $p$-value of $8.54 \times 10^{-6}$. However, no prior literature shows that GAB2 and EYA2 may interact to affect LOAD, as the current results seem to suggest. MBS discovered this possibility because it is able to tractably investigate multi-loci interactions.

Another result was that SNP rs473367 on chromosome 9 appeared in the 3rd and 4th models and in 22 of the top 99 models. It never appeared with GAB2. A previous study (WO/2008/131364) suggested that this SNP interacts with APOE to affect LOAD. The results discussed here support this association, but indicate no interaction with GAB2.

## 5 Discussion, Limitations, and Future Research

We showed a Bayesian network model for representing epistatic interactions called a DDAG and an MDL score called BNMBL designed specifically for this model. Using simulated data sets, BNMBL performed significantly better than MDR at identifying the two SNPs involved in an epistatic interaction. The BNMBL score performed well at identifying potential epistatic interactions from a real GWAS data set, as did the MML score.

We showed an algorithm called MBS that successfully identified potential epistatic interactions using a real GWAS data set. This algorithm requires quadratic time in terms of the number of SNPs from which we initiate beams (used as starting points for greedy search). If we initiated beams from all 500,000 SNPs in a given GWAS, quadratic time could take months. So in the study shown the data was pre-processed to identify the 1000 highest scoring individual SNPs from 2-parent models containing APOE and one of the 312,260 SNPs. Beams were then initiated from these 1000 SNPs and from APOE. In general, we would not suspect a gene such as APOE to be involved in the interaction. So our preprocessing would only involve scoring all 1-SNP models and choosing the 1000 highest scoring SNPs. If SNPs involved in an epistatic interaction exhibited absolutely no marginal effects, there is no reason they should appear in the top 1000 SNPs. So such an interaction would probably be missed. MBS has made progress in identifying epistasis when we either have a great deal of computational power or when at least one SNP shows a slight marginal effect. However,

further research is needed to investigate handling the situation where there are no marginal effects more efficiently.

After discovering candidate loci-phenotype relationships, researchers often report their significance using the Bonferroni correction or the False Discovery rate. However, some Bayesian statisticians (see e.g. (Neapolitan, 2008)) have argued that it is not reasonable to using these methods or any other method based on the number of hypotheses investigated, particularly in this type of domain. A simple example illustrating their argument is as follows: Suppose that one study investigates 100,000 SNPs while another investigates 500,000 SNPs. Suppose further that the data concerning a particular SNP and the disease is identical in the two studies. Due to the different corrections, that SNP could be reported as significant in one study but not the other. Yet the data concerning the SNP is identical in the two studies! It seems the only reason these corrections work at all is because they serve as surrogates for low prior probabilities. However, as the previous example illustrates, they can be very poor surrogates. It would be more consistent to ascertain prior probabilities that can be used uniformly across studies. Future research should investigate ascertaining prior probabilities in this domain, and reporting results using posterior probabilities rather than significance with a correction.

## References

Bateson, W.: Mendel's Principles of Heredity. Cambridge University Press, New York (1909)

Brooks, A.J.: The Essence of SNPs. Gene. 234, 177–186 (1999)

Chen, S.S., et al.: Atomic Decomposition by Basis Pursuit. SIAM Journal on Scientific Computing 20, 33–61 (1998)

Chickering, M.: Learning Bayesian Networks is NP-Complete. In: Fisher, D., Lenz, H. (eds.) Learning from Data. Lecture Notes in Statistics, Springer, New York (1996)

Chickering, D.: Optimal Structure Identification with Greedy Search. The Journal of Machine Learning Research 3, 507–554 (2003)

Cho, Y.M., Ritchie, M.D., Moore, J.H., Moon, M.K., et al.: Multifactor Dimensionality Reduction Reveals a Two-Locus Interaction Associated with Type 2 Diabetes Mellitus. Diabetologia 47, 549–554 (2004)

Coffey, C.S., et al.: An Application of Conditional Logistic Regression and Multifactor Dimensionality Reduction for Detecting Gene-Gene Interactions on Risk of Myocardial Infarction: the Importance of Model Validation. BMC Bioinformatics 5(49) (2004)

Coon, K.D., et al.: A High-Density Whole-Genome Association Study Reveals that APOE is the Major Susceptibility Gene for Sporadic Late-Onset Alzheimer's Disease. J. Clin. Psychiatry 68, 613–618 (2007)

Cooper, G.F., Herskovits, E.: A Bayesian Method for the Induction of Probabilistic Networks from Data. Machine Learning 9, 309–347 (1992)

Corder, E.H., et al.: Gene Dose of Apolipoprotein E type 4 Allele and the Risk of Alzheimer's Disease in Late Onset Families. Science 261, 921–923 (1993)

Epstein, M.J., Haake, P.: Very Large Scale ReliefF for Genome-Wide Association Analysis. In: Proceedings of IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (2008)

Evans, D.M., Marchini, J., Morris, A., Cardon, L.R.: Two-Stage Two-Locus Models in Genome-Wide Association. PLOS Genetics 2(9) (2006)

Friedman, N., Yakhini, Z.: On the Sample Complexity of Learning Bayesian Networks. In: Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence, pp. 206–215 (1996)

Galvin, A., Ioannidis, J.P.A., Dragani, T.A.: Beyond Genome-Wide Association Studies: Genetic Heterogeneity and Individual Predisposition to Cancer. Trends in Genetics (3), 132–141 (2010)

Hahn, L.W., Ritchie, M.D., Moore, J.H.: Multifactor Dimensionality Reduction Software for Detecting Gene-Gene and Gene-Environment Interactions. Bioinformatics 19(3), 376–382 (2003)

Han, B., Park, M., Chen, X.: Markov Blanket-Based Method for Detecting Causal SNPs in GWAS. In: Proceeding of IEEE International Conference on Bioinformatics and Biomedicine (2009)

Heckerman, D.: A Tutorial on Learning with Bayesian Networks, Technical Report # MSR-TR-95-06. Microsoft Research, Redmond, WA (1996)

Heckerman, D., Geiger, D., Chickering, D.: Learning Bayesian Networks: The Combination of Knowledge and Statistical Data, Technical Report MSR-TR-94-09. Microsoft Research, Redmond, Washington (1995)

Hoeting, J.A., Madigan, D., Raftery, A.E., Volinksy, C.T.: Bayesian Model Averaging: A Tutorial. Statistical Science 14, 382–417 (1999)

Hunter, D.J., Kraft, P., Jacobs, K.B., et al.: A Genome-Wide Association Study Identifies Alleles in FGFR2 Associated With Risk of Sporadic Postmenopausal Breast Cancer. Nature Genetics 39, 870–874 (2007)

Jiang, X., Barmada, M.M., Visweswaran, S.: Identifying Genetic Interactions From Genome-Wide Data Using Bayesian Networks. Genetic Epidemiology 34(6), 575–581 (2010a)

Jiang, X., Neapolitan, R.E., Barmada, M.M., Visweswaran, S., Cooper, G.F. : A Fast Algorithm for Learning Epistatic Genomic Relationships. In: Accepted as Proceedings Eligible by AMIA 2010 (2010b)

Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIT Press, Cambridge (2009)

Korb, K., Nicholson, A.E.: Bayesian Artificial Intelligence. Chapman & Hall/CRC, Boca Raton, FL (2003)

Lam, W., Bacchus, F.: Learning Bayesian Belief Networks: An approach based on the MDL Principle. In: Proceedings of 2nd Pacific Rim International Conference on Artificial Intelligence, pp. 1237–1243 (1992)

Logsdon, B.A., Hoffman, G.E., Mezey, J.G.: A Variational Bayes Algorithm for Fast and Accurate Multiple Locus Genome-Wide Association Analysis. BMC Bioinformatics 11(58) (2010)

Manolio, T.A., Collins, F.S.: The HapMap and Genome-Wide Association Studies in Diagnosis and Therapy. Annual Review of Medicine 60, 443–456 (2009)

Matsuzaki, H., Dong, S., Loi, H., et al.: Genotyping over 100,000 SNPs On a Pair of Oligonucleotide Arrays. Nat. Methods 1, 109–111 (2004)

Meng, Y., et al.: Two-Stage Approach for Identifying Single-Nucleotide Polymorphisms Associated With Rheumatoid Arthritis Using Random Forests and Bayesian Networks. BMC Proc. 2007 1(suppl. 1), S56 (2007)

Moore, J.H., White, B.C.: Tuning reliefF for genome-wide genetic analysis. In: Marchiori, E., Moore, J.H., Rajapakse, J.C. (eds.) EvoBIO 2007. LNCS, vol. 4447, pp. 166–175. Springer, Heidelberg (2007)

Neapolitan, R.E.: Learning Bayesian Networks. Prentice Hall, Upper Saddle River (2004)
Neapolitan, R.E.: A Polemic for Bayesian Statistics. In: Holmes, D., Jain, L. (eds.) Innovations in Bayesian Networks. Springer, Heidelberg (2008)
Neapolitan, R.E.: Probabilistic Methods for Bioinformatics: with an Introduction to Bayesian Networks. Morgan Kaufmann, Burlington (2009)
Pappassotiropoulos, A., Fountoulakis, M., Dunckley, T., Stephan, D.A., Reiman, E.M.: Genetic Transcriptomics and Proteomics of Alzheimer's Disease. J. Clin. Psychiatry 67, 652–670 (2006)
Reiman, E.M., et al.: GAB2 Alleles Modify Alzheimer's Risk in APOE $\epsilon 4$ Carriers. Neuron 54, 713–720 (2007)
Ritchie, M.D., et al.: Multifactor-Dimensionality Reduction Reveals High-Order Interactions among Estrogen-Metabolism Genes in Sporadic Breast Cancer. Am. J. Hum. Genet. 69(1), 138–147 (2001)
Rissanen, J.: Modelling by Shortest Data Description. Automatica 14, 465–471 (1978)
Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction, and Search. Springer, New York (1993); 2nd edn. MIT Press (2000)
Suzuki, J.: Learning Bayesian Belief Networks based on the Minimum Description length Principle: Basic Properties. IEICE Trans. on Fundamentals E82-A(9), 2237–2245 (1999)
Tibshirani, R.: Regression Shrinkage and Selection Via the Lasso. J. Royal. Statist. Soc. B 58(1), 267–288 (1996)
Velez, D.R., White, B.C., Motsinger, A.A., Bush, W.S., Ritchie, M.D., Williams, S.M., Moore, J.H.: A Balanced Accuracy Function for Epistasis Modeling in Imbalanced Dataset using Multifactor Dimensionality Reduction. Genetic Epidemiology 31, 306–315 (2007)
Verzilli, C.J., Stallard, N., Whittaker, J.C.: Bayesian Graphical Models for Genomewide Association Studies. The American Journal of Human Genetics 79, 100–112 (2006)
Wade, N.: A Decade Later, Genetic Map Yields Few New Cures. New York Times (June 12, 2010)
Wan, X., et al.: Predictive Rule Inference for Epistatic Interaction Detection in Genome-Wide Association Studies. Bioinformatics 26(1), 30–37 (2010)
Wang, D.G., Fan, J.B., Siao, C.J., et al.: Large-Scale Identification, Mapping, and Genotyping of Single Nucleotide Polymorphisms in the Human Genome. Science 80, 1077–1082 (1998)
Wu, T.T., Chen, Y.F., Hastie, T., Sobel, E., Lange, K.: Genome-Wide Association Analysis by Lasso Penalized Logistic Regression. Genome Analysis 25, 714–721 (2009)
Wu, J., Devlin, B., Ringguist, S., Trucco, M., Roeder, K.: Screen and Clean: A Tool for Identifying Interactions in Genome-Wide Association Studies. Genetic Epidemiology 34, 275–285 (2010)
Zabell, S.L.: W.E. Johnson's 'Sufficientness' Postulate. The Annals of Statistics 10(4) (1982)
Zhang, X., Pan, F., Xie, Y., Zou, F., Wang, W.: COE: A general approach for efficient genome-wide two-locus epistasis test in disease association study. In: Batzoglou, S. (ed.) RECOMB 2009. LNCS, vol. 5541, pp. 253–269. Springer, Heidelberg (2009)
Zhang, Y., Liu, J.S.: Bayesian Inference of Epistatic Interactions in Case Control Studies. Nature Genetics 39, 1167–1173 (2007)

# Chapter 10
# Knowledge Discovery in Adversarial Settings

D.B. Skillicorn

School of Computing Queen's University

**Abstract.** In adversarial settings, the interests of modellers and those being modelled do not run together. This includes domains such as law enforcement and counterterrorism but, increasingly, also more mainstream domains such as customer relationship management. The conventional strategy, maximizing the fit of a model to the available data, does not work in adversarial settings because the data cannot all be trusted, and because it makes the results too predictable to adversaries. Some existing techniques remain applicable, others can be used if they are modified to allow for the adversarial setting, while others must be discarded. General principles for this domain are discussed and the implications for practice outlined.

## 1 Introduction

Adversarial settings are those in which the interests of those who are analyzing the data, and those whose data is being analyzed are not aligned. Some settings with this property are obvious: law enforcement, where the analysis aims to detect criminals and their actions; fraud detection, where the analysis aims to deter fraud and prosecute it when discovered, especially in areas such as insurance and tax evasion; counterterrorism and counterintelligence, where the analysis aims to deter, detect, and pursue those who want to disrupt a country; and financial tracking, where the analysis aims to discover money laundering. In these settings, it is obvious why some of those whose data are being analyzed are motivated to disrupt the analysis process by whatever means they can.

There are other areas, however, that are also adversarial, but not so obviously. For example, many businesses use customer relationship management techniques to try and predict their 'good' customers, now and in the future – customers with the largest net future value. These customers are those who will make the largest profit for the business over the long haul. Losing such customers to a competitor is costly, so it is worth spending some money up front to acquire and retain them, in the hope that this cost will be repaid over the course of the customer-business lifetime. A customer who is categorized as profitable over the long term has an opportunity to receive some immediate benefit. Therefore it is in the interests of *every* customer to appear to be one of these long-term profitable customers; but in the interests of the business to keep this category as small and accurate as possible. The interests of the modellers and those being modelled are quite different, and so this is an adversarial setting.

D.E. Holmes, L.C. Jain (Eds.): Data Mining: Found. & Intell. Paradigms, ISRL 25, pp. 211–223.
springerlink.com 

Another area where the interests of customers and businesses tend to be in opposition is privacy. Customers are forced to share some of their details with a business, for example their credit card details and their address for delivery. This may be unavoidable (although new technologies can help), but customers then expect the business to keep the information private. However, businesses see such data, correctly, as enormously valuable, both to themselves but also to other businesses. In other words, data on customers begins to look like an asset that can be sold in times of difficulty, or leaked through incompetence. Privacy is therefore another area where customer incentives and business incentives are not aligned.

There are many situations that are win-win for both the modeller and those whose data is being modelled. But there are also many situations, perhaps the majority, where the win is all or mostly on the side of the modeller. This changes the modelling problem – the modeller cannot simply use standard techniques and expect them to perform well because those who will lose because of the modelling will do something about it.

How can adversaries disrupt the modelling process? The easiest way is to suborn some individual inside the organization that is doing the modelling to alter the data, alter the algorithms, or alter the results. However, this is of little interest from a knowledge-discovery perspective. Within the knowledge-discovery process itself, the only way in which adversaries can affect the modelling is to alter the data. However, it is useful to distinguish different ways of altering the data depending on what aspect of the modelling the alterations are targeting.

Adversaries may try to prevent their data being collected at all. How easy this is to do depends on the exact data-collection mechanism: an image cannot be collected of someone who doesn't walk near the CCTV camera. However, it is becoming increasingly difficult to avoid data collection by non-participation, so the second manipulation option is to allow the data to be collected but to make sure that the associated identifying key is either unusable or belongs to someone else. For example, creating an email address and using it only to send a single message means that, even if the message is intercepted, the information about the sender is (almost) useless. Single-use credit card numbers are also now available for those who want to buy something without the purchase being connected to other purchases by the same person. If an unusable key field is not possible, an individual may use the key field belonging to someone else, one reason why identity theft is popular. Another possibility is to corrupt the key field, preferably in a way that can be disavowed if it is noticed. For example, criminals often use permutations of their names because it muddles the search for their previous records; and humans are prone to reversing digits in long numbers, so this can always be done deliberately without consequence.

At a more sophisticated level, adversaries may alter the records being collected to make themselves appear different (better) than they are in the resulting models. For example, in many criminal settings, adversaries try to create records that appear more normal or innocuous than they otherwise would so that these records will not appear as outliers in a clustering.

The knowledge required to do this is of two qualitatively different kinds: what do innocuous records look like, and what kind of modelling analysis is being done with the data. Knowing the first is often straightforward, but not always: what is the average number of ATM withdrawals made in a month, what is the average size of a particular deduction on a tax return, what is the average number of phone calls made each week? Those behaving normally never think about whether their behavior is actually normal, but those who are trying to *seem* normal have to consider this issue; and it may be quite difficult to assess what innocuous values are. The other risk of trying to look as normal as possible is that it may create records that are excessively bland, because every attribute is as normal as possible. In other words, a normal record may not be quite the same thing as a record all of whose attributes are normal. This pattern is often noticeable if modellers think to look for it.

The second, knowing what kind of modelling is being done, is more difficult for an adversary; but, in many contexts, it is possible for an adversary to construct plausible data and consider how their own data might look in relation to it. This might require some experimentation with a variety of techniques. Adversaries have one big advantage: they know what their own data looks like, and can generate as much of it as they want, whereas adversary records are typically rare for the data modeller.

Because adversarial settings are a kind of arms race, models may be updated fairly frequently to take account of the reaction of adversaries to any successful detection. However, this creates another channel for adversaries to manipulate the modelling process, by creating records that are likely to be used to train an updated model. Although they cannot, of course, force their records to be used, they can create records that appear to be unusually interesting from the perspective of model update: very close to the current class boundaries, all to one side of an existing cluster, and so on. If such records are selected for model update, they distort the revised model and, worse still, distort it in ways that the adversaries understand. This extra knowledge allows them to exploit the updated model more effectively than its predecessor. For example, if a particular feature is treated by the current model as strongly predictive of adversarial activity, adversaries can create new records that have the feature but are innocuous. If the adversaries are fortunate, the modellers may cease to believe in this aspect of the model, or they may select some of the apparently misclassified records for retraining.

At an even more sophisticated level, adversaries may create records that are designed to subvert the actions and decisions that might be made as a result of the modelling process – social engineering. As a simple example, some airline bookings sites allow users to select their own honorific – choosing to call oneself "Doctor" or "The Honorable" tends to improve the flying experience.

Standard approaches to model building all rely on minimizing the error between a model and the data from which it was built (or, equivalently, maximizing the fit). This is not a workable strategy for model building in adversarial settings, because it is too predictable. Even if an adversary does not know precisely what kind of model is being built, knowing that the global strategy is error

minimization, it is still be possible for an adversary to estimate the likelihood of being detected, and to craft records that are likely to subvert the modelling process. As a very simple example, minimizing the least-squares error is a common and natural way to fit a surface to a set of data points; moreover, it is statistically principled. However, the actual surface can be altered substantially by a single data record that is far from the 'natural' surface since this distance contributes quadratically to the apparent error. Hence distorting the model is easy (in the sense that it requires forcing only a single record into the training data) and the effect of the distortion is known to the adversary, given the chosen record. This simple example illustrates the modeller's dilemma – using standard quality measures for models makes the process too predictable.

On the other hand, modellers do have some advantages in adversarial settings. Adversaries are taking action to avoid being modelled well, and to manipulate the process. Looking explicitly for signals associated with avoidance and manipulation may make adversaries more visible in a model than their inherent differences would have. Also, adversaries in many situations are acting in ways that are not socially sanctioned. Their awareness of this can generate negative emotions that sometimes generate signals that may be used to detect them.

## 2 Characteristics of Adversarial Modelling

We have suggested that, in adversarial settings, new approaches to knowledge discovery will be needed for effective model building, and to avoid being misled by adversarial manipulation. We now consider some of the characteristics of the data that will constrain how to build workable models.

**Adversary records will be rare, and will not be very different from other records.** At least for the present, the number of records that represent adversarial activity will usually be a small fraction of the whole dataset. This immediately creates problems for discovering boundaries or for clustering, since there are few samples from one of the classes, the class associated with adversary action. In other words, adversarial knowledge discovery is often best regarded as a one-class problem.

Furthermore, the differences between records of adversarial activity and other records will be small *because adversaries will try hard to make sure that they are*. This differentiates adversarial knowledge discovery from research areas such as outlier detection, which assume that, although the outlying records are rare, they are also very different from ordinary or normal records. Adversarial records *are* pulled away from more 'normal' records because the activities of adversaries are unusual; but adversaries will try to make the difference between these records and more normal records as small as possible.

Implication: adversarial knowledge discovery is not much like outlier or anomaly detection.

On the other hand, it can be difficult for adversaries to guess the normal range and distribution of values for an attribute, especially if it is one with which they do not have experience. Those who do an activity without regard for the fact that

data might be collected about it, just do it. Those who think about whether they are doing it 'normally' may find themselves unable to tell, and may overthink, creating values that are either extreme or, for multiple attributes, too bland. For example, suppose that a group of conspirators are communicating using telephone calls. How many calls among them would be in the normal range? What times of day would be unusual and potentially suspicious? The more such issues are thought about, the less 'natural' the eventual values become. A well-known example occurs when humans make up numbers – the digit distribution is detectably different from actual numbers from real-world activities, a property known as Benford's Law [22]. This has proven useful in looking for tax evasion and financial fraud. If it is difficult to guess the values of individual attributes, it is even harder to guess values that will make the correlations among some subset of attributes have natural values. For example, a fictitious deduction on a tax return must have a plausible numerical value – but must also have a plausible magnitude in relation to, say, total income.

Implication: the attempt to make adversarial records seem normal can backfire, creating detectable differences that are either too unusual, too bland, or correlated in unusual ways.

Implication: collecting unusual or unexpected attributes, or attributes whose values are hard to alter, may make adversarial records easier to detect because they preserve the unusual nature of adversarial activity better.

**Model building must be inductive, not pattern-driven.** Because adversarial records are rare, and not very different from other records, the problem of detecting adversarial activity cannot be framed as a two-class prediction problem: innocuous and adversarial. Even if the general form of adversarial activity is known, say terrorism, there are usually a very large number of potential targets, methods, timings, and participants; and there is unlikely to be a known *modus operandi* because of the desire to surprise, to evade countermeasures, and to evade detection based on known patterns.

There are some exceptions: for example, in credit card fraud, there is so much data both of normal transactions and fraudulent ones, and the opportunities for fraud are limited to only a few mechanisms (at least at the retail level) so robust rules for predicting fraud have been developed and are routinely applied. For certain kinds of tax and financial fraud, well-worn techniques are often used and can be checked for explicitly. Even here, though, it is as well to continue to look for newly developed fraud techniques.

The variability of adversarial actions and so profiles in data is often taken to mean that prediction is useless in adversarial settings [17, 24], but that is not the case. First, prediction can be used for risk, so that resources to deal with adversaries are deployed in ways that maximize return on effort. Second, although the particular characteristics of adversaries may be inaccessible or rapidly changing, metacharacteristics such as evasion and social shame are general across many different kinds of adversarial activity, so explicitly looking for such markers may allow useful prediction.

Implication: model building must be inductive overall; but models that explicitly look for metacharacteristics of concealment, manipulation, and shame may see better separations of adversarial records than models trained only on the overt differences in the data.

**'Normal' can be approximated by 'frequent'.** Because adversarial activity is rare and unusual, any records that are 'frequent' in the data, that is that are similar to many other records can plausibly be considered 'normal' (i.e. non-adversarial). In other words, records that capture common activities are unlikely to be adversarial records. Thus it is straightforward to divide a dataset into two parts: common and otherwise. This is a big advantage: the common part is likely to be by far the larger, and attention and resources can be focused on the smaller subset of the remaining records. These records putatively contain records of both adversarial activity and other unusual, rare, and eccentric activity. It may be more difficult to separate these cases when a single adversary (a "lone wolf") is present because it is hard to tell eccentric and bad from eccentric and innocuous. However, much adversarial activity occurs in (small) groups, and so a cluster of similar but unusual records is typically a strong signal for adversarial activity.

Implication: the difficult part of model building is separating unusual records into adversarial ones and others.

## 3 Technical Implications

These characteristics of the data used for modelling in adversarial settings mean that some standard techniques should not be used, while others will still perform well. Table 1 illustrates some of the popular algorithms in each category.

**Table 1.** Appropriateness for adversarial modelling

| | Prediction | Clustering |
|---|---|---|
| Poor | surface fitting<br>tree based<br>support vector machines<br>k-nearest neighbor | distance-based<br>distribution-based |
| Better | ensembles<br>random forests<br>neural networks | density-based<br>hierarchical |

For prediction, any form of surface fitting, for example regression equations, suffers from the problem that it is inherently error minimizing, that large outliers have much influence on the shape of the eventual model, and its general behavior is easily guessed without too much detailed information. Tree-based predictors such as decision trees [23] seem at first as if they should be quite robust predictors – after all, their internal structure is known to change substantially with small changes in the training data (although their prediction accuracies do not

change much). However, it turns out that it is relatively easy to alter the decision boundaries in predictable ways by inserting only a small number of carefully chosen training records [14]. Support vector machines [5, 10], although one of the most effective predictors known, are also vulnerable because the decision boundary chosen depends only on the support vectors, which are typically only a small fraction of the training records. Thus a single inserted record, if it lies close to the decision boundary, can cause the boundary to rotate. This creates regions where the prediction of model differs from the 'true' prediction, and in a predictable way. $k$-nearest neighbor and rule-based predictors base their predictions on some local region of the data space. This makes them vulnerable to adversaries salting particular regions with innocuous-seeming records in order to conceal the presence of adversarial records near them [14].

For clustering, distance-based methods, such as k-means, can separate innocuous and adversarial clusters when they are well-separated – but we have argued that this is usually not the case. If adversarial records lie close to larger clusters of innocuous records, distance-based clustering algorithms will merge them, concealing the existence of the adversarial clusters. Distribution-based methods, such as EM [12], also have trouble separating clusters where one is a small fringe of another. They are likely to see the combination as two overlapped clusters with similar centers and slightly different extents. This is especially so if one or two extra records far away from both clusters can be inserted into the data [13].

Techniques that are effective for adversarial modelling tend to be more complex internally. Their opacity from the point of view of understandability of the model, usually a drawback in mainstream knowledge discovery, makes them hard for adversaries to attack. For prediction, ensemble techniques work well because their prediction is a global integration of the predictions of many component models, built from different subsets of the training data. Some of these component models may be misled by adversarial manipulation, but the fraction that are is related to the fraction of misleading records which must, in practice, be small. Also adversaries cannot control which component models are built from which records, so it is much harder to estimate the effect of a particular manipulation. Random forests [4] are a special form of ensemble in which each component model is built from a subset of both records *and* attributes, making it even harder to manipulate. Since it is one of the most effective predictors known, this makes it the model of choice for prediction in adversarial settings. Neural networks [2] are also resistant to manipulation because they take into account all of the correlation among attributes. It is extremely difficult to produce artificial, misleading records for which correlation matches that of real data; and even harder to do so with a particular purpose to mislead. Unfortunately, neural networks are also expensive to train which limits their usefulness for many real-world problems with large data.

For clustering, density-based methods are more appropriate for adversarial settings because even small, fringe clusters typically have regions of lower density between them and larger clusters that they are trying to resemble. This depends, of course, on being able to choose the distance scale between neighbors

to be smaller than the separation between clusters, which might not be straightforward. Hierarchical clustering starts with each record in its own cluster and repeatedly connects the 'closest' two clusters (where 'closest' can have a number of meanings). A fringe cluster is likely to be connected to its larger neighbor later (and so higher) in the clustering than its size would indicate. Graph-based methods allocate a point to each record and connect points when the associated records are similar, using some measure such as correlation. However, spectral embeddings of the resulting adjacency matrix [27] place points so that the Euclidean distance between them reflects *global* similarity, using some extension of local (pairwise) similarity such as electrical resistance or commute time. Global similarity is emergent from the total dataset, and so is both good at discovering small but consistent separations, and resistant to manipulation.

In general, the way in which boundaries are computed in both prediction and clustering depends on an assumption that the data on either side are representative samples of the 'real' data that is being modelled. In adversarial settings, this is never true because there are so few examples of records of one type, and some records of both types may have been manipulated. Thus choosing boundaries is necessarily more problematic than in mainstream knowledge discovery. There are advantages to using approaches that allow the boundary to be chosen independent of the underlying knowledge-discovery algorithm.

Many prediction algorithms have mechanisms for associating some indication of confidence with each prediction. For example, the distance from the decision boundary in an SVM is a surrogate for confidence; ensemble techniques can provide the margin between the number of component votes for the winning class and the next most popular class; and neural networks can be built to provide a prediction plus a confidence as another output. In each case, these confidences can be interpreted as how much each record resembles those of the innocuous class. In fact, they allow records to be ranked by how ordinary they are.

From such a ranking, a practical decision about where to draw a boundary can be made, but as a separate decision from the construction of the model itself. For example, the density of records in the ranking can itself provide some indication of where normality ends and abnormality begins (under the assumption that frequent = normal). In many practical situations, there are not enough resources to pursue every adversarial record, so it is also useful to be able to determine which are the most egregious cases. For example, in tax evasion, and medical and insurance fraud, there are many more frauds than could be pursued, but it is helpful to be able confidently to pursue those likely to be worst.

Another reason to separate choosing the boundary from building the model is that it enables the overall process to be built in a pipelined, rather than a monolithic way [25, 26]. It is hard to build a single predictor with exactly the appropriate boundary in any setting. However, putting weaker predictors together in a pipeline can achieve better overall performance. The first stage constructs a model but the boundary is set in such a way that there are no (or vanishingly few) false negatives, so no adversarial activity is missed. There will, of course, be many false positives. Those records that are classified as innocuous can now be

safely discarded. Now the remaining dataset can be used to build a second-stage model. It is built from less data but can be more sophisticated. The results of the second model are treated in the same way: records that are certainly innocuous are discarded but any that *might* be adversarial are retained, and a third model built. At each stage, the boundary is chosen to pass records that are potentially adversarial, but to drop records that are certainly innocuous. In a sense, it does not matter how good each stage is, as long as some records are discarded by each one. The accuracy of prediction increases down the pipeline because the pool of training data is smaller, and has a higher fraction of adversarial records.

In clustering, too, it is helpful to be able to consider the position of boundaries between clusters independently of how those clusters were defined or discovered. For example, $k$-means places boundaries halfway between cluster centroids; while EM does not bound clusters at all since they are represented by probability densities. After an EM clustering has been constructed, clusters can be bounded by a particular probability, creating clusters with edges, and regions that do not belong to any cluster. Points within these latter are some form of outlier – they have low probability of being associated with any cluster. Points that lie in the overlap of probability-bounded clusters are ambiguous and may deserve greater attention. Density-based clustering does not have a natural way to impose boundaries after the fact, but using $k$-nearest-neighbor measures within each cluster can discover regions that are less well connected.

In the previous section, we suggested that there are two ways in which adversaries might make themselves *more* differentiated from everyone else: signals associated with their attempts to conceal and manipulate, and signals arising from social shame because they are doing things that are socially unacceptable. These signals might not be very strong if only conventional data is collected and analyzed, but can sometimes be detected if their possible presence is built into the modelling process from the beginning.

One of the problems that adversaries face is that they want, as far as possible, to remain difficult to detect using *any* form of modelling. This puts pressure on them to modify their behavior, and the records it generates, in as many different ways as they can think of. They want to seem 'ordinary' in every conceivable way. This creates records that are unusual in their internal pattern of correlation, because no individual record in a set of 'ordinary' records is individually ordinary.

In settings where the attributes are either the result of direct mental effort or arise from strong social norms, the creation of detectable artifacts is particularly strong. Whenever the normal process associated with an attribute has a large subconscious component, attempting to reproduce the process consciously causes odd results. Humans are just not good at doing something unnatural in a natural way.

For example, language production is largely subconscious. When an individual is saying something to create an effect, rather than because it is the natural way they would say it, there is a tendency for the result to seem stilted and artificial. This is the difference between good actors (rare) and amateur actors – when an amateur is speaking on stage, we keep being reminded that it is performance because the language rhythms and body language are disfluent and improperly

correlated. Fong, Skillicorn and Roussinov [15] showed that attempts to replace 'dangerous' words in communications likely to be intercepted by more innocuous words, using a word code, left a detectable signal in the resulting text, even when the word was well matched (for example, in typical frequency, part of speech, and so on) to the word it replaced.

Humans are extremely good at detecting the signals of unnatural activity behind apparently natural acts – words such as "furtive" and "skulking" exactly capture our ability to do this. Presumably the analysis behind these human skills can eventually be implemented in, for example, video analysis software. Indeed, Saligrama has built systems that are able to predict a change in trajectory of a vehicle before a human can [16].

It is easier to detect adversarial activity when the attributes being captured are either ones that adversaries are unaware of, or ones whose values they cannot change. A major category of such attributes has emerged from recent work in psychology showing that the function words in speech and text, those little words that provide a framework for the content rather than the content itself, are very revealing of internal mental state. For example, changes in the way such words are used can detect deception [21], power in a relationship [9], personality [8], gender [19], and even mental and physical health [6]. Other properties such as authorship and political orientation are also detectable from properties of text [1, 7, 11, 18, 20].

In all knowledge discovery, but especially in adversarial settings, a single model, no matter how sophisticated, should never be used alone. Every model should be hardened by placing it in the context of another, usually simpler, model that deals with expectations. With a single model, whatever happens, happens. With two models, discrepancies between what was expected and what actually happens provide important signals. In adversarial settings, such discrepancies are robust ways to identify potential weaknesses and attempts to manipulate.

Consider a simple predictor. Its decision boundaries are the result of differences between the classes in the training data that was used to build it. This training data almost certainly did not occupy the whole of the potential space spanned by the available attributes; in fact, typically the training data is a small manifold within that space. Two kinds of new data might be presented to the deployed predictor: records from part of the space unlike that used in training; and records that lie near the constructed boundaries. An example is shown in Figure 1. The point labelled A will be classified as a member of the cross class, but there is no justification for this because no data like it was seen during training. The point labelled B will be probably be predicted to be in the circle class, but this prediction should be understood as very weak, because again no data like this was seen during training. Even if this predictor provides confidence information associated with each prediction, the problem with a point like the one labelled A will not be noticed because it is far from the boundary, and so appears to be confidently predicted. This is a weakness in an adversarial setting because some records can slip through the predictor without raising the red flags that they should.

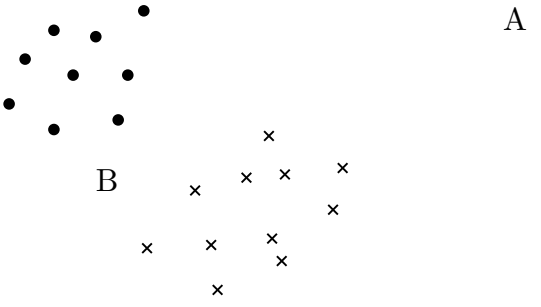



**Fig. 1.** Two points that a predictor trained on the circles and crosses cannot properly interpret

Such a predictor can be hardened by adding extra models that analyze the flow of data and predictions at a meta-level [3]. These extra models need to address three issues: Is the input similar to the data that was used to build the model? Is the mapping of the input approximately what would be expected? Is there any trend in the actions or performance of the predictor? Of course, these extra models can be simpler than the main model – their role is to act as 'sanity checks' on the performance of the main model. But consider how the presence of these extra models helps to harden the predictor. New data that does not resemble training data (for example, point A in the figure) can be reported as unpredictable for that reason, and some other appropriate action can be taken. In an adversarial setting, such records are probably inherently suspicious. New data that does not fit with the structure as understood by the model embedded in the predictor (for example, point B in the figure) can be reported as interesting. This interesting label may be interpreted as an attribute of the record, causing it to receive more careful scrutiny, but it could also be interpreted as indicating a weakness of the model itself – perhaps that it is not rich enough to represent the data fully, or that the boundaries are not properly placed. Finally, watching the input data as a stream, and the predictions and confidences as a stream, allows the detection of changes in the outside world that indicate that the predictor should be retrained. For example, if statistics of the input records are changing or the distribution of predictions across classes is changing, the setting is probably changing too, and it is dangerous to continue to use a predictor built for a situation that no longer exists.

## 4 Conclusion

Knowledge discovery in adversarial settings is not simply another application domain of mainstream knowledge discovery. The nature of the problem, and both the difficulties and new opportunities provided by the properties of adversaries require rebuilding the technology of knowledge discovery from the ground up. Some techniques, for example ensemble predictors, continue to have a role,

while some mainstream technologies, for example support vector machines, are no longer appropriate. Simple models of the process, such as DM-CRISP, are not appropriate either, but the "arms race" nature of the problem needs to be incorporated, so that adversarial knowledge discovery can exploit incremental and stream techniques. At present, adversarial knowledge discovery is being applied in areas such as law enforcement, insurance, counterterrorism, and money laundering. However, an increasing number of domains will become at least a little adversarial in nature as the benefits of knowledge discovery become better understood, and the players fight to share these benefits equitably. Adversarial knowledge discovery is therefore a growth area, both from a research and application perspective.

## References

1. Abbasi, A., Chen, H.: Visualizing authorship for identification. In: Mehrotra, S., Zeng, D.D., Chen, H., Thuraisingham, B., Wang, F.-Y. (eds.) ISI 2006. LNCS, vol. 3975, pp. 60–71. Springer, Heidelberg (2006)
2. Bishop, C.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1995)
3. Bourassa, M.A.J., Skillicorn, D.B.: Hardening adversarial prediction with anomaly tracking. In: IEEE Intelligence and Security Informatics 2009, pp. 43–48 (2009)
4. Breiman, L.: Random forests–random features. Technical Report 567, Department of Statistics, University of California, Berkeley (September 1999)
5. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. In: Data Mining and Knowledge Discovery, vol. 2, pp. 121–167 (1998)
6. Campbell, R.S., Pennebaker, J.W.: The secret life of pronouns: Flexibility in writing style and physical health. Psychological Science 14(1), 60–65 (2003)
7. Chaski, C.E.: Who's at the keyboard: Authorship attribution in digital evidence investigations. International Journal of Digital Evidence 4(1) (2005)
8. Chung, C.K., Pennebaker, W.J.: Revealing dimensions of thinking in open-ended self-descriptions: An automated meaning extraction method for natural language. Journal of Research in Personality 42, 96–132 (2008)
9. Chung, C.K., Pennebaker, J.W.: The psychological function of function words. In: Fiedler, K. (ed.) Frontiers in Social Psychology. Psychology Press (in press)
10. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press, Cambridge (2000)
11. de Vel, O., Anderson, A., Corney, M., Mohay, G.: Mining E-mail content for author identification forensics. SIGMOD Record 30(4), 55–64 (2001)
12. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, Series B 39, 138 (1977)
13. Dutrisac, J.G., Skillicorn, D.B.: Hiding clusters in adversarial settings. In: 2008 IEEE Intelligence and Security Informatics, pp. 185–187 (2008)
14. Dutrisac, J.G., Skillicorn, D.B.: Subverting prediction in adversarial settings. In: 2008 IEEE Intelligence and Security Informatics, pp. 19–24 (2008)
15. Fong, S.W., Skillicorn, D.B., Roussinov, D.: Detecting word substitutions in text. IEEE Transactions on Knowledge and Data Engineering 20(8), 1067–1076 (2008)

16. Jodoin, P.-M., Konrad, J., Saligrama, V., Gaboury, V.: Motion detection with an unstable camera. In: IEEE International Conference on Image Processing, pp. 229–232 (2008)
17. Jonas, J., Harper, J.: Effective counterterrorism and the limited role of predictive data mining. Policy Analysis 584, 1–12 (2006)
18. Koppel, M., Akiva, N., Alshech, E., Bar, K.: Automatically classifying documents by ideological and organizational affiliation. In: Proceedings of the IEEE International Conference on Intelligence and Security Informatics (ISI 2009), pp. 176–178 (2009)
19. Koppel, M., Argamon, S., Shimoni, A.R.: Automatically categorizing written texts by author gender. Literary and Linguistic Computing 17(4), 401–412 (2002)
20. Koppel, M., Schler, J., Bonchek-Dokow, E.: Measuring differentiability: Unmasking pseudonymous authors. Journal of Machine Learning Research 8, 1261–1276 (2007)
21. Newman, M.L., Pennebaker, J.W., Berry, D.S., Richards, J.M.: Lying words: Predicting deception from linguistic style. Personality and Social Psychology Bulletin 29, 665–675 (2003)
22. Pietronero, L., Tosattib, E., Tosattib, V., Vespignani, A.: Explaining the uneven distribution of numbers in nature: the laws of Benford and Zipf. Physica A: Statistical Mechanics and its Applications 1-2, 297–304 (2001)
23. Quinlan, J.R.: Induction of decision trees. Machine Learning 1, 81–106 (1986)
24. Schneier, B.: Why data mining won't stop terror. Wired (2006)
25. Senator, T.E.: Multi-stage classification. In: Proceedings of the Fifth IEEE International Conference on Data Mining, pp. 386–393 (2005)
26. Skillicorn, D.B.: Knowledge Discovery for Counterterrorism and Law Enforcement. CRC Press, Boca Raton (2008)
27. von Luxburg, U.: A tutorial on spectral clustering. Technical Report 149, Max Plank Institute for Biological Cybernetics (August 2006)

# Chapter 11
# Analysis and Mining of Online Communities of Internet Forum Users

Mikołaj Morzy*

Institute of Computing Science,
Poznan University of Technology,
Piotrowo 2, 60-965 Poznan, Poland
Mikolaj.Morzy@put.poznan.pl

**Abstract.** In this chapter we provide an overview of Internet forums, their architecture, and characteristics of social-driven data generated by the online community of Internet forum users. We discuss issues involved in Internet forum data acquisition and processing, and we outline some of the challenges that need to be addressed. Then, we present a framework for analysis and mining of Internet forum data for social role discovery. Our framework consists of a multi-tier model, with statistical, index and network analysis tiers serving as knowledge discovery tools at different levels of analysis. We also show how using methods of social network analysis, in particular, the analysis of egocentric graphs of Internet forum users, may help in understanding social role attribution between users.

## 1 Introduction

In this section we introduce the reader into the world of Web 2.0. In Section 1.1 we informally define Web 2.0 and we discuss main rules for Web 2.0 applications, explaining how this notion revolutionizes modern software development. We concentrate on distinct features that differentiate Web 2.0 from previous software development paradigms and we demonstrate how changes induced by Web 2.0 affect the data generated by social applications. In particular, in Section 1.2 we describe an interesting switch from the pull architecture to the push architecture of information transfer. This switch bears considerable consequences for the research presented in the chapter. Section 1.3 is devoted to Internet forums, which are a new form of conversation. We describe the nature of Internet forums and unique characteristics of this communication medium.

### 1.1 What Is Web 2.0?

The term *Web 2.0* has been coined by Tim O'Reilly in 2004 during the first O'Reilly Media Web 2.0 conference. According to O'Reilly [20]

* Research supported by the Polish Ministry of Science grant N N516 371236.

D.E. Holmes, L.C. Jain (Eds.): Data Mining: Found. & Intell. Paradigms, ISRL 25, pp. 225–263.
springerlink.com 

> Web 2.0 is the business revolution in the computer industry caused by the move to the Internet as platform, and an attempt to understand the rules for success on that new platform. Chief among those rules is this: build applications that harness network effects to get better the more people use them.

The above definition of Web 2.0 can be extended by a few rules formulated by O'Reilly [20]:

- **the perpetual beta:** software is not an artifact, it is a way to engage with users,
- **small pieces loosely joined:** data and services should be opened to others for re-use, similarly, external data and services should be re-used whenever possible,
- **software above the level of a single device:** applications are not bound to specific clients or servers, instead, they exist in the space between devices,
- **the law of conservation of attractive profits:** open APIs and standard protocols do not contradict the idea of competitive advantage,
- **data is the Intel inside:** source of future lock-in and competitive advantage is not software or hardware architecture, but user-generated data, namespaces, and proprietary formats.

The above rules are important and relevant in the context of the chapter, because they define a business model transition. Many people perceive Web 2.0 as a set of buzzwords, teenager Web applications, and a certain graphic design style. These people criticize the entire idea of Web 2.0 as being yet another dot-com bubble, similar to the one of March, 2000. Contrary to these claims, we will accept the perspective advocated by O'Reilly et al., that Web 2.0 defines a new quality in software engineering and provides a new paradigm for both the application design and mining of social data. Figure 1 is the famous mind map of Web 2.0 created by Markus Angermeier and it perfectly summarizes main themes and subjects of Web 2.0.Below we discuss the rules of Web 2.0 in detail and we explain how they relate to the research on Internet forums.

The first rule disqualifies software as an artifact, a by-product of software development, but defines software as means to communicate and engage with users. Indeed, a user contribution is crucial for the success of modern software. This rule coincides with the second rule that encourages opening of application programming interfaces (APIs) and services, as well as sharing the data. One can hardly challenge this point. The immense success of services such as Google Search, Goolge Maps, Facebook, Amazon, and many others, lies primarily in enabling access to these services via open APIs. Openness and accessibility of both services and data allow thousands of external contributors to develop value-added solutions on top of existing systems. As an example, consider the hugely successful social network Facebook. In only six months dating from the launch of Facebook Platform, the open API to create applications that can be merged with Facebook personal pages or that utilize core Facebook features, the number of applications created by external contributors exceeded 10 000! As


**Fig. 1.** The map of Web 2.0

another example, consider Yahoo! Pipes, a mash-up feed aggregator that allows users to create vertical search applications merging hundreds of data sources into one flow of logic and data. Today, thousands of pipes exist that aggregate news and data feeds from places so diverse as the photo sharing service Flickr, the weather forecast service provided by US National Oceanic and Atmospheric Administration, stock quotes from Google Finance, or personal favorite music lists scratched from iTunes feeds. This re-usal of data and services creates incredible synergy of Web 2.0 services. The third rule basically departs from deprecated software architectures, such as the client-server architecture, and advocates the adoption of multi-tier and service-oriented architectures, where software components act like autonomous agents communicating and exchanging data. This rule is mainly dictated by unprecedented advances in personal and mobile computing hardware. The pace of innovation in these devices makes software development window very tight and, effectively, dictates the need to develop software independently of a current hardware platform. The fourth rule is the axiom of open source community. The main idea is that the profit and competitive advantage lies not in proprietary closed code, but in innovation, adaptation, and ease of approval. This belief, being questionable a few years back, gains more weight as the number of free services increases. In almost all software categories one may find open source replacements of proprietary software, being it office suites, business intelligence tools, web analytics, hosting services, etc. Finally, the most important fifth rule states that the main reason for lock-in and the most valuable asset is neither software nor hardware, but the data itself. This conviction has been long popular among database engineers, but not widespread in computer science community. Software platforms change every five years, hardware platforms change every seven years, but the data remain forever. This claim may seem exaggerated, but it is counter-intuitively correct. Most contemporary database systems do not consider deleting any data and many free services do not impose limits on the volume of data being stored. Storing and querying

data has become easier with advances in relational database technology, but acquiring data and feedback from users is difficult. The main mantra of Web 2.0 proponents is to turn spectators into participants. This means creating incentives for users to participate and contribute. Many solutions from the world of Web 2.0 rely heavily on users' input, e.g., folksonomies, blog networks, wikis, recommender engines. Companies that manage to attract the biggest crowd of devoted followers are most likely to gain competitive edge and win the race. Other sources of success are namespaces and proprietary formats. Examples of namespaces include Compact Disc Database (CDDB) operated by Gracenote Inc., Network Solutions managing almost 8 million domain names, or VeriSign managing digital certificates, payments processing, and root nameservers. Proprietary formats include, among others, formats from Microsoft Office tools or Apple iTunes. All these examples represent popular formats and solutions that have reached the critical mass of popularity and acceptance, and are perceived as *de facto* standards despite being proprietary assets.

The above rules define the transition to a new software development paradigm, a new approach to users, and a new approach to collaboration and cooperation. By itself, these changes should not affect any data mining activities, because the algorithms for pattern discovery are oblivious to particular software architecture. However, the paradigm shift enforced by the Web 2.0 revolution strongly affects the data gathered and processed by Web 2.0 applications. As we will discuss in Section 2, the changes to data distributions, data quality and data semantics are so profound, that traditional data mining algorithms cannot cope with these data and new methods must be developed in order to discover meaningful patterns.

### 1.2 New Forms of Participation — Push or Pull?

One notable change introduced by Web 2.0 patterns in software development is the switch from the *pull* architecture to the *push* architecture. Generally, it is not an improvement, but merely a philosophical change. For many years proponents of both architectures have been waging a religious war over merits of both approaches. Briefly speaking, the pull architecture refers to a situation where information is actively searched and retrieved (pulled) from the network. A classical example of the pull architecture is the World Wide Web, where users send HTTP requests to servers in order to obtain documents. In contrast, in the push architecture a recipient of information waits passively and information is being automatically delivered (pushed) by an underlying framework. An example of the push architecture is electronic mail, where desired information is pushed into recipient's mailbox by an underlying mail transfer protocol. The particular choice of one architecture over another has many important consequences outlined below.

- *openness*: pull is publicly available and can be performed by anyone, push requires prior registration, subscription, or any other type of relationship,
- *demand anticipation*: pull is unpredictable and difficult to forecast, push is limited to a known number of subscribes,

- *deliverability*: pull is usually based on request-response semantics and can be armored with delivery guarantees, push must use some kind of acknowledgment protocol to guarantee delivery,
- *updates*: pull is less scalable w.r.t. updates due to constant polling for updates, push is more scalable as updates can be managed by notifications,
- *network bandwidth*: pull is suited for frequent polling because response is delivered for every request, push is suited for infrequent updates because notifications are sent only when necessary.

In Web 2.0 environment the push architecture is strongly preferred over the pull approach. Pushing data to users is the leading characteristic of technologies such as blogs, podcasts, instant messaging, mashups, and pipes. The departure from the model in which a user is actively searching for information in favor of information actively searching for a user results in a drastic improvement of quality and reliability of information obtained by a user. This phenomenon can be explained as follows. As we have noticed earlier, the push approach is based on a subscription (or a similar relationship) to a data source. Subscriptions may be managed individually for each service (a podcast or a blog), but most users rely on centralized subscription hubs to manage their subscriptions. For podcasts, the most important hub is the iTunes online service. For blogs, there are blog ranking sites (Technorati, Alexa), blog hosting sites (Blogger), blog publishing systems (Wordpress), and blog aggregators (Bloglines, Google Reader). These hubs are used both as entry points to the blog network (or podcasts, or any other information service), and as quality guarantors. Hubs promote high quality data and help disseminate valuable content among information recipients. As the result, quality and reliability of data increase, making data more attractive from data mining perspective. We discuss implications of the Web 2.0 architecture on social-driven data in greater detail in Section 2.

### 1.3 Internet Forums as New Forms of Conversation

An Internet forum is a Web application for publishing user-generated content under the form of a discussion. Usually, the term *forum* refers to the entire community of users. Discussions considering particular subjects are called *topics* or *threads*. Internet forums the Latin plural *fora* may also occasionally be used) are sometimes called Web forums, discussion boards, message boards, discussion groups, or bulletin boards. Internet forums are not new to the network community. They are successors of tools such as Usenet newsgroups and bulletin board systems (BBS) that were popular before the advent of the World Wide Web. Messages posted to a forum can be displayed either chronologically, or using threads of discussion. Most forums are limited to textual messages with some multimedia content embedded (such as images or flash objects). Internet forum systems also provide sophisticated search tools that allow users to search for messages containing search criteria, to limit the search to particular threads or subforums, to search for messages posted by a particular user, to search within the subject or body of the post, etc.

The most important feature of Internet forums is their social aspect. Many forums are active for a long period of time and attract a group of dedicated users, who build a tight social community around a forum. With great abundance of forums devoted to every possible aspect of human activity, such as politics, religion, sports, technology, entertainment, economy, fashion, and many more, users are able to find a forum that perfectly suits their needs and interests. Usually, upon joining the forum a user pledges to adhere to the netiquette of a forum, i.e., the set of rules governing the accepted format and content of posts. Some forums are very strict about enforcing netiquette rules (e.g., a family forum may not tolerate any form of cursing or sexually explicit content), and some forums do not enforce netiquette rules at all. Two types of special users are present to protect the forum and enforce the netiquette. Administrators have a complete set of rights and permissions to edit and delete abusing posts, to manage threads of discussion, to change global forum settings, to conduct software upgrades, to manage users and their accounts, to ban users who do not comply with the netiquette, and to stick popular threads and create word filters for posts. Moderators enjoy a subset of rights and permissions granted to administrators. Moderators are commonly assigned the task to run a single forum or a single thread and moderate the discussion by issuing warnings to users who do not comply with the netiquette. Moderators can suspend users, edit and delete questionable posts, and temporarily ban users who breach the rules of the forum. Forums may require a registration and a login prior to posting a message, but there are also popular forums where anonymous users are allowed to post. As we shall see in Section 2, anonymity and pseudo-anonymity drastically lower the quality of data and information available on a forum. Besides, the registration requirement creates a strong liaison between a user and a forum, building durable social bindings. Unfortunately, even registration does not shield forum community from trolls, malevolent users whose sole intention is to spark heated discussion, ignite controversy, or post offensive and abusive contents. Trolls usually have no other merit or purpose than to irritate and offend other users, misusing pseudo-anonymity offered by the Internet. Trolling is just one example of the obtrusive social cost of cheap pseudonyms which can be created and used at no cost. Indeed, trolls fear no real retaliation for their vandalizing behavior other than a ban on a given identity. As the registration is almost always free of charge, there is no incentive for a troll to preserve her identity.

In Section 1.2 we have discussed major differences between the pull and push software architectures. Internet forums are a great example of the pull architecture. Users are required to login to a forum on a regular basis and follow discussions pro-actively. Without regular readership and involvement a user can easily loose the context of discussion or weaken the ties to the community. Therefore, Internet forums require active user participation. Some forums offer email notification, which allows users to subscribe to notifications about posts to a specific thread or subforum that is particularly interesting to them. Unfortunately, this does not significantly change the overall pull architecture, because this option is of little usefulness. There are simply too many posts for an email

interface to handle and users have to choose between having to login to a forum and browse posts manually, or cluttering their email inbox with hundreds of highly similar messages.

As we have noted earlier, Internet forums are not a new concept, but merely an improvement over Usenet newsgroups and dial-up bulletin boards. They differ from Usenet groups in that they do not use email interface or specialized news reader, but utilize a standard Web interface accessible from a standard browser. They differ from blogs, because they use the pull architecture (as opposed to the push architecture commonly used by blog readers) and they support the model of "many writers, many readers" (as opposed to the model of "one writer, many readers" employed e.g. by blogs). Also, the ability to create strong social bonds among members of a forum is quite unique with respect to other Web 2.0 technologies. We use this distinctive feature of Internet forums to mine forum data in search of social roles of participants in Section 3.

## 2 Social-Driven Data

In this section we investigate the properties of social-driven data. In Section 2.1 we introduce and formally define the notion of social-driven data and we outline the problems and challenges in analyzing these data. In Section 2.2 we describe data harvested from Internet forums. We first present the technological background on the Internet forum architecture and we describe volumes and nature of Internet forum data.

### 2.1 What Are Social-Driven Data?

The Web 2.0 revolution changes many things, among others, it creates a flood of social-driven data. An enormous increase in volumes of data being produced and gathered can be only compared to the widespread adoption of automatic data generation and data gathering devices witnessed in the mid-90's, when technologies such as bar code readers ushered in the immense increase in volumes of data processed by information systems. These data were generated and processed automatically, giving birth to data mining and online analytical processing. By contrast, data to be analyzed and mined today are not created automatically, but they emerge from social relationships and direct inputs from humans. Wikis, tags, blogs, or Internet forums produce vast amounts of data, but the quality and consistency of the data is questionable, to say the least.

Prior to defining the notion of social-driven data, we need to introduce the notions of a social network and a social process. A social network is a structure made of entities that are connected by one or more types of interdependency. Entities constituting a social network may represent individuals, groups or services, and relationships between entities reflect real-world dependencies resulting from financial exchange, friendship, kinship, trust, dislike, conflict, trade, web links, sexual relationship, transmission of diseases, similarity of hobbies, co-occurrence, participation, etc. Relationships may be directed or undirected,

as well as weighted or unweighted. This description is very broad and covers many different types of network structures. The most important and distinctive feature of a social network is the fact, that the entities constituting the network represent humans or human groups. The nature of a network and its parameters (such as density, average path length, diameter, centrality, etc.) are defined by the characteristics of relationships linking individual entities. Therefore, the analysis methods are mostly driven by the definition of relationships.

A social process is a process involved in the formation of groups of persons. The formation of groups may lead to absorbing one group into another (assimilation), to achieve an advanced stage of development and organization (civilization), to become or being made marginal, especially as a group within a larger society (marginalization), or to engage in an activity for pay or as a means of livelihood (professionalization). A social process can be perceived as any kind of process involving humans or human groups. We use such broad definition purposefully to allow both online auctions and moving objects to fall into this category. Social processes emerging around social networks bear important implications both for content creators and content consumers.

Social-driven data are the data created by an underlying social process or acquired from a social network. Social-driven data are either created explicitly by humans (tags, wikis, blogs) or created as a direct result of human social activities (trading, moving, conversating). This definition of social-driven data is general enough to contain blogs and Internet forums as obvious examples of social-driven data, but also online auctions and moving objects as data resulting from social processes. Social-driven data are, in many aspects, different from traditional structured data. Main differences arise from the way social data are generated, acquired, and used. Below we outline main problems and challenges in analyzing social-driven data.

1. *Social-driven data are incomplete.* Apart from rare situations when social networking sites publish their data, most available social-driven data is acquired by programmatically scratching social networking sites with crawlers. This method can be used to gather large amounts of blog data, Internet forum data, or wikis. Unfortunately, harvesting data from the Web using crawlers imposes limitations that are difficult to overcome. Firstly, crawlers must adhere to the netiquette with respect to the frequency of requests and available download paths. These rules are often explicitly defined by the means of the Robots Exclusion Protocol (also known as `robots.txt` protocol) or Robots `<META>` tag. Secondly, crawlers are capable of reaching only the surface Web, i.e., the portion of the World Wide Web indexed by search engines. Underneath the surface Web, also known as the visible Web, lurks a much bigger deep Web, also referred to as the Deepnet, the invisible Web, or the hidden Web. This is the part of the Internet that is unreachable for search engines. There are many reasons of why a particular document may belong to the deep Web. For instance, almost all dynamic content generated in response to parametrized user queries is unreachable to crawlers, as well as content belonging to the private Web hidden behind a mandatory login.

Recently, scripting languages and frameworks such as JavaScript, Flash or AJAX are often used, but the links generated by client-side scripts are not accessible to crawlers. It is estimated that the deep Web is larger than the visible Web by a few orders of magnitude. As of the time of writing the size of the deep Web is estimated to reach 100 000 TB whereas the surface Web amounts to merely 170 TB.

2. *Social-driven data are dirty.* This is probably the biggest problem hindering the ability to analyze and process social-driven data. The amount of spam in social-driven data is difficult to measure, and depending on the source of information the estimates vary from 80% to over 95% of all user-generated data being spam. As of 2010, there were more than 140 million blogs indexed by Technorati blog search engine, among them around 35-40 million non-spam blogs and 12-16 million active blogs in terms of new posts. Probably, the most meaningful part of the blogosphere is covered by 4-5 million blogs and all the rest is just plain rubbish. Spam social sites are so abundant due to their impact on the ranking of documents produced by search engines. These sites serve two main purposes: to display ads and to artificially inflate the ranking of the referred sites. In the first case, a social site contains a copy-paste of some meaningful content stolen from a regular site or a mashup of popular keywords randomly thrown into legitimate text. In the second case, a social site belongs to a spam farm (a network of thousands of interlinked sites) to give its referred sites more inbound links and deceive link mining algorithms, such as PageRank. Pruning spam content is difficult and requires an intensive pre-processing of harvested social-driven data.
3. *Social-driven data lack structure.* The ability to derive structure is essential in mining any type of data. Unfortunately, with social-driven data it is very difficult to assume anything about the structure and the format of data acquired from crawlers. The primary cause of this difficulty is the lack of standards regarding social-driven data. For Internet forums no agreed upon formats exist and almost every website uses custom data structures. As a consequence, a tailor-made mapping must be constructed for each single website to cast the data harvested by a crawler into a database. Needless to say, this requirement makes social-data acquisition expensive and cumbersome.
4. *Social-driven data are mutable.* Social-driven data are usually presented in the form of a graph or a network. Nodes in the network appear and disappear freely, relationships forming edges in the network may fade over time, and the topology of the network changes constantly. As the result, constructing a static view of the network may be difficult, if not impossible. The lack of a single unifying view of the network makes the transition between network states indeterministic and must be accounted for in every algorithm for mining social-driven data. On the other hand, social-driven data usually form a scale-free network [6], with the distribution of node degrees given by $P(k) \sim k^{-\gamma}$ where $P(k)$ is the probability that a node has the degree of $k$, and the coefficient $\gamma$ usually varies between 2 and 3. In other words, a scale-free network consists of a small number of high degree nodes, called hubs, and a large number of nodes with low degree. Because social-driven

data networks grow accordingly to the Barabási-Albert model (also known as the preferential attachment), the resulting network is a scale-free network. One of the most important properties of scale-free networks is their robustness and resistance to failures. If nodes in the network fail with equal probability, then a failure of an ordinary node does not affect the rest of the network, these are only failures of hubs that may influence the network as a whole. Because the number of hubs is negligible when compared to the size of the network, the peril of bringing the network down is minimal. In addition, every change of an ordinary node passes almost unnoticed. From the algorithmic point of view, only a small percentage of data alterations is interesting, and all alterations of ordinary nodes may be ignored without a significant loss of precision, because the alterations of ordinary nodes do not affect the global properties of the social networks.

5. *Social-driven data are multi-lingual.* In the beginning, the *lingua franca* of the Internet was English. Today, the Web has become linguistically much more diverse. According to Internet World Stats [10], top 10 languages used in the Web cover almost 85% of Internet users population, with English accounting for 30.4% of all Internet users, Chinese accounting for 16.6% of Internet users, and Spanish accounting for 8.7% of Internet users. Interestingly, the fastest language growth in the Internet can be observed for Arabic (2060% in years 2000–2008), Portuguese (668% over the same period) and Chinese (622% over the same period). Often, mining social-driven data requires some basic degree of natural language processing. When analyzed documents are written in different languages, multiple lexers and parsers are required, with multiple tokenizers and stop-word lists. Furthermore, semantic similarity is harder to discover when translations are present in the data.

Having outlined the main problems and challenges in mining social-driven data, we now proceed to the presentation of the data crawled from Internet forums.

### 2.2 Data from Internet Forums

Internet forums have recently become the leading form of peer communication in the Internet. Many software products are available for creating and administering Internet forums. Usually, Internet forums are complex scripts prepared in the technology of choice: PHP, CGI, Perl, ASP.NET, or Java. Threads and posts are stored either in a relational database or in a flat file. Internet forums vary in terms of functions offered to its users. The simplest forums allow only to prepare textual messages that are displayed chronologically using threads of discussion. Sophisticated forum management systems allow users to prepare posts using visual editors and markup languages, such as BBCode or HTML, and to address replies to individual posts. In particular, using BBCode tags makes the analysis and parsing of forum posts difficult. BBCode (Bulletin Board Code) is a lightweight markup language used to beautify discussion list posts. The use of BBCode allows forum administrators to turn off the ability to embed HTML content in forum posts for security reasons, at the same time allowing users

to prepare visually attractive posts. Unfortunately, different implementations of the BBCode markup are not compatible across Internet forum management systems, with varying keyword lists or inconsistencies in enforcing capital letters in BBCode tags.

There are numerous platforms for Internet forum management. PHP Bulletin Board (PhpBB) is a very popular Internet forum software written entirely in the PHP programming language and distributed using the GNU General Public License as open source software. Since its publication in the year 2000, PhpBB has become undoubtedly the most popular open source software for managing Internet forums. PhpBB is characterized by a simple and intuitive installation process, overall friendliness of the interface, high efficiency, flexibility and customizability. An additional advantage of PhpBB is a very large and supportive user community providing inexperienced users with many tutorials and guidelines. PhpBB is one of the very few solutions that provide interfaces to almost all contemporary database management systems, among others, to MySQL, PostgreSQL, SQL Server, FireBird, Oracle, and SQLite. Invision Power Board (IPB) is produced by Invision Power Services Inc. IPB is a commercial product popular among companies and organizations due to its low cost of maintenance and effective support. IPB provides interfaces only to the most popular databases: MySQL, Oracle, and SQL Server. Differently from PhpBB, IPB is a fully-fledged content management system, which releases administrators from the bulk of manually creating and managing HTML pages. In addition, IPB fully utilizes newest capabilities of asynchronous data transfer using AJAX and JavaScript calls, and supports a modular forum design using the technology of plugins.

**Table 1.** Top 10 biggest Internet forums

| | **forum name** | **technology** | **# posts** | **# users** |
|---|---|---|---|---|
| 1 | Gaia Online | PhpBB | 1 331 380 030 | 12 589 038 |
| 2 | Club RSX | vBulletin | 1 058 841 235 | 99 915 |
| 3 | IGN Boards | IGNBoards | 179 160 942 | 1 255 254 |
| 4 | Nexopia | custom | 160 156 027 | 1 334 380 |
| 5 | FaceTheJury | custom | 143 692 161 | 552 000 |
| 6 | 4chan | custom | 126 997 430 | 26 328 |
| 7 | Vault Networks | IGNBoards | 117 556 542 | 626 203 |
| 8 | d2jsp | custom | 102 958 312 | 340 997 |
| 9 | Jogos | PhpBB | 100 694 272 | 167 357 |
| 10 | Offtopic.com | vBulletin | 99 561 264 | 192 169 |

In order to better understand scalability-related challenges of mining Internet forum data, we now present statistics on selected forums. All statistics were gathered by Big Boards (`www.big-boards.com`), the catalog of the biggest Internet forums exceeding 500 000 posts. Estimating the size of a forum is a difficult and error-prone task. By far the most popular metric is the total number of posts. This metric can be biased by the presence of automatically generated spam messages, but other metrics are also susceptible to manipulation. For instance, measuring

the number of registered users may be tricky, because many Internet forums allow posts not only from directly registered users, but from users who registered to other parts of a portal as well. Similarly, the frequency of posting per registered user may be biased by the vague definition of the term "registered user". Table 1 summarizes the Top 10 Biggest Internet Forums list. This list is very interesting as it clearly demonstrates the diversity of Internet forums. The biggest forum, Gaia Online, is an anime role-playing community, with a huge base of members. A relatively small community of users constitutes Club RSX, a forum for Honda RSX fans. FaceTheJury is a set of real-life picture rating forums and 4chan is a message board for discussing topics related to Japanese culture. The numbers behind these forums demonstrate the unprecedented sizes of virtual communities emerging around Internet forums. It is clear that such vast repositories of data and information cannot be analyzed and browsed manually, and that automated methods for knowledge discovery and extraction are necessary.

**Table 2.** Top 10 Internet forum software technologies

| | **technology** | **# forums** | **% forums** |
|---|---|---|---|
| 1 | vBulletin | 1303 | 62% |
| 2 | IGN Boards | 276 | 13% |
| 3 | PhpBB | 246 | 12% |
| 4 | custom | 105 | 5% |
| 5 | SMF | 42 | 2% |
| 6 | UBB | 41 | 2% |
| 7 | ezBoard | 37 | 2% |
| 8 | MesDiscussions.net | 18 | 1% |
| 9 | ASP Playground | 16 | 1% |
| 10 | Burning Board | 14 | 1% |

Table 2 presents the list of top 10 Internet forum software technologies. This list presents the distribution of Internet forum software technologies only for Internet forums indexed by Big Boards. From the results we may conclude that the majority of popular Internet forum software technologies are commercial products (vBulletin, IGN Boards, UBB, ezBoard, MesDiscussions.net, or Burning Board), but open source solutions are paving their way (PhpBB, SMF). As for the programming language of choice, Table 3 shows that the overwhelming majority of Internet forums are implemented in PHP, with negligible presence of ASP.NET and Perl/CGI.

Initially, main subjects of discussions on Usenet groups and bulletin boards were related to computers and technology. With the widespread adoption of the Internet within the society, discussion boards and Internet forums were quickly attracting more casual users. Today, entertainment and computer games are the most popular categories of discussions. Table 4 summarizes the most popular categories of Internet forums. The data clearly shows that Internet forums span a very broad spectrum of subjects and, importantly, there is no dominant category or subject.

**Table 3.** Top Internet forum implementation languages

| | language | # forums | % forums |
|---|---|---|---|
| 1 | PHP | 2047 | 94% |
| 2 | ASP.NET | 49 | 2% |
| 3 | Perl/CGI | 43 | 2% |
| 4 | SmallTalk | 22 | 1% |
| 5 | ColdFusion | 11 | 1% |
| 6 | Java | 7 | 0% |

**Table 4.** Top 10 Internet forum categories

| | category | # forums | % forums |
|---|---|---|---|
| 1 | entertainment | 438 | 17% |
| 2 | recreation | 408 | 16% |
| 3 | games | 391 | 15% |
| 4 | computers | 324 | 13% |
| 5 | society | 257 | 10% |
| 6 | sports | 221 | 9% |
| 7 | general | 218 | 9% |
| 8 | art | 138 | 5% |
| 9 | home | 79 | 3% |
| 10 | science | 70 | 3% |

Finally, let us consider the degree of internationalization of Internet forum communities. Table 5 presents the distribution of languages spoken on Internet forums. As expected, Internet forums are dominated by English, followed by European languages (German, French, Dutch, etc.). What comes as a surprise is that Chinese is not very popular among Internet forum users, as compared with the distribution of languages for the Web.

## 3 Internet Forums

This section is devoted to problems and challenges in mining Internet forums. In Section 3.1 the acquisition process for Internet forum data is described, with special emphasis on difficult obstacles one must cope with when crawling Internet forums. Section 3.2 presents a statistical method for Internet forum analysis. In particular, base statistics used to construct indexes are defined. An index-based analysis of Internet forums is introduced in Section 3.3. Internet forum communities form strong social networks. Methods for mining this social network in search for social roles are presented in Section 3.4.

### 3.1 Crawling Internet Forums

Analysis and mining of Internet forum data requires data to be cleansed, pruned, pre-processed, and stored conveniently, most favorably in a relational database

**Table 5.** Top Internet forum languages

| | language | # forums | % forums |
|---|---|---|---|
| 1 | English | 1641 | 77% |
| 2 | German | 137 | 6% |
| 3 | French | 74 | 3% |
| 4 | Dutch | 52 | 2% |
| 5 | Russian | 48 | 2% |
| 6 | Spanish | 46 | 2% |
| 7 | Italian | 34 | 2% |
| 8 | Turkish | 32 | 2% |
| 9 | Chinese | 19 | 1% |
| 10 | Swedish | 18 | 1% |
| 11 | Polish | 17 | 1% |

due to the maturity of the relational database technology and the broad availability of tools for mining relational databases. Unfortunately, crawling Internet forums is a daunting and difficult task. Forum discussions can lead to tree-like structures of arbitrary depths, posts pertaining to a single discussion can be paginated, and Internet forum engines, such as PhpBB, can be personalized, leading to HTML code that is different from the original engine HTML output. Therefore, a robust and reliable method for crawling and harvesting Internet forum pages into a structured form must be developed.

The first step in Internet forum analysis is the development of a web crawler capable of crawling and downloading an entire forum identified by its URL address. Preliminary results presented in this chapter used an open-source library WebSphinx to automate the majority of repetitive and difficult tasks. Among others, the WebSphinx library takes care of the following tasks: maintaining the cache of URL addresses to visit, scheduling requests, conforming to the netiquette, and managing the set of threads that access the address cache in parallel. For each document processed by the crawler, the method `visit(Page)` is called to parse and analyze the contents of the document. In addition, for each link encountered on the processed page the result of the method `boolean shouldVisit(Link)` indicates whether the link should be followed, i.e., whether the link should be added to the address cache. The address cache is accessed in parallel by multiple threads of the crawler, where each thread picks a single address from the cache and processes it. Additional parameters can be set that govern the behavior of the crawler, e.g., the depth of the crawl or the maximum size of documents to be fetched. To make the crawler general enough to handle different Internet forums, the crawler is implemented using abstract classes for document processing and URL analysis. In this way, adding a new Internet forum engine template to the crawler is simplified and the architecture of the crawler becomes flexible and extensible.

Downloaded documents are parsed in search of topics and posts. Then, the discovered structures are loaded into the database. The database schema for the Internet forum analysis is fairly simple and consists of seven tables joined

by several foreign keys. Main tables include `PARENT_FORUM` and `FORUM` for storing hierarchical structure of Internet forums, `AUTHOR` and `AUTHOR_COMM` tables for storing information on users and their communication, `TOPIC`, `POST`, and `ENTITY` tables for storing information on topics and posts, as well as named entities referenced in posts. The latter is required for discovering experts and trend setters among users. In addition to tables, several B-tree indexes are created to speed up query processing. During crawling and initial processing of Internet forum data, several obstacles may appear. First of all, automatic threaded crawler may overload the forum server, even when it conforms to the netiquette and follows `robots.txt` and `<META>` protocols. In such case, no new session can be created on the server. The crawler must be robust and must handle such situations. Secondly, many links are duplicated across the forum. Duplication helps humans to better navigate the forum, but is very troublesome for web crawlers. In particular, the links leading to the same document can differ, e.g., in parameter lists. In such case, the crawler can not recognize an already visited document, which, in turn, leads to parsing the same document several times. Lastly, many methods for forum analysis rely on the chronological order of topics and posts. Unfortunately, parsing the date is an extremely difficult task. There are no fixed formats for displaying dates and each date can be displayed differently depending on the language being used. In addition, some Internet forum engines display dates using relative descriptions, e.g. "yesterday" or "today". These differences must be accounted for when customizing the crawler for a particular forum.

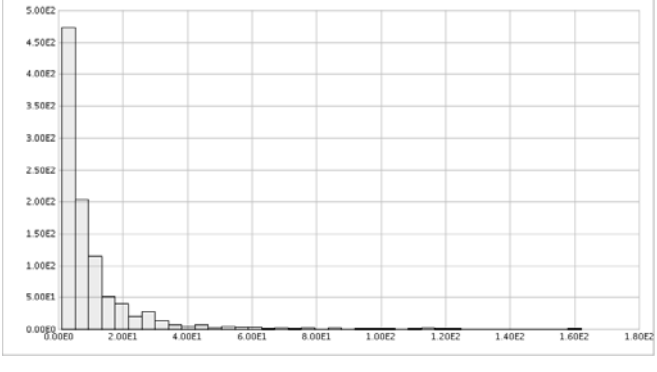



**Fig. 2.** Posts per topic

### 3.2 Statistical Analysis

A statistical analysis of an Internet forum consists in identifying basic building blocks for indexes. Basic statistics on topics, posts, and users are used to define activity, controversy, popularity, and other measures introduced in the next section. The analysis of these basic statistics provides great insight into the characteristics of Internet forums. Basic statistics can be computed during the

loading of an Internet forum into the database, or on demand upon the analysis of a forum. The latter technique is used for statistics that are time-bound, for instance, when calculating the activity of an Internet forum relative to the date of the analysis. In this section we present the results of the analysis of an exemplary Internet forum that gathers bicycle lovers. As of the day of the analysis the forum contained 1099 topics with 11595 posts and 2463 distinct contributors. Of course, the results presented below pertain to this specific forum and these statistics could vary among different forums, but the discussion of the importance and implications of each of these statistics is quite general and may be applied to a wide variety of Internet forums.

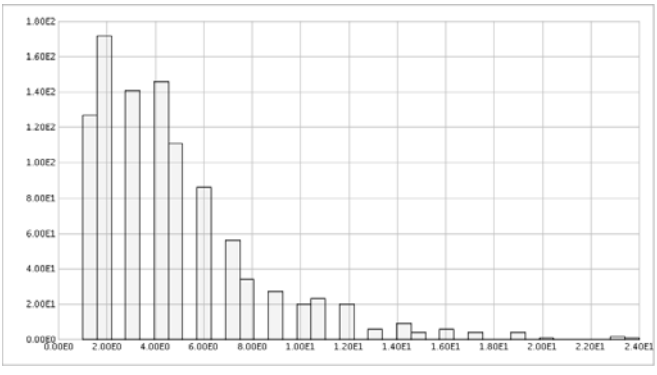



**Fig. 3.** Maximum depth

**Topic Statistics.** The most important factor in the analysis of Internet forums is the knowledge embedded in Internet forum topics. A variety of topics provides users with a wealth of information, but, at the same time, makes searching for particular knowledge difficult. The main aim of mining Internet forums is to provide users with automatic means of discovering useful knowledge from these vasts amounts of textual data. Below we present the basic statistics on topics gathered during the crawling and parsing phases.

Figure 2 presents the distribution of the number of posts per topic. Most topics contain a single post. This is either a question that has never been answered, or a post that did not spark any discussion. Posts leading to long heated discussions with many posts are very rare, and if a post generates a response, then continuing the discussion is not very likely. Almost every Internet forum has a small set of discussions that are very active (these are usually "sticky" topics). The biggest number of posts per topic can be generated by the most controversial posts that provoke heated disputes.

The distribution of the maximum depth for each topic is depicted in Figure 3. Topic depth may be computed only for Internet forums that allow for threaded discussions. Flat architectures, such as PhpBB, where each post is a direct answer to the previous post, do not allow to create deeply threaded discussions. The

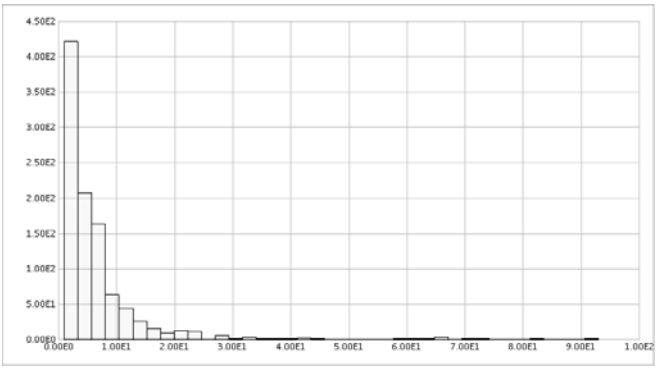



**Fig. 4.** Number of distinct users

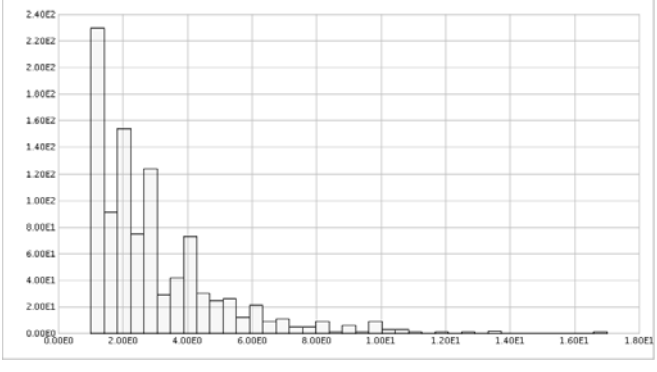



**Fig. 5.** Average number of posts per day

depth of a topic is a very good indicator of topic's controversy. Controversial topics usually result in long, deeply threaded discussions between small subsets of participants. From the figure follows that deeply threaded discussions are not frequent (although not negligible) and the majority of topics is either almost flat, or slightly threaded.

Another important statistic, depicted in Figure 4, concerns the number of distinct users who participate in and contribute to the topic. Most topics attract a small number of users. Sometimes, there is only one user posting to a topic (an example is a question that was answered by no one) or just two users (an example could be a question with a single answer). Some questions may encourage a dispute among experts, in such case a single question may generate a few conflicting answers from several users. Finally, certain topics stimulate many users to post, especially if the subject of the opening post, or some subsequent answers, are controversial. This statistic is useful when assessing the popularity and interestingness of a topic, under the assumption that interesting topics attract many users. This statistic can also be used to measure the controversy surrounding a

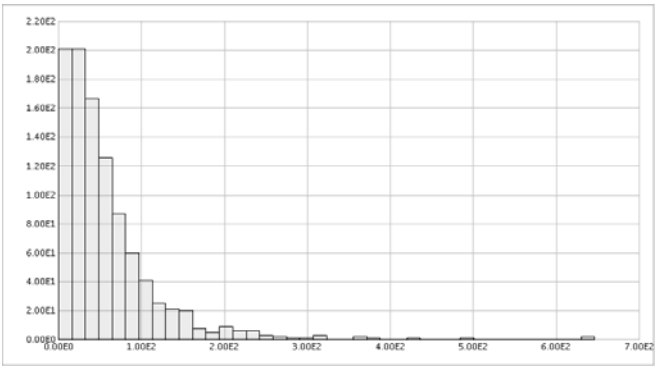



**Fig. 6.** Post length in words

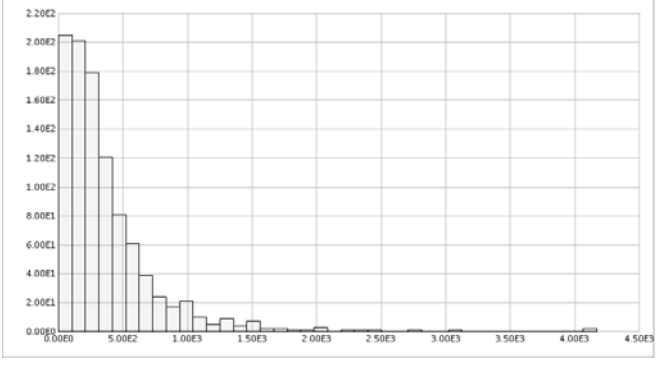



**Fig. 7.** Post length in characters

topic. If a topic is controversial, more users are likely to express their views and opinions on such topic. Combined with the analysis of the depth of the discussion, this statistic allows to quickly discover the most controversial topics.

Finally, for each topic a statistic on the average number of posts per day is collected. Figure 5 presents the distribution of the average number of posts per day. Most topics are not updated frequently, with the average number of posts ranging from 1 to 5, but there is also a significant number of hot topics that gather numerous submissions. If a topic concerns a recent development, e.g. a political event, many users are likely to share their thoughts and opinions. Also, some posts are labeled as urgent and the utility of an answer is directly related to the promptness of the answer.

**Post Statistics.** Interesting statistics can be gathered at the granularity level more detailed than a topic, namely, by analyzing individual posts. Posts may differ significantly by content, length, information value, etc. Our main goal is to derive as much knowledge as possible by analyzing only the structure of the

forum, and not its contents. Therefore, we deliberately refrain from using well-established methods of natural language processing and we use only the most elementary statistics.

Figures 6 and 7 present the distributions of post lengths measured in the number of words and characters, respectively. The shapes of the distributions are naturally very similar, but there are subtle differences. We choose to collect both statistics to account for the variability in vocabulary used in different forums. The language used by many Internet forum participants is a form of an Internet slang, full of abbreviations and acronyms. When a post is written using this type of language, then measuring the number of words is more appropriate to assess the information value of the post. On the other hand, forums that attract eloquent and educated people usually uphold high standards of linguistic correctness and measuring the information value of a post using the number of characters may be less biased.

**User Statistics.** Apart from statistically measuring topics and posts, we collect a fair amount of statistics describing the behavior of users. Users are the most important asset of every Internet forum, they provide knowledge and expertise, moderate discussions, and form the living backbone of the Internet forum community. The most interesting aspect of the Internet forum analysis is the clustering of users based on their social roles. Some users play the role of experts, answering questions and providing invaluable help. Other users play roles of visitors, newbies, or even trolls. Basic statistics gathered during downloading and parsing of an Internet forum provide building blocks that will further allow us to attribute certain roles to users.

The simplest measure of user activity and importance is the number of posts submitted by a user. Figure 8 presents the distribution of the number of posts per user. We clearly see that the overwhelming majority of users appears only once to post a single message, presumably a question. These users do not contribute to the forum, but benefit from the presence of experts who volunteer to

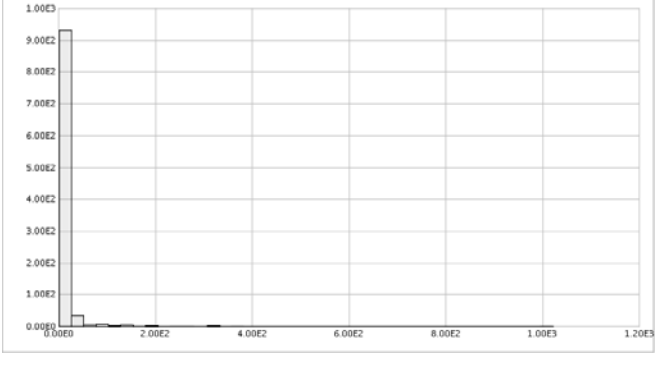



**Fig. 8.** Posts per user

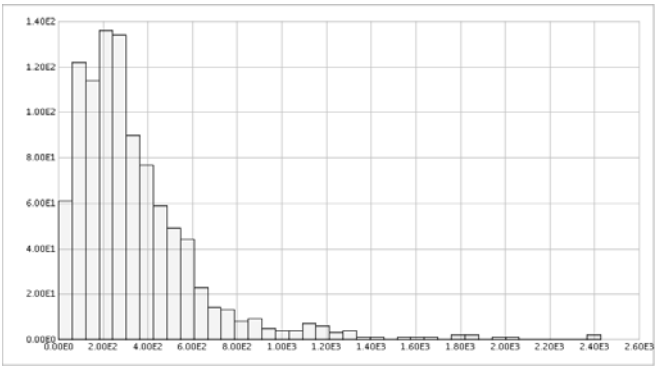



**Fig. 9.** Average post length in characters per user

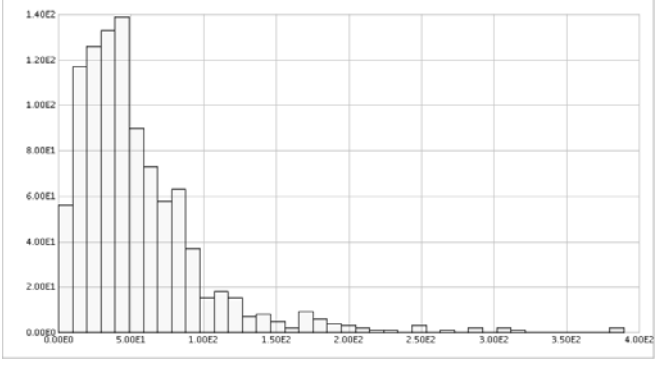



**Fig. 10.** Average post length in words per user

answer their questions. The distribution visible in Figure 8 is very characteristic of anonymous or semi-anonymous Internet forums (i.e., forums that allow to post messages either anonymously, or using a pseudonym, but without the requirement to register).

The number of posts created by a user may be somehow misleading. Consider two posts, the first one with a detailed description of how to solve a complex problem, and another post with a simple "thank you" message. Both posts equally contribute to the previous statistic. Therefore, we choose to include another statistic that measures the average length of a post per user expressed as the number of characters. Figure 9 presents the distribution of the average post length per user. Most posts are relatively short, rarely exceeding four sentences. The average English word length is 5.10 letters, and the average English sentence length is 14.3 words, which results in the average of 72.93 letters per sentence.

A very similar statistic is presented in Figure 10, where the histogram of the average post length in words is computed. Most users submit short posts, up to 50 words (again, this translates roughly into 5 sentences). Both above statistics

require a word of caution. It is very common for Internet forum posts to include quotations from other sources. Often, a user posting an answer to a query uses the text originating from another site to validate and endorse the answer. In such case, both statistics favor users who quote other material and make their submissions longer. On the other hand, identifying and removing quoted contents is very difficult, or even impossible. Other popular form of quotation consists in including a hyperlink to the quoted contents. Such post is much shorter, but interested users can follow the hyperlink to find relevant and useful information.

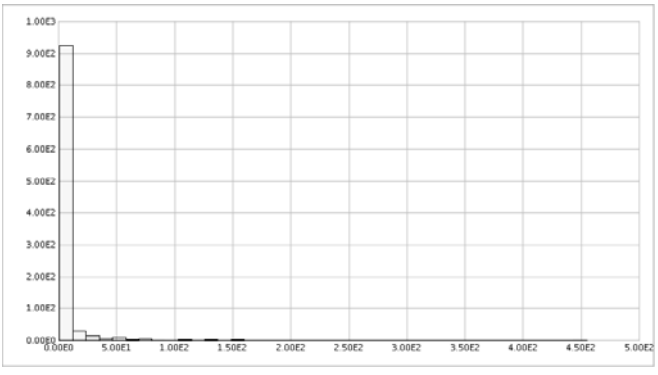



**Fig. 11.** Number of distinct topics per user

The final statistic considers the average number of topics in which a given user has participated. The rationale behind this statistic is twofold. First, it measures the versatility of a user. Users participating in many topics are usually capable of answering a broad spectrum of questions, and therefore can be perceived as experts. On the other hand, users who post questions to many topics are actively seeking for information and knowledge. Secondly, this statistic measures the commitment of a user. Users who participate in many topics contribute to the existence and vitality of the Internet forum community. As can be seen in Figure 11, most users participate in a single topic. Compare this result with the result presented in Figure 8, the community of Internet forum users is dominated by one-time visitors who post a question, receive an answer, and never come back to the Internet forum. Of course, all these statistics consider only active participants and do not consider consumers of information, who read but do not post.

Most of the distributions presented in this section resemble the Pareto distribution (also known as the Bradford distribution), a popular pattern emerging frequently in social, scientific, and many other observable phenomena, in particular, in Web analysis. The Pareto distribution shows exponentially diminishing probability $f(x)$ of a random variable $X$ to take larger values $x$. This distribution is used to describe the allocation of wealth among individuals (few own most, many own little), the sizes of human settlements (few large cities, many little villages), standardized price returns on individual stocks (few stocks bring huge returns, most stocks bring little returns), to name a few. The Pareto distribution is often

simplified and presented as the so-called Pareto principle of 80-20, which states that 20% of the population owns 80% of its wealth. To be more precise, Pareto distributions are continuous distributions, so we should be considering their discrete counterparts, the zeta distribution and the Zipf distribution. The reason we choose the Pareto distribution for comparison is simply the fact that this family of distributions has been widely popularized in many aspects of link analysis, e-commerce, and social network analysis, under the term of the *Long Tail*.

In October 2004 Chris Anderson, the editor-in-chief of Wired Magazine, first introduced the term *Long Tail* [2]. After highly acclaimed reception of the paper, Anderson presented his extended ideas in the book [3]. Although the findings were not new and the basic concept of a heavily skewed distribution has been studied by statisticians for years, the catch phrase quickly gained popularity and fame. The idea of the long tail is a straight adaptation of the Pareto distribution to the world of e-commerce and Web analysis. Many Internet businesses operate according to the long tail strategy. Low maintenance costs, combined with cheap distribution costs allow these businesses to realize significant profits from selling niche products. In a regular market the selection and buying pattern of the population results in a normal distribution curve. In contrast, the Internet reduces inventory and distribution costs, and, at the same time, offers huge availability of choices. In such environment, the selection and buying pattern of the population results in the Pareto distribution curve and the group of customers buying niche products is called the *Long Tail*[1]. The dominant 20% of products (called *hits* or *head*) is favored by the market over the remaining 80% of products (called *non-hits* or *long tail*), but the tail part is stronger and bigger than in traditional markets, making it easier for entrepreneurs to realize their profits within the long tail.

Interestingly, this popular pattern, so ubiquitous in e-commerce, manifests itself in Internet forums as well. The majority of topics is never continued, finishing after the first unanswered question. Most participants post only once never to return to the Internet forum. Almost always there is only one participant of a topic. All these observations provide us with a very unfavorable picture of Internet discussions. Indeed, discussions finish after the first post, posts are short, and users are not interested in participation. A vast majority of information contained in every forum is simply a useless rubbish. This result should not be dispiriting, on the contrary, it clearly shows that the ultimate aim of the Internet forum analysis and mining is the discovery of useful knowledge contained within interesting discussions hidden somewhere in the long tail.

### 3.3 Index Analysis

In this section we present selected measures for assessing the importance and quality of Internet forums. We refrain from creating a universal ranking among Internet forums and we do not try to derive a single unifying measure for all Internet forums. Instead, we construct several indexes which utilize the basic

[1] Sometimes the term *Long Tail* is used to describe these niche products, and not the customers. Other terms are also used to describe this phenomenon, e.g. Pareto tail, heavy tail, or power-law tail.

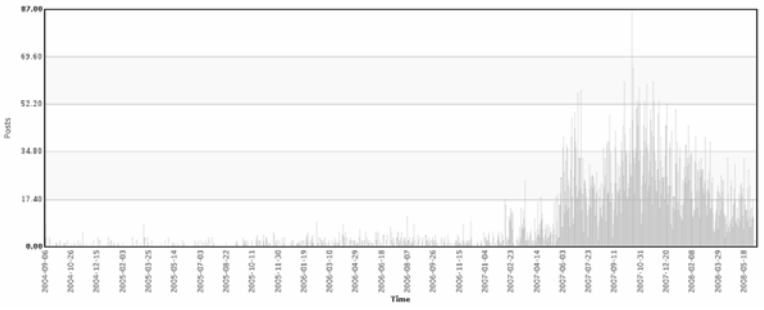



**Fig. 12.** Activity chart of a forum

statistical measures presented in the previous section. These indexes can be manipulated and validated by users, who subjectively rank forums, topics, and posts using our multi-criterion rankings.

Prior to define indexes used to rank individual topics, posts, and users, we introduce two additional measures that can be used to characterize the entire forum. These measures are related to the activeness of the forum. Figure 12 presents the activity of a forum (measured as the number of posts per day) since the creation of the forum. This figure allows to assess the liveliness of the forum and to follow the development of the forum, from its infancy, throughout adolescence and maturity, until its sunset years. In addition, the development trend of the forum can be deduced, as well as the stability of the forum and its dynamics.

Another interesting time-related measure is the chart of the weekly activity of the Internet forum, again, measured as the number of posts submitted daily. An example of such a chart is presented in Figure 13. The shape of this chart discloses much about the Internet forum community. Some Internet forums gather professionals and are used mainly to solve problems that occur during work. Such Internet forums have the main peak of their activity on working days from 9am to 4pm. Other Internet forums may serve as the meeting point for hobbyists, who discuss issues in their spare time, usually in the evenings and over weekends. After identifying the main type of the Internet forum (morning activity, weekend activity, evening and night activity, etc.) its participants and contributors may be additionally tagged based on this type.

One of the interesting aspects of the Internet forum analysis is the discovery of main subjects and themes of discussions. This knowledge cannot be mined from the structure of the social network, but must be determined from the text. On the other hand, the scalability requirements and the sheer volume of data exclude all techniques of advanced natural language processing. A reasonable compromise is to use only the most basic technique of the natural language processing, i.e., entity identification. An *entity* is a proper noun used during a discussion and referring to a name of a person, organization, location, product, acronym, or any other external being. Entities appearing frequently in a discussion represent popular subjects or notions. If an entity appears frequently within a given topic, it is most probably strongly related to the subject of the topic.

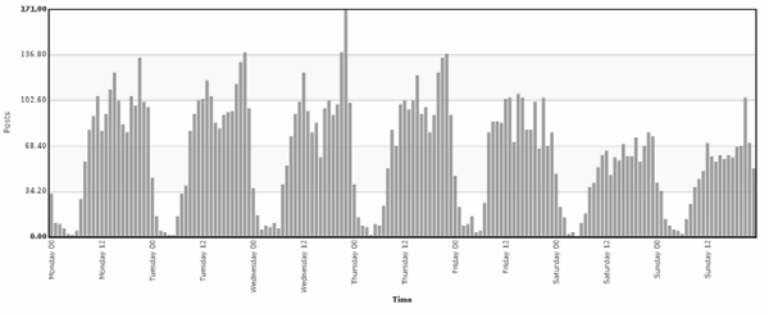



**Fig. 13.** Weekly activity chart of a forum

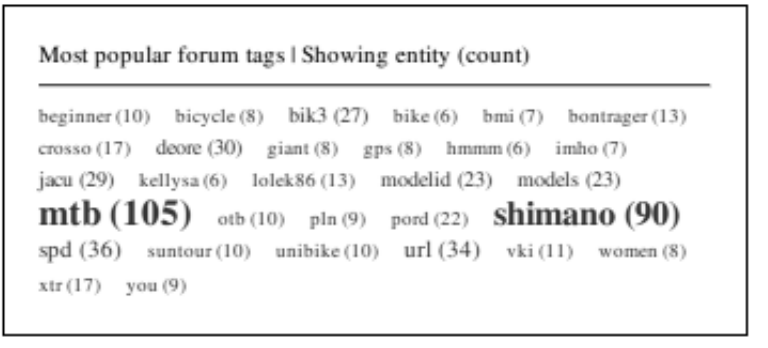



**Fig. 14.** Tag cloud

Users who enter many entities into the discussion (i.e., users who are first to use an entity in their post) are likely to be experts or trend-setters. Discovering entities can be very helpful in ranking topics and posts, but it can also be used to succinctly summarize discussions. In particular, identification of entities used in expert topics allows to learn products, brands, and keywords considered important in the context of the topic by experts. Such lists of entities can be invaluable guides for newcomers who can quickly identify leading brands and products in a given domain. Figure 14 presents the tag cloud derived from an Internet forum discussing bicycles. A tag cloud is simply a list of entities, where the size of each entity corresponds to the popularity of the entity. As expected, the most popular entities represent either brand names (*shimano*, *deore*), slang words (*spd*, *xtr*), or acronyms (*pln*, *gps*). There is also a small fraction of entities resulting from errors during the parsing phrase, these include loanwords (*bike*, *beginner*) or user pseudonyms (*jacu*, *kellysa*).

At the heart of every Internet forum lie discussion topics. The main emphasis of every analysis of Internet forums must focus on topics. It is the topic that attracts the activity and productivity of users. Since most Internet forums are very versatile, they contain topics that are active and passive, interesting and

useless, popular and interesting only to a small fraction of users. In other words, topics within a single Internet forum may be very different and a robust method of topic analysis is required to unearth interesting, intriguing and stimulating discussions. We begin by presenting indexes for topic analysis, then we proceed to indexes for user analysis, and we conclude with the presentation of indexes for post analysis.

Topics can be ranked according to a plethora of criteria. Below we present three exemplary indexes that can be constructed using the basic statistics introduced in Section 3.2.

1. **Activity.** Topic activity, similarly to Internet forum activity, is always defined in the context of a given time period. To help identify the most active topics that attract vivid discussions we define the Topic Activity Index (TAI) that measures the number of posts submitted during days preceding the date of the analysis. Posts are aged relatively to the date of the analysis, so the posts submitted earlier have lower weight. Formally, TAI is defined as $\text{TAI} = w_1 * P_{0,1} + w_2 * P_{1,2} + w_3 * P_{2,4} + w_4 * P_{4,10}$, where $P_{i,j}$ denotes the number of posts submitted to the topic between $i$ and $j$ days before the date of the analysis, and $w_i$ are arbitrary weights such that $\sum_i w_i = 1$. The coefficients were chosen empirically as the result of the analysis of several different Internet forums and set in the following way: $w_1 = 0.4$, $w_2 = 0.3$, $w_3 = 0.2$, $w_4 = 0.1$. We decide to remove from consideration posts older than 10 days. In addition, the TAI measure strongly favors topics that attract many submissions as of the time of the analysis, and prunes topics that were suspended, became quiet, or simply lost popularity. The last case is often seen on Internet forums discussing newest developments in politics, entertainment, and sports, where very popular topics appear instantly with the discussed event, and then quickly loose freshness and relevance.
2. **Popularity.** The Popularity index (PI) measures the overall popularity of a topic, outside of the context of the current analysis. Therefore, the popularity index of a topic is a monotonically increasing measure. PI is defined as $\text{PI} = w_1 * U + w_2 * P$, where $U$ denotes the number of users contributing to the topic, $P$ denotes the number of posts submitted to the topic, and $w_1, w_2$ are arbitrary weights such that $\sum_i w_i = 1$. We have found that the most reliable results can be achieved for $w_1$ significantly greater than $w_2$, e.g. $w_1 = 0.75$, $w_2 = 0.25$.
3. **Controversy.** The Controversy Index (CI) aims at identifying the most interesting and heated discussions. Controversy may result from two different reasons. Firstly, a topic may have been started by a controversial question or may touch an issue on which there are conflicting views within the community. Secondly, a topic may be fueled by trolling or flaming (i.e. posting intentionally abusive and conflicting posts with the sole purpose of irritating and annoying other members of the community). Of course, the aim of the analysis is to identify the first type of the controversy present in topic posts. The main difficulty in designing of the Controversy Index is a very high subjectivity of what should be considered controversial. Users vary

significantly in their tolerance to emotional language, cursing, or critically attacking sensitive issues. Other users take such posts very personally and do not tolerate any controversy. One good marker of the type of the controversy is the depth of the discussion tree. Users, who feel offended by a controversial post, tend to express their contempt but, usually, they do not continue the discussion. On the other hand, if the controversy stems from the natural disagreement on a given issue, then participants are far more likely to continue the discussion with the aim of convincing their adversary. In addition to measuring the depth of the discussion, we also measure the number of distinct users who submitted posts below a given discussion depth threshold. This additional measure allows to decide whether the discussion was a heated exchange of opinions between two quarreling users, or the subject was interesting for a broader audience. Combining these two numbers allows to prune discussions where only two users participate and the community is indifferent to the issue. The last building block of the CI is the emotionality of posts submitted to the topic. We define the emotionality measure a little further when discussing measures for individual post ranking. Suffice to say, high emotionality characterizes posts that either contain emotional words, or their punctuation indicates strong emotions. Formally, the CI is defined as $\mathrm{CI} = w_1 * avg\,(E) + w_2 * U + w_3 * W$, where $avg\,(E)$ denotes the average emotionality of a post submitted to the topic, $U$ denotes the number of distinct contributors who have passed the topic depth threshold, $W$ denotes the number of posts, and $w_1, w_2, w_3$ are arbitrary weights such that $\sum_i w_i = 1$. In our experiments we have found that the following values of weights produce high quality results: $w_1 = 0.5$, $w_2 = 0.375$, and $w_3 = 0.125$.

Figure 15 presents an exemplary ranking of topics ordered by their Activity Index. The ranking is computed from the Internet forum devoted to banks, investment trusts, mortgages, and other financial tools. As can be easily seen, the most active topics concern clearing of a credit card, consolidation of credits, and a recently advertised checking account in a particular bank.


**Fig. 15.** Activity Index ranking of topics


**Fig. 16.** Controversy Index ranking of topics

The Controversy Index ranking of posts for the same Internet forum is presented in Figure 16. One instantly notices that the Controversy Index favors a different type of topics. Almost all topics in the Top 10 are complaints, warnings, and accounts of embarrassing customer treatment. There are topics openly accusing banks of fraud and cheating, topics warning against dishonest banks, and topics discussing controversial offers (like the discussions on the Cetelem Bank, the leading provider of credit cards for chain super-stores).

Another dimension of the Internet forum mining is the analysis of users. As in case of topics, users may be ranked according to several different, and sometimes conflicting, criteria. Below we present two simple indexes that can be used to rank Internet forum users by employing basic statistics introduced in Section 3.2.

1. **Activity.** User activity may be measured primarily by the number of posts submitted to the forum, but there is a subtle difference between submitting ten posts to a single topic, and submitting one post to ten topics. Users who participate in many discussions and who post to different topics or maintain a high average of posts per day are likely to be the most valuable users of the Internet forum. Participation in several topics signals not only the versatility of the user, but her commitment to the Internet forum community. The Activity Index for Users (AIU) considers both the number of posts submitted by the user, and the number of distinct topics in which the user has participated. Formally, the AIU is defined as $AIU = w_1 * T + w_2 * W$, where $T$ denotes the number of topics in which the user took part, $W$ is the number of posts submitted by the user, and $w_1, w_2$ are arbitrary weights such that $\sum_i w_i = 1$. We strongly suggest that $w_1 > w_2$, which allows to prune users who post selectively, but their activity is limited to a small number of topics.
2. **Productivity.** The Productivity Index (PI) of a user captures the efficiency of the user in passing knowledge. This index computes the total length of all posts submitted by the user. However, the PI can be misleading, because it favors users who write long posts, but severely punishes users who reply by pointing to external resources (FAQs, wikis, etc.) providing the URL of the external resource. In such case, the user may provide valid and valuable

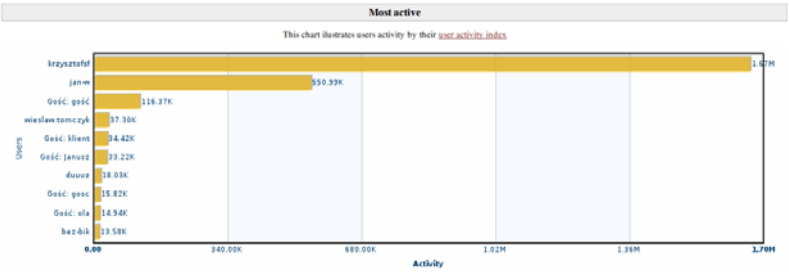



**Fig. 17.** Activity Index ranking of users

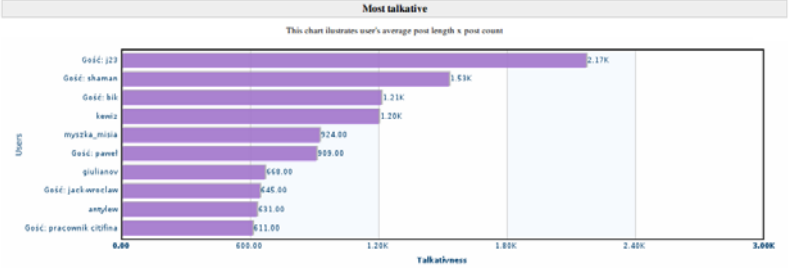



**Fig. 18.** Productivity Index ranking of users

knowledge by passing a relatively short text. On the other hand, Figure 6 shows that the majority of posts consist only of one or two words, which are either pointers to external resources, links leading to unrelated sites as the result of spamming, or useless text such as "thank you", "wow!", and the like. The Productivity Index helps to sieve through these short texts at the cost of ignoring some valuable submissions.

Figure 17 presents the ranking of users according to the Activity Index for Users (AIU). We see that two users, `krzysztofsf` and `jan-w`, stand out among the crowd. As we shall see in Section 3.4 , these two users form strong social groups and serve the community as experts, answering many questions from newbie users.

The ranking of users based on the Productivity Index is presented in Figure 18. Interestingly, in the Top 10 of the most productive users there is no one from the Top 10 of the most active users. This result is slightly counter-intuitive, because we would rather expect the most active users to submit longer posts. Quite the contrary, the most productive user `j23` has written only 9 posts on one topic, and the second most productive user `Guest:shaman` has written 8 posts.

The final dimension of Internet forum mining is the analysis of individual posts. As we have mentioned earlier, for each post we collect all possible statistics, such as the length of the post in characters, the number of words in the post, the number of entities in the post, and the post's depth in the topic tree (where

applicable). Below we present one exemplary index that can be constructed from these basic statistics.

1. **Emotionality.** The Emotionality Index (EI) serves at least three different purposes. Firstly, it allows to assess the temperature of the discussion. Secondly, it shows the mutual relationships between users. Finally, it may be used to compute the degree of controversy around a given topic. Unfortunately, similarly to the controversy, individual perception of emotionality in a post is a highly subjective matter. We use a method similar to the method presented in [4]. The emotionality of a post is estimated using two factors. The first factor is the number of words in the post that bear strong emotional weight. We find these words using a predefined vocabulary of emotional words and the semantic lexicon WordNet. The second factor utilizes emoticons and punctuation used in the text. We are looking for dots, commas, and combinations of special characters, known as emoticons. The presence of certain emoticons in the text may very well account for certain emotional state of writing.

### 3.4 Network Analysis

In Section 2 we have introduced the notion of a social network. Let us recall that a social network is a structure made of entities that are connected by one or more types of interdependency. Entities constituting a social network represent individuals, groups or services, and relationships between entities reflect real-world dependencies. Social networks are best represented by sociograms, which are graphic representations of social links connecting individuals within the network. Nodes in a sociogram represent individuals, and edges connecting nodes represent relationships. Edges can be directed (e.g., a relationship of professional subordination), undirected (e.g., a relationship of acquaintance), one-directional (e.g., a relationship of trust), and bi-directional (e.g., a relationship of discussion). Sociograms are the main tool used in sociometry, a quantitative method of measuring various features of social links.

In order to compute the measures of social importance and coherence of Internet forums, we must first create a model of a social network for Internet forums. When developing a model of a social network for a given domain, we must carefully design the sociogram for the domain: what constitutes nodes and edges of the sociogram, are there any weights associated with edges, and whether edges are directed or undirected. Let us first consider the choice of nodes, and then to proceed to the design of edges.

**Model of Internet Forum Sociogram.** The participation in an Internet forum is tantamount to the participation in an established social community defined by the Internet forum subject. The degree of coherence of the community may vary from very strict (a closed group of experts who know each other), through moderate (a semi-opened group consisting of a core of experts and

a cloud of visitors), to loose (fully opened group of casual contributors who participate sporadically in selected topics). The degree of coherence informs about information value of the forum. Opened forums are least likely to contain interesting and valuable knowledge content. These forums are dominated by random visitors, and sometimes attract a small group of habitual guests who tend to come back to the forum on a regular basis. Discussions on opened forums are often shallow, emotional, inconsistent, lacking discipline and manners. Opened forums rarely contain useful practical knowledge or specialized information. On the other hand, opened forums are the best place to analyze controversy, emotionality, and social interactions between participants of the discussion. Their spontaneous and impulsive character encourages users to form their opinions openly, so opened forums may be perceived as the main source of information about attitudes and beliefs of John Q Public. On the opposite side lie closed specialized forums. These forums provide high quality knowledge on selected subject, they are characterized by discipline, consistency, and credibility. Users are almost always well known to the community, random guests are very rare, and users pay attention to maintain their status within the community by providing reliable answers to submitted questions. Closed forums account for a small fraction of the available Internet forums. The majority of forums are semi-opened forums that allow both registered and anonymous submissions. Such forums may be devoted to a narrow subject, but may also consider a broad range of topics. Usually, such forum attracts a group of dedicated users, who form the core of the community, but casual users are also welcomed. These forums are a compromise between the strictly closed specialized forums and the totally opened forums. One may dig such forum in search of practical information, or browse through the forum with no particular search criterion.

Our first assumption behind the sociogram of the social network formed around the Internet forum concerns users. We decide to consider only regular users as the members of the social network. Casual visitors, who submit a single question and never return to the forum, are marked as outliers and do not form nodes in the sociogram. This assumption is perfectly valid and reasonable, as casual users do not contribute to the information contents of the forum and provide no additional value to the forum. The threshold for considering a given user to be a regular user depends on the chosen forum and may be defined using the number of submitted posts and the frequency of posting. The second assumption used during the construction of the sociogram is that edges in the sociogram are created on the basis of participation in the same discussion within a single topic. Again, this assumption is natural in the domain of Internet forums. The core functionality of the Internet forum is to allow users to discuss and exchange views, opinions, and remarks. Therefore, the relationships mirrored in the sociogram must reflect real-world relationships between users. These relationships, in turn, result from discussing similar topics. The more frequent the exchange of opinions between two users, the stronger the relationship binding these users. Of course, the nature of this relationship may be diverse. If two users frequently exchange opinions, it may signify an antagonism, contrariness, and dislike, but

it may also be used to reflect strong interaction between users. In our model the nature of the relationship between two users is reflected in the type of the edge connecting these two users in the sociogram: if the edge is bi-directional, then it represents a conflict, if the edge is one-directional, then it represents a follow-up (usually an answer to a question), and if the edge is undirected, then the nature of the relationship cannot be determined. The final element of the sociogram is the computation of edge weights. In a more sophisticated model the weight of an edge could represent the emotionality of the relationship (e.g., friendliness, enmity, or indifference). Such emotionality could be determined by analyzing posts and computing their emotionality. Unfortunately, this would require the employment of natural language processing techniques to analyze not only the structure, but the semantics of posts as well. In this research we constrained ourselves to analyzing the structure of the social network only, therefore, we postpone this interesting research direction until further. For the time being weights of edges represent the number of posts exchanged between users.

The definition of the participation in the same discussion requires a few words of explanation. Many Internet forum engines allow for threaded discussions, where each post can be directed as the reply to a particular previous post. In the case of such engines the entire topic can be drawn as a tree structure with a single initial post in the root of the tree, and all subsequent posts forming branches and leaves of the tree. With threaded Internet forum engines we may distinguish between participating in the same topic, participating in the same thread of the discussion (i.e., posting in the same branch of the discussion), and direct communication (i.e., replying directly to a post). A well-balanced tree of discussion represents an even and steady flow of the discussion, whereas a strongly unbalanced tree represents a heated discussion characterized by frequent exchange of posts. Unfortunately, most Internet forum engines do not allow for threading. Usually, every post is appended to the sequential list of posts ordered chronologically. Users, who want to reply to a post other than the last one, often quote the original post, or the parts thereof. Due to message formatting and different quoting styles, determining the true structure of such flat Internet forum is very difficult, if impossible. In our model we have assumed that in the case of flat forums, where no threading is available, each post is the reply to the precedent post. This somehow simplistic assumption may introduce a slight bias during the analysis, but our empirical observations justify such assumption. In addition, imposing virtual threads onto flat forum structure allows to compute the depth of a submission as one of the basic statistics. The depth of a post is computed using a sliding window technique with the width of 5 subsequent posts (the threshold has been set up experimentally). For each post, we are looking for another post submitted by the same author within the last five posts. If such post is encountered, the depth of the current post is increased, otherwise we treat the post as the new branch of the discussion. Table 6 presents an example of such virtual thread derived from the flat forum structure.

**Table 6.** Example of a virtual thread (forum.probasket.pl)

| | user | depth (references) |
|---|---|---|
| 1 | Redman | 1 (*null*) |
| 2 | Small | 1 (*null*) |
| 3 | Redman | 2 (# 1) |
| 4 | Small | 2 (# 2) |
| 5 | Redman | 3 (# 3) |
| 6 | Londer | 1 (*null*) |
| 7 | Small | 3 (# 4) |
| 8 | Londer | 2 (# 6) |
| 9 | Redman | 4 (# 5) |
| 10 | Londer | 3 (# 8) |
| 11 | Nameno | 1 (*null*) |
| 12 | Londer | 4 (# 10) |
| 13 | Redman | 5 (# 9) |
| 14 | Small | 1 (*null*) |
| 15 | Nameno | 2 (# 11) |

**Topic Analysis.** The social network built on top of the Internet forum community accounts for the following types of users:

- *key users* who are placed in the center of the discussion,
- *casual users* who appear on the outskirts of the network,
- *commenting users* who answer many questions, but receive few replies,
- *hot users* who receive many answers from many other users (e.g., authors of controversial or provoking posts).

The above-mentioned types of users are clearly visible from the shape of the social network. Figure 19 presents an example of a social network derived from the Internet forum on bicycles. Weights of edges represent the number of posts exchanged between users represented by respective nodes. For clarity, only the strongest edges are drawn on the sociogram. We can clearly see small isolated groups consisting of two or three users in the left-hand side of the sociogram. The number of posts exchanged between users and isolation from other users suggest, that these nodes represent a long dispute between the users, most often, being the result of a controversial post. We also see a central cluster of strongly interconnected users visible in the right-hand side of the sociogram. Within the cluster a few nodes tend to collect more edges, but there is no clear central node in this network. Interestingly, most edges in the cluster are bi-directional, which implies a balanced and popular discussion, where multiple users are involved.

Another type of a sociogram is presented in Figure 20. The Internet forum, for which the sociogram is computed, is devoted to banks, stock exchange, and investment funds. The central and the most important node in the sociogram is `krzysztofsf`. This user always answers and never asks questions or initializes a topic. Clearly, this user is an expert providing answers and expertise to other

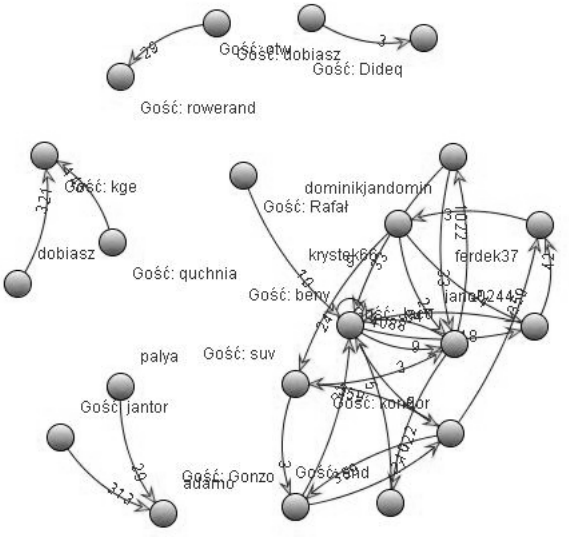



**Fig. 19.** Sociogram for the forum on bicycles

members of the community. In particular, observe the weight of the edge connecting `krzysztofsf` to `Gość:gość` (which denotes an anonymous login). This single expert has posted 2652 replies to questions asked by casual visitors! Another very interesting formation is visible to the bottom of the figure. There is a linked list of users connected mostly by one-directional edges and isolated from the main cluster. We suspect that this formation denotes a small community within the Internet forum community. It may be an openly acknowledged group of users, but it may also be an informal group that continues their discussions on very narrowly defined subjects.

**User Analysis.** Apart from analyzing the social network of users participating in a given forum or topic, we may also want to analyze individual users in terms of their global relationships. The sociogram centered on a particular node is called an egocentric graph and it can be used to discover the activity of the node, the nature of the communication with other nodes, and thus, to attribute a given social role to the node [27]. The egocentric graph for a given user consists of the node representing the user, the nodes directly connected to the central node, and all edges between nodes included in the egocentric graph. Figure 21 presents the egocentric graph for the user `wieslaw.tomczyk`. We clearly see a star pattern, where the node in the center connects radially by one-directional edges with multiple nodes, and those nodes are not connected by edges. This pattern is characteristic of experts who answer many questions, and users who ask questions do not form any relationships (usually, these are casual users who seek an advice on a particular subject).

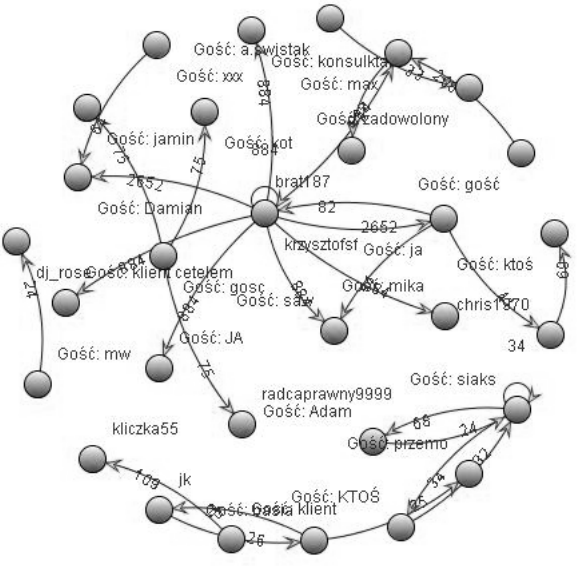



**Fig. 20.** Sociogram for the forum on banks

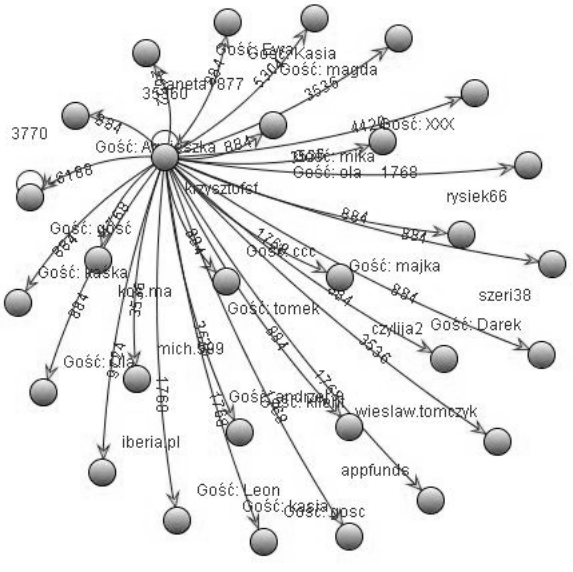



**Fig. 21.** Egocentric graph for the user `wieslaw.tomczyk`

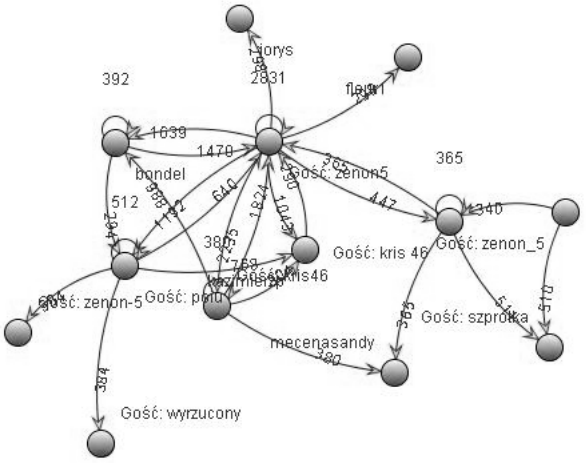



**Fig. 22.** Egocentric graph for the user `kris 46`

A very different egocentric graph is presented in Figure 22. Here, the user `kris 46` belongs to a small and strongly tied community consisting of five more users forming almost a clique. Apart from the core group including users `kazimierzp`, `polu`, `bondel`, and `zenon5`, user `kris 46` occasionally communicates with a few other users, who lie outside of the core group. This cloud structure consisting of a densely connected core and loosely connected outlier nodes is characteristic for users who participate in the forum community for a longer period of time. This long participation allows them to form substructures within the community that harden their commitment to the community.

**Role Analysis.** One of the most interesting and challenging problems in mining Internet forum communities is the discovery and attribution of social roles in the social network of users [13,27]. Social roles may be statically attributed to users, or may be dynamically assigned to users for each discussion. The latter solution is more flexible, because it accounts for the situation where a user may act as an expert on one topic, and a commenter on another topic. For the sake of simplicity we assume the static attribution of social roles to users.

Many different social roles may be derived from the social network of Internet forum users. Every role should be distinct from other roles and identifiable from the structure of the social network only, i.e., the identification of the social role for a given user must not require the semantic analysis of posts submitted by the user. Below we present an exemplary classification of social roles:

- **newbie:** a user who asks a few questions and then disappears from the community, very easy to discover because her egocentric graph is empty,
- **observer:** similar to a newbie, but participates in the community on the regular basis, rarely posts, her egocentric graph is sparse,

- **expert:** a comprehensive user with the high authority, does not ask questions, participates in discussions on multiple topics, the egocentric graph follows the star pattern,
- **commentator:** a comprehensive user, answers many questions, often follows an expert and adds comments and remarks, similar to an expert, but the average length of posts is much shorter,
- **troll:** a provoking and irritating user, initiates many discussions characterized by the high controversy and temperature, the egocentric graph often follows the inverted star pattern (many users answer the troll).

Of course, social role identification serves a more important goal than just tagging users. For a closed specialized forum identifying experts is crucial for interacting with knowledge contents hidden within the Internet forum. One may quickly rank users by their authority and focus on reading posts written by experts. Another possibility is an automatic knowledge acquisition, where posts submitted by experts may be retrieved and parsed in search for named entity references. For common opened forums one may want to identify trolls in order to create spam filters for the forum. Usually, discussions stoked by trolls bear little knowledge contents and following these discussions is a waste of time. The identification of social roles based solely on the shape of the egocentric graph for a given user is difficult and error-prone. Additional statistics, such as the statistics described in Section 3.2, are useful to improve the precision and recall of social role attribution. For instance, in order to identify an expert we may consider the following basic statistics: the number of distinct topics with user submissions (must be large), the depth of the discussion following an expert's post (expert opinions tend to close the discussion and do not spark long disputes), the average length of a post (moderate, neither too long nor too short). Similar additional basic statistics can be derived for other social roles.

## 4 Related Work

The data acquired from the Web has its own distinct properties and characteristics that make these data particularly difficult to mine. Web mining has recently emerged as a new field of research, which copes with the task of designing efficient and robust algorithms for mining Internet data. A very good introduction to Web mining methods and models can be found in [5]. Much research has been conducted on text mining and knowledge discovery from unstructured data. An interested reader may find a detailed summary of recent findings in this domain in [12] and [26]. In addition, much work has been done on statistical natural language processing. Statistical methods for text mining are described and discussed in detail in [17].

Analysis of threaded conversations, which are the predominant pattern of communications in the contemporary Web, is an actively researched domain [15,22]. In particular, many proposals have been submitted to derive social roles solely based on the structural patterns of conversations. Examples of earlier proposals include [13,25,31].

A thorough overview of structural patterns associated with particular social roles, that can be used as structural signatures, can be found in [23]. Most of the recent work has been performed on the basis of social network analysis methods [7,11,16], but the investigation of role typology has been an important challenge in sociology [19,21]. Recently, more attention has been given to the identification of social roles that are not general, but specific to online communities. The existence of local experts, trolls, answer people, fans, conversationalists, etc. has been verified [9,14,18,24]. Moreover, the benefits of being able to deduce the social role of an individual without having to analyze the contents generated by that individual are becoming apparent [28,29,30].

## 5 Conclusions

In this chapter we have presented the methodology of mining the most popular Internet discussion engine – Internet forums. First, we have described the non-trivial process of data acquisition by crawling Internet forums. Then, we have described the three levels of Internet forum analysis: the statistical analysis, the index analysis, and the network analysis. On each level, interesting knowledge can be discovered and presented to the user. We have designed a few indexes that allow users to rank forum topics, posts, and users, according to a plethora of criteria: the most controversial, the most active, the most popular, etc. Finally, we have discussed the process of modeling a forum as a social network linking users through discussions. We have shown how analysis of such a social network allows to discover social roles of users, and, in result, to filter interesting knowledge from huge volumes of posts.

The abundance of Internet forums, ranging from specialized to popular, makes the subject of mining Internet forums both interesting and very desirable. Internet forums hide enormous amounts of high quality knowledge generated by immense communities of users. Unfortunately, the lack of structure and standards makes the acquisition of this knowledge very difficult. Research presented in this chapter is a step towards automatic knowledge extraction from these opened repositories of knowledge. Our statistics, heuristics and indexes are fairly simple, but work surprisingly well in the real world. We have found that our prototype generated high quality rankings of topics and users for a wide variety of Internet forums.

The results of the research presented in this chapter reach beyond the interests of the academic community. The ability to mine knowledge hidden in Internet forums, to discover emerging trends and fashions, to compute social reception of brands and products, all these are of extreme interest to the marketing industry. Pollsters, advertisers, and media monitors are among those who may profit from the development of the presented technology. We intend to continue this initial research into the topic of mining Internet forums and extend it in several directions. Our future work agenda includes, among others, the investigation into the Internet forum evolution, the analysis of macro- and micro-measures pertaining to the social network of Internet forum users, and further examination of social roles. We also plan to enrich our prototype with the ability to perform selected text mining techniques during data acquisition.

## References


1. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules in Large Databases. In: 20th International Conference on Very Large Data Bases, Santiago de Chile, Chile, pp. 487–499. Morgan Kaufmann, San Francisco (1994)
2. Anderson, C.: The long tail. Wired (October 2004)
3. Anderson, C.: The Long Tail: Why the Future of Business Is Selling Less of More. Hyperion (2006)
4. Alm, C.O., Roth, D., Sproat, R.: Emotions from Text: Machine Learning for Text-based Emotion Prediction. In: Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, pp. 579–586. Association for Computational Linguistics, Vancouver (2005)
5. Chakrabarti, S.: Mining the Web: Discovering Knowledge from Hypertext Data. Morgan Kaufmann, San Francisco (2002)
6. Barabasi, A.L., Bonabeau, E.: Scale-free networks. Scientific American (May 2003)
7. Brandes, U., Erlebach, T.: Network Analysis, Methodological Foundations. Springer, Heidelberg (2005)
8. Breslin, J.G., Kass, R., Bojars, U.: The Boardscape: Creating a Super Social Network of Message Boards. In: International Conference on Weblogs and Social Media ICWSM 2007, Boulder, Colorado, USA (2007)
9. Burkharter, B., Smith, M.: Inhabitant's Uses and Reactions to Usenet Social Accounting Data. In: Snowdon, D., Churchill, E.F., Frecon, E. (eds.) Inhabited Information Spaces, pp. 291–305. Springer, Heidelberg (2004)
10. Internet world stats, `http://www.internetworldstats.com`
11. Carrington, P.J., Scott, J., Wasserman, S.: Models and Methods in Social Network Analysis. Cambridge University Press, Cambridge (2005)
12. Feldman, R., Sanger, J.: The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data. Cambridge University Press, Cambridge (2006)
13. Fisher, D., Smith, M., Welser, H.T.: You Are Who You Talk To: Detecting Roles in Usenet Newsgroups. In: 39th Annual Hawaii International Conference on System Sciences. IEEE Computer Society, Kauai (2006)
14. Golder, S.A.: A Typology of Social Roles in Usenet. A thesis submitted to the Department of Linguistics. Harvard University, Cambridge (2003)
15. Gomez, V., Kaltenbrunner, A., Lopez, V.: Statistical analysis of the social network and discussion threads in Slashdot. In: 17th International Conference on World Wide Web (WWW 2008). ACM Press, Beijing (2008)
16. Hanneman, R., Riddle, M.: Introduction to social network methods. University of California, Riverside (2005)
17. Manning, C.D., Schuetze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge (1999)
18. Marcoccia, M.: On-line Polylogues: Conversation Structure and Participation Framework in Internet Newsgroups. Journal of Pragmatics 36(1), 115–145 (2004)
19. Merton, R.K.: Social Theory and Social Structure. Free Press, New York (1968)
20. O'Reilly, T.: O'reilly radar: Web 2.0 compact definition: Trying again (April 2006), `http://radar.oreilly.com/archives/2006/12/web_20_compact.html`
21. Parsons, T.: The Social System. Routledge & Kegan Paul Ltd., London (1951)
22. Shi, X., Zhu, J., Cai, R., Zhang, L.: User grouping behavior in online forums. In: 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2009). ACM Press, Paris (2009)

23. Skvoretz, J., Faust, K.: Relations, Species, and Network Structure. Journal of Social Structure 3(3) (2002)
24. Turner, T.C., Smith, M., Fisher, D., Welser, H.T.: Picturing Usenet: Mapping Computer-Mediated Collective Action. Journal of Computer-Mediated Communication 10(4) (2005)
25. Viegas, F.B., Smith, M.: Newsgroup Crowds and AuthorLines: Visualizing the Activity of Individuals in Conversational Cyberspaces. In: 37th Annual Hawaii International Conference on System Sciences (HICSS 2004) - Track 4. IEEE Computer Society, Kauai (2004)
26. Weiss, S., Indurkhya, N., Zhang, T., Damerau, F.: Text Mining: Predictive Methods for Analyzing Unstructured Information. Springer, Heidelberg (2004)
27. Welser, H.T., Gleave, E., Fisher, D., Smith, M.: Visualizing the Signatures of Social Roles in Online Discussion Groups. Journal of Social Structure 8(2) (2007)
28. Wenger, E.: Communities of Practice: Learning, Meaning, and Identity. Cambridge University Press, Cambridge (1999)
29. Wenger, E., Snyder, M.: Communities of Practice: The Organizational Frontier. Harvard Business Review, 139–145 (2000)
30. Wenger, E., McDermott, R., Snyder, W.S.: Cultivating Communities of Practice: A Guide to Managing Knowledge. Harvard Business School Press, Boston (2002)
31. White, H.C., Boorman, S.A., Breiger, R.L.: Social-Structure from Multiple Networks: 1. Blockmodels of Roles and Positions. American Journal of Sociology 81(4), 730–780 (1976)

# Chapter 12
# Data Mining for Information Literacy

Bettina Berendt

Department of Computer Science, K.U. Leuven, B-3001 Heverlee, Belgium
http://www.cs.kuleuven.be/~berendt

**Abstract.** This paper argues for extending the scope of applying data mining towards making it a means to help people better understand, reflect and influence the information and information-producing and -consuming activities that they are surrounded by in today's knowledge societies. Data mining is thereby seen as a means to furthering information literacy and specifically critical literacy. We discuss and extend classical definitions of these two constructs and derive *critical data literacy* and *privacy literacy* as two essential new key sub-competences. We describe an analysis framework for concrete efforts to use data mining in this way, structuring by techniques and objects and characterising by how principles of successful learning are supported. We then analyse a number of examples of current Web-based tools within this framework, investigating how they can further critical data literacy and privacy literacy. We conclude with an outlook on next steps in the proposed new field of Data Mining for Information Literacy.

**Keywords:** Web mining, text mining, knowledge discovery, learning, information literacy, privacy.

## 1 Introduction

When one thinks of Data Mining, it is usually as a family of automated techniques for sifting through huge amounts of data in order to find "nuggets" of knowledge. These can be deployed directly in an application such as the buying recommendations in online bookstores or the ranking of results in Web search engines [32]. Alternatively, they are used to support managements via business analytics, for example to identify high-value customers or detect credit-card fraud [11]. Governments may use data mining for identifying suspicious individuals [54].

So who are the primary beneficiaries of Data Mining (DM)? In the business-analytics and national-security application areas, they are businesses and governments, while (some) customers and citizens profit from increased efficiency and security. In business models where DM "is" the product (such as recommender systems or search engines), there is also a rather straightforward advantage for customers/users in terms of functionality. But can we go beyond this view that focuses on people as customers or advertising consumers? The purpose of this paper is to argue that data mining can do much more, that it can be a highly

D.E. Holmes, L.C. Jain (Eds.): Data Mining: Found. & Intell. Paradigms, ISRL 25, pp. 265–297.
springerlink.com 

effective means for making people aware of the nature, potential and limitations of the information they are surrounded by – and that they themselves constantly produce; that this has important social and democratic aspects; thus, in short, that data mining can serve as a tool for furthering information literacy.

Much has been said about the potentials for education and democracy of the Internet and specifically the Web, e.g. [26]. When one of the goals is democracy, is it enough for a society to make the Internet widely and cheaply available, to teach school children (and interested citizens) how to use it, to install e-Government services, allow for online petitions, and allow free speech in blogging sites? Arguably, such a mainly technological approach is not sufficient; democracy also needs a culture of democratic behaviour *in the informational environment of the respective society*. But what does this mean? Along with Shapiro and Hughes [52], we ask:

> "What does a person need to know today to be a full-fledged, competent and literate member of the information society? As we witness not only the saturation of our daily lives with information organized and transmitted via information technology, but the way in which public issues and social life increasingly are affected by information-technology issues – from intellectual property to privacy and the structure of work to entertainment, art and fantasy life – the issue of what it means to be information-literate becomes more acute for our whole society. Should everyone take a course in creating a Web page, computer programming, TCP/IP protocols or multimedia authoring? Or are we looking at a broader and deeper challenge – to rethink our entire educational curriculum in terms of information?"

Based on these questions, the authors propose to re-define the long-used concept of *information literacy* to extend beyond the narrow notion of knowing how to use a library or search on the Web.

> "Or is [information literacy] [...] something broader, something that enables individuals not only to use information and information technology effectively and adapt to their constant changes but also to think critically about the entire information enterprise and information society? Something more akin to a 'liberal art' – knowledge that is part of what it means to be a free person in the present historical context of the dawn of the information age?"

This notion of information literacy is closely linked to education and its main goal in the sense of Dewey [17, pp. 76ff.]: increasing the ability to perceive and act on meaning in one's society, advancing students' ability to understand, articulate, and act democratically in their social experience. We will therefore base the remainder of this article on pertinent background in educational studies (Section 2), propose a framework derived from these to structure and characterize concrete tools (Section 3), and use this framework to analyze a number of data-mining and other data collection and analysis efforts as examples of "data mining for information literacy" (Section 4). These are, for reasons of space, exemplary and not meant to be a comprehensive survey. Section 5 concludes with an outlook.

## 2 Background

We build on the notions of information literacy and critical literacy as defined by Shapiro and Hughes [52] and Shor [53]. Both are intimately linked with educational approaches for achieving this kind of literacy. The combination of data mining and education has some touchpoints with the emerging field of Educational Data Mining.

### 2.1 Information Literacy

Shapiro and Hughes [52] propose the following dimensions of information literacy as a sketch for a curriculum:

> **"Tool literacy** , or the ability to understand and use the practical and conceptual tools of current information technology, including software, hardware and multimedia, that are relevant to education and the areas of work and professional life that the individual expects to inhabit.
>
> **Resource literacy** , or the ability to understand the form, format, location and access methods of information resources, especially daily expanding networked information resources. This is practically identical with librarians' conceptions of information literacy, and includes concepts of the classification and organization of such resources.
>
> **Social-structural literacy** , or knowing that and how information is socially situated and produced. This means knowing about how information fits into the life of groups; about the institutions and social networks – such as the universities, libraries, researcher communities, corporations, government agencies, community groups – that create and organize information and knowledge; and the social processes through which it is generated – such as the trajectory of publication of scholarly articles (peer review, etc.), the relationship between a Listserv and a shared interest group, or the audience served by a specialized library or Web site.
>
> **Research literacy** , or the ability to understand and use the IT-based tools relevant to the work of today's researcher and scholar. For those in graduate education, this would include discipline-related computer software for quantitative analysis, qualitative analysis and simulation, as well as an understanding of the conceptual and analytical limitations of such software.
>
> **Publishing literacy** , or the ability to format and publish research and ideas electronically, in textual and multimedia forms (including via World Wide Web, electronic mail and distribution lists, and CD-ROMs), to introduce them into the electronic public realm and the electronic community of scholars. Writing is always shaped by its tools and its audience. Computer tools and network audiences represent genuine changes in writing itself.

> **Emerging technology literacy** , or the ability to ongoingly adapt to, understand, evaluate and make use of the continually emerging innovations in information technology so as not to be a prisoner of prior tools and resources, and to make intelligent decisions about the adoption of new ones. Clearly this includes understanding of the human, organizational and social context of technologies as well as criteria for their evaluation.
>
> **Critical literacy** , or the ability to evaluate critically the intellectual, human and social strengths and weaknesses, potentials and limits, benefits and costs of information technologies. [...] This would need to include a historical perspective (e.g. the connection between algorithmic thinking, formalization in mathematics, and the development of Western science and rationality and their limits); a philosophical perspective (current debates in the philosophy of technology, the critique of instrumental reason, the possibility and nature of artificial intelligence); a sociopolitical perspective (e.g. the impact of information technology on work, public policy issues in the development of a global information infrastructure); and a cultural perspective (e.g. current discussions of the virtual body and of the definition of human being as an information-processing machine)."

Revisiting the brief comments made in the Introduction about DM in "everyday tools", one can see that the use of DM is today clearly a part of tool literacy and to some extent resource literacy for everyone. For example, people need to know how to use a search engine and have at least some knowledge about how to interpret a ranking (e.g., knowing that popular sites come first) and how to get the underlying resources, whether they are online (and may need a browser plugin to be viewed) or offline (and may require a visit to a local library). However, in using such tools, people are merely consumers of fixed DM applications and do not attain much in-depth knowledge about their workings, side effects, etc. Such knowledge tends to be imparted mainly in advanced computer-science courses that take detailed looks at underlying algorithms, weaknesses, attacks (such as "Google bombs" or "sybil attacks" on recommender systems[1]). Thus, for these students DM is part of the research literacy taught in Artificial Intelligence, Databases or Statistics courses.

Going beyond this, we here want to argue – and put forward as a research programme – that DM also has the potential to become an essential part of critical literacy for a wider audience. To understand why, we want to go into Shor's [53] understanding of critical literacy. These considerations will show that when used like this, DM can also be an important part of socio-structural literacy (a dimension from [52] that is arguably not separable from critical literacy in the [53] sense). Further, in Sections 4.4 and 4.5, we will argue that the 1996 notion of

[1] Google bombing (also known as search-engine bombing or link bombing) refers to practices that influence the ranking of particular pages in results. In a sybil attack, an attacker subverts a reputation system by creating a large number of pseudonymous entities, using them to gain a disproportionately large influence.

publishing literacy needs to be extended to also comprise decisions *not* to publish something, and that DM has a key role to play in the development of such a *privacy literacy*. It is to be hoped that curricula for teaching DM with respect to information literacy in this way will be sufficiently up-to-date regarding current developments and thereby contribute also to emerging-technology literacy.

### 2.2 Critical Literacy

Shor [53] defines critical literacy as "[habits] of thought, reading, writing, and speaking which go beneath surface meaning, first impressions, dominant myths, official pronouncements, traditional clichés, received wisdom, and mere opinions, to understand the deep meaning, root causes, social context, ideology, and personal consequences of any action, event, object, process, organization, experience, text, subject matter, policy, mass media, or discourse".

For Shor, the motivation for this is clearly humanistic and political: "Critical literacy involves questioning received knowledge and immediate experience with the goal of challenging inequality and developing an activist citizenry." This perspective implies that information (critical) literacy be a part of general education; and this could happen at every level, adapted to the respective students' intellectual and political knowledge, motivation and possibilities.

When one considers how critical literacy expresses itself in behaviours, the close link to education becomes clear: "Critical literacy can be thought of as a social practice in itself and as a tool for the study of other social practices. That is, critical literacy is reflective and reflexive: Language use and education are social practices used to critically study all social practices including the social practices of language use and education. Globally, this literate practice seeks the larger cultural context of any specific situation. 'Only as we interpret school activities with reference to the larger circle of social activities to which they relate do we find any standard for judging their moral significance,' Dewey wrote (Moral Principles in Education, 13)." Shor then investigates the case in which this "how" is taught and learned in settings that (a) are curricular courses, (b) deal intrinsically with language-as-practice, and (c) leave a certain extent of freedom. These he finds in (language) composition classes.

The goal of language composition classes is to teach students how to understand and use language – viewed as a primary means of human communication and knowledge transfer – to the best of their abilities. Our fundamental assumption is that today, the presentation of numbers or other data and their analyses by mathematical-statistical models and/or visualization techniques has become a second very important form of human communication, such that "data analysis" is by now as much of a "social practice" as language.[2] We will henceforth refer to Shor's notion as *critical language literacy* and to our new notion as *critical data literacy*. Thus, rephrasing Shor, we can formulate our vision: "Critical data literacy is reflective and reflexive: Uses of data analysis are social practices used to critically study all social practices including the social practices of data analysis".

[2] Note that we will treat data mining as a special form of data analysis, preferentially using the latter term for its generality.

### 2.3 Educational Data Mining

The combination of data mining and education also evokes Educational Data Mining (EDM). This is "an emerging discipline concerned with developing methods for exploring the unique types of data that come from an educational context. Given the widespread use of e-learning and the vast amount of data accumulated recently, researchers in various fields have begun to investigate data mining methods to improve e-learning systems." [50]. Thus, EDM can be said to focus on "data mining *of* data from educational processes", while we are interested in "data mining *for* learning processes". These two approaches are not mutually exclusive, one example is the use of data mining for personalization in e-Learning, see [3] for a recent literature overview. However, in contrast to current EDM, we focus on the learner actively and consciously performing data mining techniques (data mining as a social practice), and we treat data analysis as an important topic/object of teaching and learning in knowledge societies. We believe that an exploration of similarities and differences between EDM and our vision can be very fruitful for theory and education; this will be the topic of future work.

## 3 Towards Critical Data Literacy: A Frame for Analysis and Design

Based on the related work on information and critical language literacy, we now want to design a framework for identifying and classifying tools and their use that could help foster critical data literacy. Of course, natural language remains dominant for most communicational and rhetorical purposes; thus efforts aimed at critical data literacy should not displace, but augment those aimed at critical language literacy. (In fact, a thorough understanding of "how to lie with statistics", i.e. common misconceptions, misinterpretations of information visualizations, etc. is an essential basis for optimal use and understanding of such tools!) In a second step, we will then propose criteria by which the learning effectiveness of tools can be characterized.

### 3.1 A Frame of Analysis: Technique and Object

One way of structuring solution proposals (such as tools) that can contribute to Data Mining for Information Literacy is by their primary technique and object in the sense of Shor's definition of critical literacy. When language (and education) are the technique as well as the object of analysis, the analysis is an activity that can foster critical language literacy *sensu* Shor. The archetypal critical data literacy would then involve data analysis as technique and as object. However, a wider look at different types of objects and combinations can yield more insights; consider in particular the following (of which the last four will be the topic of the present paper):

**Language as Technique, Anything as Object:** This is a typical approach of basic essay writing (which could be a first step in a basic composition class, a prerequisite for later activities designed to foster critical literacy).

**Language as Technique, Data Analysis as Object:** this is the approach taken by a large range of writing designed to foster students' or the public's understanding of statistics. It ranges from books such as [25] about common errors, both intentional and unintentional, associated with the interpretation of statistics, and the incorrect conclusions that can follow from them – which has become one of the most widely read statistics books in history –, to specific media-critical observations like [39] about persistently incorrect interpretations and their wide and uncritical take-up by other media.

**Data Analysis as Technique, Anything as Object:** This is a typical approach of basic statistics or data mining classes. In Section 4.1, we show examples of Web-based tools that make this available to everyone in an engaging, thought-provoking and user-friendly way.

**Data Analysis as Technique, Language as Object:** an approach followed by information retrieval, text mining, and corpus linguistics (see e.g. Introduction). In Section 4.2, we will investigate different tools whose goal is a *critical analysis* of texts.

**Data Analysis as Technique, Data Analysis as Object:** A "pure" form of critical data literacy that probably requires the most abstraction capabilities. To the best of our knowledge, this is not very wide-spread yet. An example will be described in Section 4.3.

**Data Analysis as Technique, Behaviour in Data Spaces as Object:** This combination has created much novelty and interest in recent years, probably since behaviours in data spaces like the Internet are increasingly "authentic/situated" behaviours of many people. Sections 4.4 and 4.5 will discuss Web-based tools whose goal is a *critical analysis* of such behaviours.

As for language use, education appears to be the adequate *cultural environment* for this. This could range from the integration into the school or university curriculum to uptake by adult-education institutions and other voluntary, semi-formal settings. The Web appears to be a very well-suited *medial environment* for the storage, transfer, and use of such learning activities, but like in other areas, an approach that solely relies on autodidactic approaches appears risky. Data analysis is an activity best done with the help of software, thus, we focus on (usually Web-based) software tools as cognitive tools.[3] (Thus, the use and ideally also creation of tools becomes the analogue of natural-language use.)

[3] "The term 'cognitive tools' was coined by the book edited by Lajoie and Derry [29] [... : ] computers could support learning by explicitly supporting or representing cognitive processes. In such a sense computers could serve as being a 'mind extension', augmenting the limited capacity of the brain. More general we can define cognitive tools as being instruments that are designed for supporting cognitive processes and thereby extending the limits of the human cognitive capacities. In principle anything can be a cognitive tool, for instance a sheet of paper and a pencil can be a cognitive tool to support the cognitive process of remembering items, extending the limited capacity of working memory." [58, p. 389]

### 3.2 On the Chances of Achieving Critical Data Literacy: Principles of Successful Learning as Description Criteria

Modern learning theories name the following principles of successful learning[4]

1. It is situated and authentic; multiple contexts are investigated.
2. It is active and constructive.
3. Multiple perspectives are taken.
4. It is social.
5. It involves articulation and reflection.

Principle 1 is very relative to what interests a learner – "situated" is often thought of being about one's "natural environment", but this is very different for different learners, and an avid mathematician (for example) may "live" in a mathematics world. The investigation of multiple contexts is relevant to support transfer learning. All these demands can be met easily with the Web – data and questions to the data that 'matter' to the learners can be chosen from the vast array of possibilities. We will focus on current-events topics as likely to be interesting to a wide range of (news-consuming) people.

Principle 2 is given with data-analysis tools: Learners take an active role in defining and performing analyses, and they construct new representations (such as classifiers from data). However, the true extent of constructiveness and freedom of re-representation needs to be investigated for each case.

Principle 3 is not automatically given. We will investigate below how data mining can support the exploration of and creation of different perspectives. Note the 'playful' element of such role-playing activities (today often referred to as "identity management"), which is considered highly important for intellectual flexibility and even successful behaviour in societies [51].

Principle 4 is a key point of the Social Web. It is realized mainly through today's extensive social networks, in which people are free (and often quite active) in commenting on each other's utterances and extending on them. Note, however, that this is not automatically given just by offering a tool; in an educational setting, appropriate instructions can help to realize it.

Principle 5 requires two things: To articulate content, one needs to represent it, for example in words, by a drawing, etc. It is well known from the cognitive and learning sciences that a mere repetition of a given representation is usually not effective, that rather, a different re-representation is helpful and often necessary to gain insights, solve problems, etc. [41]. Reflection involves the inspection and active construction of such re-representations and in general meta-cognitive strategies for managing the learning process [49].

We will use these five principles as criteria for the characterization of the case studies / tool examples to be presented in the following section. In addition, we will use criterion

6. Data mining sophistication, in order to gauge to what extent the examples use the power of state-of-the-art data mining.

[4] This list is my own summary of relevant findings from (social) constructivism and constructionism, based on sources such as [1,14,28,36,43,59,61,64].

# 4 Examples: Tools and Other Approaches Supporting Data Mining for Information Literacy

In the following, we will analyse different examples of current-day uses of data mining for critical data literacy as defined above. They will be ordered by their primary technique and object as laid out in Section 3.1, and characterized by the six criteria as laid out in Section 3.2. Both perspectives are designed to outline what current tools can do and what is missing most.

## 4.1 Analysing Data: Do-It-Yourself Statistics Visualization

The press is full of information graphics about current events, and "smart graphics" can be "worth a thousand words". *Thematic maps* are maps that focus not on the geographic features of countries and other regions, but on political, social or economic quantities that characterize those regions. They become especially interesting when innovative visualization strategies are chosen.

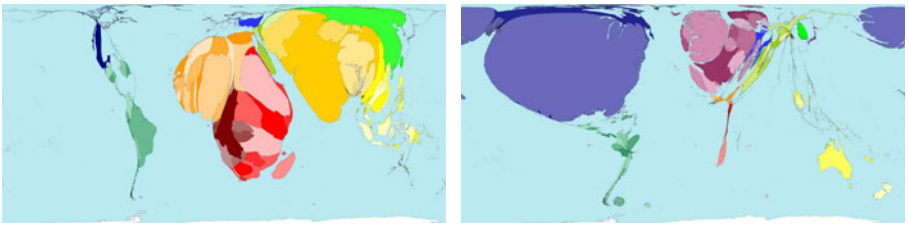


**Fig. 1.** Worldmapper: Child labour (left) and Toys imports (right).[5]

The site `www.worldmapper.org` transforms each country's area by a quantity associated with that country. Technically, Worldmapper is a collection of world maps, using equal area cartograms where territories are re-sized on each map according to a particular variable. One example is population. In addition, juxtapositions and animations can show how these relationships between countries differ when different statistics are chosen, such as the one shown in Fig. 1. Some additional features such as interactive maps that support zooming and panning are offered for some datasets. The site contains 696 maps, with associated information and PDF 'poster' file. Each map relates to a particular subject. Data files that underlie the maps can be downloaded in a popular and interoperable format (Excel). The project is a collaboration of researchers and practitioners at the universities of Sheffield and Michigan, the Leverhulme Trust and the Geographical Association. The data were mostly supplied by the UN.

`www.gapminder.org` offers a software that allows one to plot various timeseries and contrast them. Its goal is to "unveil the beauty of statistics for a fact based world view". Data come from official sources like the OECD or the International Labor Organization. Fig. 2 shows an example. Different options of overview and

[5] from `http://www.worldmapper.org`, retrieved on 2010-07-30.

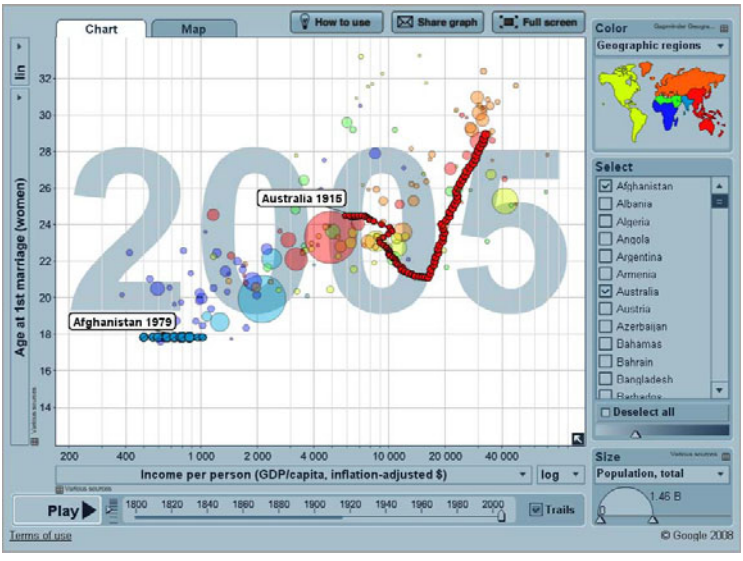



**Fig. 2.** Gapminder: income vs. age at first marriage, Australia vs. Afghanistan, tracking years shown as circles. Snapshot taken at the end of animation.[6]

detail sequences and zooming and changing between representations (e.g. chart or tabular) are offered. GapMinder is available as an online version and as an offline desktop version. Different output formats such as videos, Flash or PDF are supported. A plugin in GoogleDocs makes it possible to create these graphs with one's own data. [7] Special attention is paid to the needs of teachers or others who want to use GapMinder for their own presentations. GapMinder was created as a Foundation in Sweden and was bought by Google in 2007, where the original developers continue to work.

An analysis along our six principles reveals the following.

*1. Situatedness and authenticity, multiple contexts.* Both tools apply statistics and data visualization to authentic official data that are clearly situated in systems of development indicators and official statistics. Multiple contexts are supported by different datasets.

*2. Active and constructive.* In Worldmapper, activity is basically limited to choosing graphs; in GapMinder, further interaction options (like the selection of countries) support a stronger sense of user activity and choice. Activity is more strongly supported by GapMinder when users choose their own datasets.

[6] from `http://www.gapminder.org`, retrieved on 2010-07-30.

[7] `http://www.gapminder.org/upload-data/motion-chart/`.

[8] `http://flowingdata.com/wp-content/uploads/2007/09/snow_cholera_mapsm.jpg`; `http://www.evl.uic.edu/luc/422/GIFs/challenger2.gif`, see [56, p. 47] `http://revcompany.com/blog/wp-content/uploads/2009/08/challenger-disaster.gif`, adapted from [56, p. 45]; retrieved on 2010-07-30.

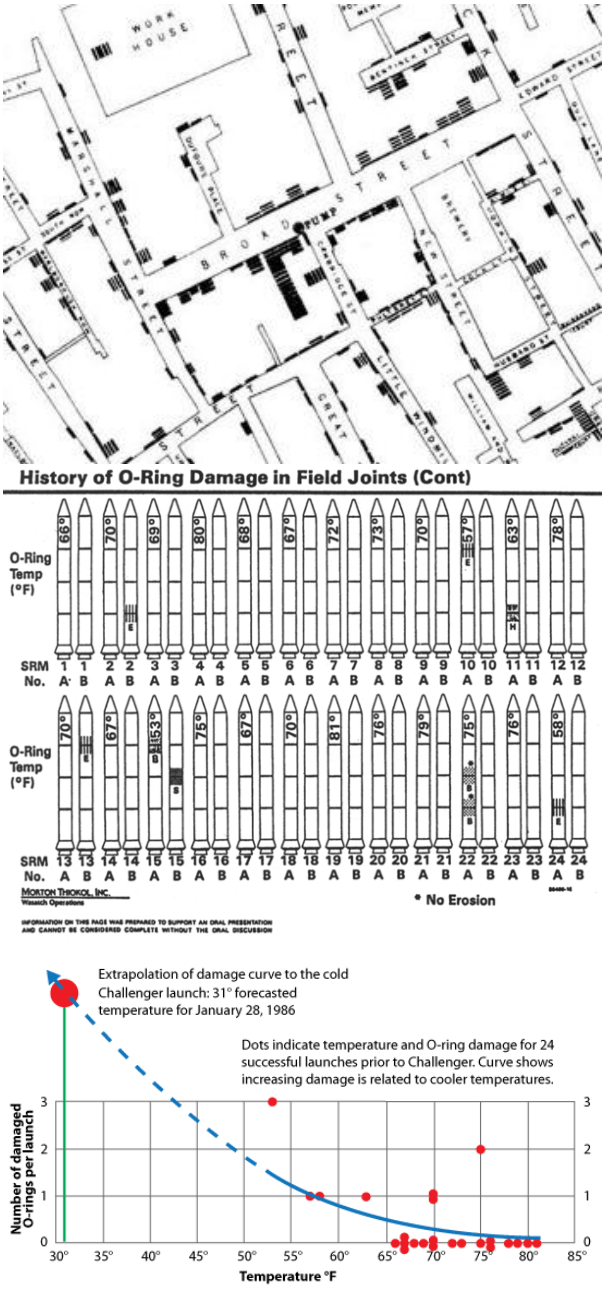



**Fig. 3.** Classical examples of visual data mining: (a, top): Snow's cholera map, excerpt, showing the Broad Street pump location and the number of deaths per house as bars. Challenger launch temperature and number of damaged O-rings: (b, middle): graphic used by the engineers before the launch, which did not prevent the launch. (c, bottom): proposal by Tufte.[8]

A sense of construction can be created in the sense that one transforms relational tables into graphics; although the constructive element is limited by the available analysis options.

*3. Multiple perspectives.* Multiple perspectives could be supported by the choice of different but complementary indicators of a common thematic complex (e.g. girls' illiteracy and child labour as indicators of poverty, or prevalence of diseases and alcohol consumption as indicators of health status). More challenging appears to be the use of different datasets that operationalize the same construct in different ways, collected by different agents – this could stimulate discussions on multiple perspectives more strongly. The latter is only supported by tools such as GapMinder that allow users to visualize datasets of their own choosing.

*4. Social.* Beyond the sharing of graphs/analysis results, the tools offer no specific social usage models. Sharing is supported by GapMinder in a common Web2.0 fashion: a link that supplies a URL to be inserted into a website or email. Sharing and joint editing via GoogleDocs could be a basis for further social usage models.

*5. Articulation and reflection.* The visualizations and especially the juxtapositions offer re-representations of tabular and quantitative data; as such, they articulate and re-represent. This can arguably be highly conducive to reflection and the discovery of new solutions in the sense of successful information visualizations or *visual data mining*.

*6. Data mining sophistication.* There is none in the algorithmic procedures used; but the visual re-representation can support a (potentially misled) human visual perception of correlations and identify areas where "asking more questions" could be interesting (cf. active learning in machine learning). However, as mentioned above, both tools can support visual data mining.

Further analysis and visualization options are offered by the tool ManyEyes, which will be presented with a focus on its text-analysis capabilities in the following Section.

**Visual Data Mining.** Visual data mining has been defined as focusing on integrating the user in the KDD process in terms of effective and efficient visualization techniques, interaction capabilities and knowledge transfer.[9]

Well-known examples are two cases explained in detail by Edward Tufte [56]: Snow's Cholera Map, which in the mid-19th century helped to identify polluted water as the source of cholera (by visually correlating a specific water pump with a high incidence of cholera deaths in the neighbourhood), see Fig. 3 (a), or the hypothetical prevention of the 1986 Challenger disaster by the replacement of the actually used, inadequate data visualization, by one that clearly visualizes the intended argument, see Fig. 3 (b) and (c).

[9] `http://www.dbs.informatik.uni-muenchen.de/Forschung/KDD/VisualDM/`, 2001-01-18, retrieved on 2010-07-26.

### 4.2 Analysing Language: Viewpoints and Bias in Media Reporting

Text mining is the application of data mining techniques to texts. This can mean, for example, that classifiers are learned to assign text documents such as Web pages, Twitter messages or even tag sets to a category. These categories may be determined by content (e.g. "sports" vs. "computers", cf. [37]), by opinion polarity (e.g. expressing a positive or a negative opinion about a previously determined topic, cf. [42]), the likely author [57,20], or other criteria.

A criterion that has recently received a lot of attention is "viewpoint" or "bias", as identified by, for example, the (often self-declared) belonging to one of a small set of factions on an issue or even an image-creating identity itself. Examples of the first case are adherents of a political party or members of societal or national groups with pronounced stances on controversial issues (such as Palestinians and Israelis on issues such as Gaza). Examples of the second case are media that represent such stances, such as Al Jazeera on the one hand and Fox on the other. The analysis of, e.g., different lexical choices depending on viewpoint is a classical method in Critical Discourse Analysis, see e.g. [48]. There, however, it is still predominantly done manually and thus on small samples, but corpus-analytic/text-mining methods are gradually beginning to be applied and allow the analysis of a much larger set of documents.

Such analyses are done with supervised learning of global models (e.g., classifier learning), with unsupervised learning of global models (e.g., clustering), or with local-pattern detection (such as frequent phrases).

Viewpoint mining has so far been explored mostly in research efforts. Supervised modelling typically analyses a topic and then learns a classifier to distinguish between viewpoints on this topic. One example is a media analysis of local

**Table 4:** Decision tree for the W1W2 feature set, with previous wrapper feature subset selection (accuracy: 80.56%).

```
tribe <= 0
|  political/leaders <= 0
|  |  what <= 0.001479
|  |  |  pledged<= 0
|  |  |  |  forces <= 0
|  |  |  |  |  what <= 0.000516
|  |  |  |  |  |  it/had<= 0
|  |  |  |  |  |  |  union <= 0.002273: LO (30.0/7.0)
|  |  |  |  |  |  |  union > 0.002273: WE (4.0)
|  |  |  |  |  |  it/had> 0: WE (3.0)
|  |  |  |  |  what > 0.000516: WE (5.0)
|  |  |  |  forces > 0: WE (3.0)
|  |  |  pledged> 0: WE (3.0)
|  |  what > 0.001479: LO (8.0)
|  political/leaders > 0: LO (4.0)
tribe > 0: WE (12.0/1.0)
```

**Fig. 4.** Decision tree learned to predict LO (local newspaper) vs. WE (Western newspaper), from [46]


the topical and ideological weights learned by the joint topic and perspective model from the bitterlemons corpus . Red: words emphasized more by the Israeli authors. Blue: words emphasized more by the Palestinian authors.

**Fig. 5.** Tag cloud showing word frequency and typicality in the two corpora, from [31]

| | |
|---|---|
| TOPIC | Iraq, Baghdad, Hussein, shiite, trials, insurgents, troops |
| CNN | insurgents, Hussein, attorney, Kember, family, British |
| AJ | shia, sunnis, occupation, Saddams, rebels, attack, killed, car |
| TOPIC | Palestinian, Gaza, Israel, Sharon, Hamas, Abbas, militant |
| CNN | militant, Israel, pullout, missiles, launch, Putin, Beirut, jews |
| AJ | settlers, Hamas, barriers, Israeli, clashes, Hezbollah, farms, suffer |

**Fig. 6.** Two example topics with the words characterising the topic as a whole and those specific for the two sources, from [19]

vs. Western media reporting on the Kenyan elections; various classification algorithms found, for example, that usage of the word "tribe" was a near-certain predictor of a text having appeared in a Western newspaper [46]. This was found by applying various standard classification-learning algorithms such as decision trees, see Fig. 4.

In [31], a language model of documents for two viewpoint categories was learned; the results were visualized in a tag cloud where size shows frequency and colouring typicality of one of the two viewpoints, see Fig. 5.

In [19], two coarse-grained viewpoints (documents either published by Al Jazeera or CNN) were input as data labels; based on them, topics were learned in an unsupervised way alongside words that are typical of how the two viewpoints/media report on these topics. The method relies on (a) nearest neighbour / best reciprocal hit for document matching to identify topics and (b) Kernel Canonical Correlation Analysis and vector operations for finding topics and characteristic keywords for viewpoint characterization. Figure 6 shows two examples.

*Table 1.* The ten highest climbers in the list of nearest neighbors to islam and christendom after 9/11 for the syntax-based model

| islam | christendom |
|---|---|
| *terrorisme* 'terrorism' | *rechtsstaat* 'constitutional state' |
| *koran* 'Quran' | *jodendom* 'Judaism' |
| *moderniteit* 'modernity' | *burgerschap* 'citizenship' |
| *islamist* 'islamist' | *moderniteit* 'modernity' |
| *verzorgingsstaat* 'welfare state' | *hindoeïsme* 'Hinduism' |
| *religie* 'religion' | *christen* 'Christian' |
| *fundamentalist* 'fundamentalist' | *moraal* 'moral' |
| *islamiet* 'Islamite' | *protestante* 'Protestant' |
| *moslim* 'muslim' | *fundamentalisme* 'fundamentalism' |
| *jihad* 'jihad' | *idealism* 'idealism' |

**Fig. 7.** Words associated most strongly with the two target words "Christianity" and "Islam", from [44]

Unsupervised learning has been employed to identify media bias by [44]. They performed two orthogonal analyses on a large corpus of Dutch newspapers: one that distinguishes between articles published before 9/11 and articles published after it, and a second analysis that distinguishes between five newspapers ranging from quality to popular. The question is how the perception of Islam and Christianity may have changed (or differs between sources); the method is based on lexical co-occurrences. The authors use two types of such word vector space models: one based on co-occurrence in documents (the typical information-retrieval / text-mining vector-space model that is, for example, used in the studies described above), and a syntax-based model that takes into account how frequently two words occur in the same syntactical roles. The target words "Islam"/"Christianity" are described by vectors in these feature spaces, and other words with (cosine-)similar vectors are identified as related in the wider meaning. Some sample results (document- and syntax-based models coincided on the general results) are shown in Fig. 7. In addition, significant differences were found between the newspapers, including a higher co-occurrence of "Islam" and "terrorism" in the most popular newspaper.

However, none of these examples are available in the form of freely accessible and/or Web-based tools; thus they cannot be used easily. In principle, everyone could set up and perform similar mining tasks with the help of the many free tools that exist on the Web; however, the effort needed to pre-process Web documents for mining as well as the expertise needed to operate powerful tools are considerable, such that this is not a realistic option. Individual and task-specific exceptions confirm this rule, especially in the computational humanities.

"... I want to make three brief points about the resignations of the eight United States' attorneys, a topic that I know is foremost in your minds.
First, those eight attorneys deserved better. ... Each is a fine lawyer and dedicated professional. I regret how they were treated, and I apologize to them and to their families for allowing this matter to become an unfortunate and undignified public spectacle. I accept full responsibility for this.
Second, I want to address allegations that I have failed to tell the truth about my involvement in these resignations. These attacks on my integrity have been very painful to me. ..."

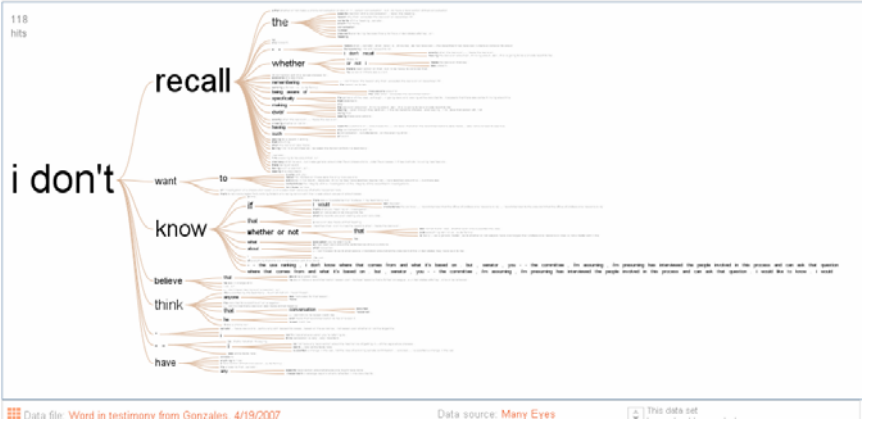



**Fig. 8.** (top) Excerpts from Gonzalez' resignation speech. (bottom) A Word Tree.[10]

One example is the MONK workbench that "includes approximately 525 works of American literature from the 18th and 19th centuries, and 37 plays and 5 works of poetry by William Shakespeare. [...] MONK provides these texts along with tools to enable literary research through the discovery, exploration, and visualization of patterns. Users typically start a project with one of the toolsets that has been predefined by the MONK team. Each toolset is made up of individual tools (e.g. a search tool, a browsing tool, a rating tool, and a visualization), and these tools are applied to worksets of texts selected by the user from the MONK datastore. Worksets and results can be saved for later use or modification, and results can be exported in some standard formats (e.g., CSV files)."[11] The documents in the MONK collection are pre-processed, such that users can concentrate directly on the text-mining tasks. However, the interface is more demanding than, for example, that of GapMinder, such that casual users may be discouraged from experimenting with the tool. An earlier, simpler version of MONK demonstrated the ideas by allowing users to label selected poems by Emily Dickinson as "erotic" or not and then determining, via Naive Bayes learning, which words contributed most to this classification [45].

[10] `http://www.justice.gov/archive/ag/testimony/2007/ag_speech_070419.html`; Word Tree generated with `http://manyeyes.alphaworks.ibm.com/manyeyes/`, retrieved on 2010-07-30.

[11] `http://monkpublic.library.illinois.edu/monkmiddleware/public/index.html`

An example of local pattern mining that is highly accessible are the Word Trees of `http://manyeyes.alphaworks.ibm.com/manyeyes/`. The method is the discovery of a word (sequence) trie, and the visualization of the sequences with common prefixes together with size indicating frequency. As an example, consider Fig. 8, which shows an analysis of the 2007 resignation speech of US Attorney General Alberto R. Gonzalez. The speech at first glance appears to emphasize his regret over the dismissals of eight attorneys that others claimed he had an important role in, but at second sight looks more self-related.

An analysis along our six principles reveals the following.

*1. Situatedness and authenticity, multiple contexts.* All tools apply text mining to authentic data that are clearly situated as documents from or reporting on current events. Multiple contexts are supported by different datasets, especially in ManyEyes where they can be uploaded by anyone.

*2. Active and constructive.* The interaction and analysis options in MONK provide user activity and choice. Activity is more strongly supported by ManyEyes when users choose their own datasets. A sense of construction can be created in the sense that one transforms texts into graphics; especially in MONK, users can get an idea of the degrees of freedom inherent in making decisions along the different steps of data mining. In addition, the arguments of Section 4.1 apply analogously.

*3. Multiple perspectives.* Multiple perspectives are supported in analogous ways (and for the same reasons) as in the tools presented in Section 4.1.

*4. Social.* This is supported in analogous ways as in the tools presented in Section 4.1. In addition, the uploading of one's own datasets (which can then be re-used by others) can give rise to social usage in ManyEyes.

*5. Articulation and reflection.* The visualizations and especially the juxtapositions offer re-representations of textual, sentential data; as such, they articulate and re-represent. This happens in various ways: In classifier learning, significant words are extracted that are highly characteristic of a certain category (e.g. viewpoint); in topic detection, "issues" may be discovered (along with typical words common to all reporting on them), and in word trees, frequently used phrases may be discovered. This can arguably be highly conducive to reflection about the connotations of these words or rhetorical effects of these phrases.

*6. Data mining sophistication.* Data mining sophistication ranges from the straightforward application of known algorithms to new problems, to the formulation of new and advanced mining methods. It is likely that freely and flexibly accessible Web-based tools (which are our focus here) will continue to concentrate on straightforward methods; also they are likely to continue concentrating on the application of one type of analysis, since the composition of analysis steps is very demanding for non-experts. (Witness the mixed reception of mashup tools like Yahoo! pipes or its competitors, cf. [55,15].)

### 4.3 Analysing Data Mining: Building, Comparing and Re-using Own and Others' Conceptualizations of a Domain

Two key elements of the Social Web are complementary: the realization that people are highly different, but that, if brought together by the infrastructure of a worldwide network and appropriate algorithms for finding similarity, many people who are alike (even if they are only 10 worldwide) can profit from each other, and that even more people can profit from others who are only partially alike.[12] This observation has been utilized in a large number of end-user systems, most notably recommender systems based on various forms of collaborative filtering. Collaborative filtering employs a wide range of sophisticated data-mining techniques, analysing actions (who bought what? who rated what?), content (what are the features of the items people like?) and structures (how do social ties interact with other forms of similarity?). However, the dominant form of presentation of these data-mining results is a focus on the result (with the implicit understanding that this is helpful), and less a focus on how it was reached.

The CiteSeerCluster/Damilicious tool [10,60] is an attempt to use the basic logic of data-mining for a purpose (in this case, the search for scientific literature) to help people reflect on how these results arise and what this means for them – and thus how they may be able to re-use the results. The idea is to help users in sense-making of the results of their literature searches on the Web: on an individual level, by supporting the construction of semantics of the domain described by their search term, and on the collective level, by encouraging users to explore and selectively re-use other users' semantics.

The user can, starting from an automatically generated clustering, group a search result document set into meaningful groups (see Fig. 9), and she can learn about alternative groupings determined by other users. To transfer a clustering of one set of documents to another set of documents, the tool learns a model for this clustering, which can be applied to cluster alternative sets of documents. We refer to this model as the clustering's intension ("there is a group of papers dealing with security and privacy"), as opposed to its extension, which is the original, unannotated grouping of the documents ("documents 1, 13 and 35 belong together").

This approach supports various measures of diversity. Such measures can be used to make recommendations and present new, possibly interesting viewpoints of structuring the result set. Specifically, a measure is introduced to quantify the diversity of users, defined on how they have in the past grouped identical sets of resources (here: scientific publications). Two users are maximally diverse if they have imposed orthogonal groupings and minimally diverse if they have imposed the same grouping. By convention, users who have not imposed structure on identical sets of resources could be considered maximally diverse or even as a separate class. A visualization based on this measure of diversity as distance metric and multi-dimensional scaling can then serve to give users a first overview of how "close" others are to them, see the overlaid image at the bottom right of Fig. 9.

[12] The "long tail" is an expression of these complementary principles in terms of how to produce for and make profit in a world composed of niche markets.

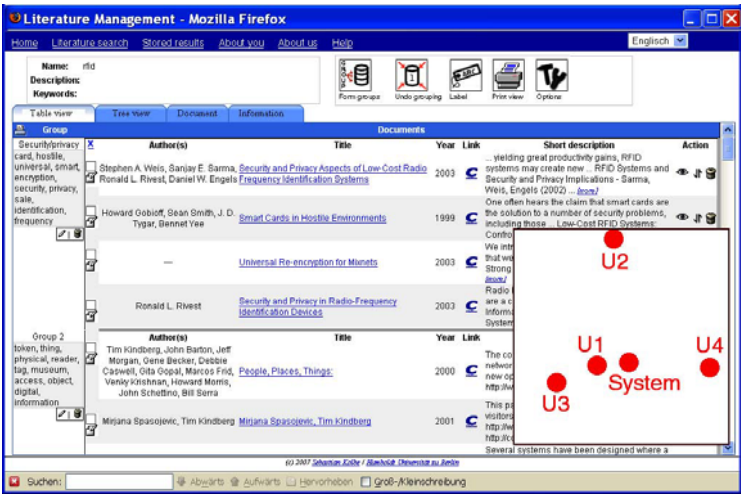



**Fig. 9.** Main window: User interface for creating one's own grouping and ontology from the search result, from [10]; overlaid image: a visualization of user diversity using multi-dimensional scaling ("System": a user who did not change the system's grouping proposal), from [60].

An analysis according to the six criteria shows the following.

*1. Situatedness and authenticity, multiple contexts.* The tool supports situated and authentic activities to the extent that literature search is an integral part of the work of an advanced student or other academic. Multiple contexts are only supported to a limited extent (various top-level queries as selectors of the starting set of scientific publications).

*2. Active and constructive.* The tools involves the user strongly in active and constructive sense-making. While users can, in principle, use the tool like a search engine that performs automatic clustering (for example, `www.clusty.com`), a user study showed that the possibility to re-group is welcomed and used extensively and with good results [10]. Likewise, dealing with user diversity by choosing, for example, very similar users (to be supported in one's ways of thinking) or very dissimilar users (to broaden one's horizon), can be highly constructive and active.

*3. Multiple perspectives.* Damilicious can help users take different perspectives on search results and thereby reflect more deeply about resources on the Web and their meaning.

*4. Social.* The explicit representation of different users can represent a good basis for treating literature search as a social activity. The current version of Damilicious is only a prototype; it therefore needs to be extended by social interaction functionalities such as those of `www.citeulike.org` or `www.bibsonomy.org`.

*5. Articulation and reflection.* The interplay between the formation of concept extensions and concept intensions involves constant re-representations, articulations and also reflections of the content concepts of the scientific domain in which literature is being investigated. Similarly, the meta-notion of "people who do (don't) think like me in the sense that they structure the world in the same (different) ways" can serve to articulate the comparatively unreflected notion "people like me" and make it more amenable to reflection.

*6. Data mining sophistication.* The interplay between concept extensions and intensions extends state-of-the-art methods in conceptual and predictive clustering [4, 16].[13]

### 4.4 Analysing Actions: Feedback and Awareness Tools

Doing data or media analyses, as described in the previous sections, probably requires a comparatively deep interest in the phenomena being studied that, for many people, will go beyond a basic need to "be informed". A rather different approach is to take advantage of the liking that many people have to "look into a mirror", where the phenomenon being studied is – them. *Feedback and awareness tools* analyse the log files an individual produces, and compile various statistics about these usage data or run mining analyses on them.

Several such tools have been touted in recent years as enhancers of personal productivity: "time-management software" allow users to see which applications they use for how long, which Web sites they visit and for how long, etc. This can be used as the basis for billing clients.[14] One site claims that "On average [our tool] recovers 3 hours and 54 minutes worth of productive time per week per person".[15] These tools generally show users simple statistics such as hours of time spent on Facebook during the last week.

Such self-observation can be useful for learning itself. As studies such as [21] have shown, the support of metacognitive activities that reflect a learning process *ex post* can improve learning success. Feedback may be given about resource usage and their timing during learning, e.g. in [21], or about the resources used, their semantic classes, and the type of navigation/search between them [6,5].

A recent related development are *privacy feedback and awareness tools* that are motivated by the increasing revelation and spread of highly personal self-profiling especially in today's Social Media, and the observation that in spite of many people describing themselves as highly privacy-conscious when asked explicitly, online behaviour differs markedly from these attitudes [7,2]. Alongside this, appeals to be more protective of one's personal data scarcely have any effect; the idea of privacy feedback and awareness tools is to show users *within the context of their potentially privacy-related activities* (e.g. within their social network platforms) important consequences of activities they have performed.

[13] For a method with similar end results but a different data-mining approach and goals, see [40].

[14] `http://manictime.com/`

[15] `http://www.rescuetime.com/`

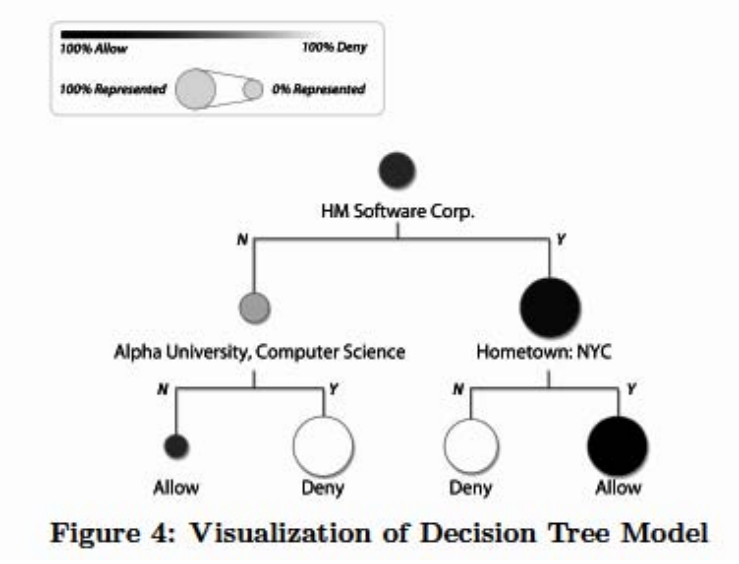



**Fig. 10.** How the current user has decided to share her birthday or not in an online social network based on her friends' features, from [18].

Privacy awareness tools aim at fostering understanding and reflection: For example, [30] suggest improving privacy sensitivity in systems through feedback that enhances users' understanding of the privacy implications of their system use. This can be coupled with control mechanisms that allow users to conduct socially meaningful actions through them. These ideas have led to suggestions like the identityMirror [33] which learn and visualize a dynamic model of user's identity and tastes. Similar ideas are embodied in the concept of privacy mirrors [38] or in the [24] proposal for linkage control in identity management systems.

*Data mining* can extend the scope of these applications and simulations for information inference by employing more sophisticated forms of induction and deduction for demonstrating the possible consequences of a user's actions. In [23], we have shown how the spread of visibility/accessibility of a user's profile and relational data may be computed. In the remainder of this section, we first outline how inferencing could be employed in the SNS models we studied and then sketch more sophisticated approaches.

**Towards Conflict Avoidance and Resolution: Feedback and Negotiation Mechanisms.** The analyses of the cases in [23] have shown, among other things, access permission inconsistencies: some information becomes visible beyond the group of people originally intended. (A simple example is the following that uses standard social-network functionality: Let $A$ restrict visibility of his friendships to "friends", and let $A$ be friends with $B$. $B$ sets the visibility of his friendships to "friends-of-friends". Then, a friend $C$ of $B$'s can, by choosing her friends, determine who gets to see the relationship $A$–$B$. The recipients may

include $D$, with whom $A$ never wanted to share any information.) A "feedback mechanism" could be implemented to make users aware of this. It would signal, upon the intention of users $A$ and $B$ to establish a relation, to them that the actual group of people who will be able to see the relationship will be larger than what they (probably) expect based on their individual permissions.

If such access permission inconsistencies are judged to be acceptable, there is no problem. However, if users disagree, other models will have to be considered that either avoid access permission conflicts or allow users to articulate conflicting requirements and find designs that allow users to negotiate these prior to design or during run-time. It is important to underline the fact that users cannot and do not decide on their preferences alone as long as relational information (such as friendships) and transitive access control (such as the de-facto delegation through access rights for friends etc.) is implemented.

In any case, users should be provided with feedback, informing them about how far their relationship information travels through the graph. This feedback can be coupled with collective privacy setting negotiation mechanisms, building on policy visualization techniques like the Expandable Grids [47]. The intended result are better-informed user choices about what information – about themselves or others – to publish and how.

**Design Choices in Feedback Mechanisms based on Data Mining.** Going beyond straightforward what-if simulations, we believe that feedback for awareness-raising simulations should not only be limited to the application of data-mining models such as classifiers or graph inference results. Rather, it is vital to consider also the dynamics with which users' data-related activities contribute to the learning of these models. Thus, we propose to integrate data mining more fully – by considering also statistical information, by considering also the learning stages of a model – into creating privacy awareness tools.

As one example in social networks, consider the problem of inserting structure into the set of "friends". In current SNS, these sets have no internal structure, or friends can be assigned to predefined classes [12]. These sets grow too fast for many users and easily become unmanageable. This is reminiscent of the email structuring problem (which has been addressed by several machine-learning approaches such as [16]). In addition, it is however an increasing privacy problem, because profile and relational information is distributed either to all friends or to none. To improve on this situation, the user's set of "friends" could be clustered by connectivity, a classifier could be learned from the user's own past communication behavior with these different clusters, and a recommender could be derived from it to suggest that in the future, it might be advisable to withhold certain information from this group. Such mechanisms were implemented in [22] based on tie strength characterized by multiple dimensions representing trust and closeness among friends, and in [18], which also showed that friends' connectivity patterns are a better predictor than these friends' profile information for the willingness to share certain information. This type of clustering / classification / recommendation mining could be incremental, such that the effects of decisions such as accepting an invitation to become friends attain visibility.

This basic idea gives rise to a number of choices and questions: (a) The implications could be shown to users in pull or push fashion. Push has the advantage of potentially reaching more people, but the disadvantage of potentially becoming tiresome and ignored if too many warnings are issued. Machine learning could in turn be used to learn how and when to make proposals to a user to maximize effectiveness (cf. earlier work on desktop agents). (b) Inferences can be based on already-stored data or on what-if simulations. The latter have the advantage of warning people "before it's too late", but may therefore also create a false sense of security. This tradeoff remains an open issue for interaction design. (c) The target groups to whom inferences are shown can range from end users (natural persons in SNS applications, businesses in applications like [8]) to SNS providers. (d) Due to the interdependencies between users and the external effects of their decisions, mechanisms may need to address groups rather than individuals. The challenge then lies in how to best address groups rather than individuals only. Some of the issues involved (such as preference aggregation) will be similar to those in issuing recommendations to groups [27], but further ones will surface due to the as-yet little-explored nature of privacy seen as a collective good.

An analysis according to the six criteria shows the following.

*1. Situatedness and authenticity, multiple contexts.* All tools apply data mining to authentic data that are clearly situated as documents from or reporting on the user's own actions. Multiple contexts are supported to the extent that data are collected in multiple work or life contexts. (However, it is an unexplored topic how these contexts should be kept apart.)

*2. Active and constructive.* Activity and constructiveness are similar to those in the tools described in Section 4.1: Users can create representations, but they are limited to the given analysis options. In addition, by being the ones who create the data (by their actions), users can become active in new ways, for example trying out different behaviours. Thus, these tools may have a stronger feedback effect on behaviour.

*3. Multiple perspectives.* In principle, different analysis options could support multiple perspectives; however, it may be that only one perspective (the user watching herself) is encouraged.

*4. Social.* Feedback and awareness tools may comprise social functionalities of different types. Several time-management tools allow one to see, for example, which applications other users use or how one is ranked based on a given measure relative to others (based on a small social network of tool users).

*5. Articulation and reflection.* The visualizations and especially the juxtapositions offer re-representations of textual, sentential data; as such, they articulate and re-represent: Activity is translated into data (such as number of hours spent with a certain application) or into consequences (such as friendships becoming visible to a certain group of people). This can arguably be highly conducive to reflection about the consequences of one's actions (and action based on it, cf. RescueTimes' claim regarding "rescued" work time).

*6. Data mining sophistication.* Data mining sophistication today mostly ranges from the straightforward depiction of basic data statistics, to the application of known algorithms for purposes such as classifier learning. A continued focus on simpler methods appears likely, for the same reasons as discussed in Section 4.2.

### 4.5 Analysing Actions: Role Reversals in Data Collection and Analysis

As discussed in the previous section, the privacy of one's activities and data on the Internet are becoming an increasing focus of many efforts to increase information literacy. Here, the notion that 'privacy consists of protecting personal data of individuals' is too narrow to give rise to a true understanding of the dynamics of data disclosure, sharing and hiding. First, as already noticed in the previous section, in richly linked worlds like online social networks, there are many external effects of any one person's privacy-related behaviour. Second, not only

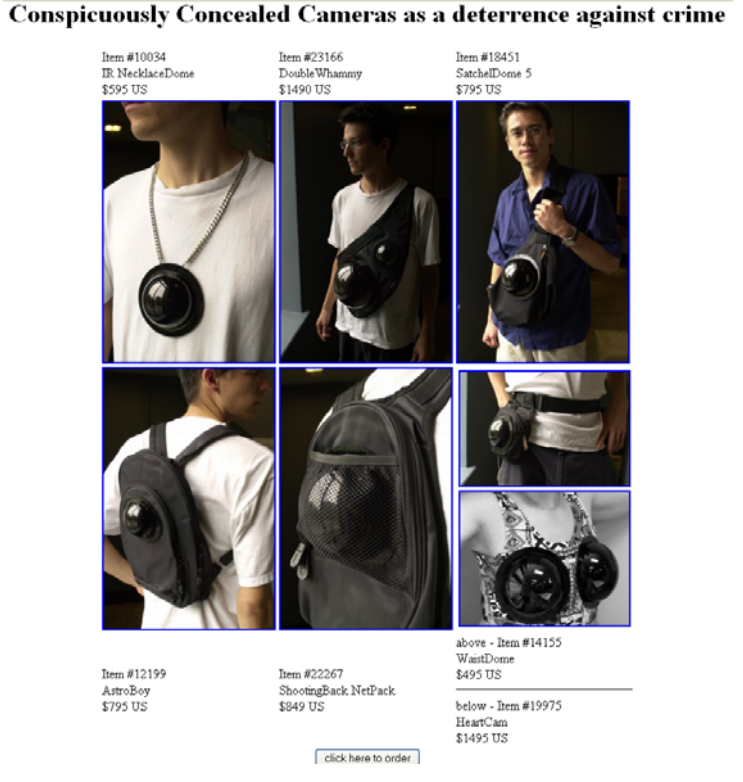



**Fig. 11.** Sousveillance equipment as high-end consumer goods.[16]

[16] `http://wearcam.org/domewear/`, retrieved on 2010-07-30.

individuals have interests in hiding their data; one then speaks of "trade secrets" (when the agent is a business), "national security and classified information" (when the agent is a state), etc.

One of the truisms of information hiding is that it is often used to maintain power. Thus, role reversals may open up discussions of how unequal distributions of information found unequal distributions of power, or even change these power relations.

In the area of collecting data about individuals and groups, often over the Internet, this idea has recently been discussed in the emerging research area of Surveillance Studies [34] under the term sousveillance. Sousveillance is not tied to any particular software and analysis tool; in fact, in line with the focus on data *collection*, most of the technical attention in this regard is currently focussed on mobile, ubiquitous and wearable devices [35].

*Surveillance* "is the monitoring of the behavior, activities, or other changing information, usually of people and often in a surreptitious manner. It most usually refers to observation of individuals or groups by government organizations." [63]

*Sousveillance* "and inverse surveillance are terms [... that] describe the recording of an activity from the perspective of a participant in the activity. [...] Sousveillance typically involves community-based recording from first person perspectives, without necessarily involving any specific political agenda, whereas inverse-surveillance is a form of sousveillance that is typically directed at, or used to collect data to analyze or study, surveillance or its proponents (e.g., the actions of police at a protest rally)." [62]

Just as privacy mirrors can be made more powerful by data mining (see previous section), sousveillance activities are conceivable that involve not only data collection, but also data analysis. A prime example of this is the Wikiscanner[17], developed by Virgil Griffith: It "allows users to automatically track anonymous edits that people make to Wikipedia entries and trace them to their source. It does so by taking the IP address of the anonymous person who made the Wikipedia changes and identifying who owns the computer network from which the person made the edits.

The tool exposed how insiders at Diebold Election Systems, Exxon, the Central Intelligence Agency and other companies and organizations were surreptitiously deleting or changing information that was unflattering to them or contradicted the company line." [65][18] [19]

[17] `http://wikiscanner.virgil.gr/`

[18] Interestingly, the Wikiscanner is seldom if at all (a Google search on 2010-08-05 returned no results) mentioned in relation to sousveillance, possibly because the data have already been collected.

[19] The Wikiscanner has also been used to identify and classify authors representing different viewpoints, e.g. Israeli and Palestinian authors of jointly interesting pages [13]. It can thus, as long as IP address spaces can roughly be identified with viewpoints, be used to create corpora for tools such as those described in Section 4.2.

**Table 1.** Remarkable Contributions to the German-Language Wikipedia, excerpt[20]

| Article | IP owner (2007) | Date | Kind | Duration of stay in Wikipedia |
|---|---|---|---|---|
| Food and Drug Administration | Fresenius AG | 2004-11-04 | reproach that the US Food and Drug Administration abuses security means as a trade impediment | > 2 years |
| Nuclear power station Biblis | RWE [22] | 2006-06-30 (not only) | "Biblis is a milestone in terms of security" etc. | 10 min |
| Clopidogrel | Sanofi-Aventis | 2007-03-20 | recommendation of S.-A.'s own ingredient, downplaying of risks | 22:42 h |
| The Great Global Warming Swindle | IP of the German Parliament | 2007-04-18 | removal of factual criticism, ascription of lobbying to critics | < 1 min |

An analysis according to the six criteria shows the following.

*1. Situatedness and authenticity, multiple contexts.* All tools apply data analysis to authentic data that are clearly situated as reporting on actions by agents the user is interested in. Many sousveillance activities, in addition, refer to the user's own actions (like keeping a complete log of one's life based on the recorded data). Multiple contexts are supported to the extent that different agents (e.g., governmental or business) and different edited resources are involved.

*2. Active and constructive.* Activity and constructiveness in the Wikiscanner are similar to those in the tools described in Section 4.1: Users can create representations, but they are limited to the given analysis options. More variation appears to be inherent in several sousveillance activities, but this creativity may be limited (at least at present) to relatively computer-/Internet-savvy people.

*3. Multiple perspectives.* The very idea of sousveillance or role reversals in watching and analysing rests on establishing multiple perspectives. Thus, these approaches present the strongest support for multiple perspectives.

*4. Social.* Sousveillance is an excellent example of a use of technology as a social practice, such that the focus on the technology itself would be too narrow. Thus, Mann et al. [35, p. 337] observe that "[t]he goal of the performances reported here is less to understand the nature of surveillance than to engage in dialogues with front-line officials and customer service personnel at the point-of-contact in semi-public and commercial locations." While the performances reported in that paper were deliberate activities at the interface between research and arts performances, the same holds for more "incidental" occurrences of sousveillance:

[20] RWE is a major German electricity/gas/water provider. `http://de.wikipedia.org/w/index.php?title=Wikipedia:WikiScanner&oldid=66534749`, retrieved on 2010-07-30, own translation.

Recording a situation is only part of the sousveillance process; communicating about this is another part. This makes video- and photo-sharing sites such as YouTube or Flickr important media.[21] And Web-based communication underlines the close connection between the two forms of observing: "Social software such as Facebook and MySpace aid surveillance by encouraging people to publish their interests and their friendship networks. They also aid sousveillance by making this information available to peer networks as well as to the authorities." [62]

The Wikiscanner itself has no social functionalities, but it has sparked intense debates (including in the Social Web) and, for example, contributed to the collaborative authoring of Wikiscanner result overviews such as that shown in Fig. 1.[22]

*5. Articulation and reflection.* All activities reported on in this section have a strong component of re-representation and articulation: from turning one's life or perceptions into data, to turning anonymous editors into agents with intentionality. And reflection is the prime motivator: "Sousveillance is a form of 'reflectionism,' a term [...] for a philosophy and procedures of using technology to mirror and confront bureaucratic organizations. Reflectionism holds up the mirror and asks the question: 'Do you like what you see?' If you do not, then you will know that other approaches by which we integrate society and technology must be considered." [35, p. 333]

The goal of reflection, in turn, is social, thus linking our criteria 4. and 5.: "Sousveillance disrupts the power relationship of surveillance when it restores a traditional balance that the institutionalization of Bentham's Panopticon itself disrupted. It is a conceptual model of reflective awareness that seeks to problematize social interactions and factors of contemporary life. It is a model, with its root in previous emancipatory movements, with the goal of social engagement and dialogue." [35, p. 347]

*6. Data mining sophistication.* So far, the sophistication of the *automatic analysis* of the gathered data is limited. Typical sousveillance activities focus on data collection and/or various, manually configured, transformations of these data[23] such as the superimposition of various still and video footage [35] or the transposition

[21] [continued citation] "For example, police agents provocateur were quickly revealed on YouTube when they infiltrated a demonstration in Montebello, Quebec, against the leaders of Canada, Mexico and the United States (August 2007). When the head of the Quebec police publicly stated that there was no police presence, a sousveillance video showed him to be wrong. When he revised his statement to say that the police provocateurs were peaceful observers, the same video showed them to be masked, wearing police boots, and holding a rock." [62]

[22] cf. for example `http://de.wikipedia.org/w/index.php?title=Wikipedia:WikiScanner&oldid=35921623vs.http://de.wikipedia.org/w/index.php?title=Wikipedia:WikiScanner&oldid=35936088`

[23] These transformations are straightforward/non-sophisticated only from the perspective of data mining!

into a "CyborgLog" comprehensive electronic diary, cf. [4]. The Wikiscanner uses a straightforward data lookup and record linkage algorithm.

**Towards Privacy Literacy.** In sum, this and the previous section have shown that in a world in which "people are documents too" by virtue of the traces they leave through many of their online and offline activities and of the compilation efforts performed on these data [9], certain aspects of information literacy as formulated in the 1990s have to be re-evaluated. This can be linked best to Shapiro and Hughes' [52] notion of publishing literacy. This should not only include the "ability to format and publish research and ideas electronically", but also to understand the implications of publishing ideas, whether as an intentional publishing activity in the narrow sense or an unintentional trace-leaving activity, and therefore also the "willingness and ability to *not* publish material on self or others". This should not be designed to make people become paranoid, fear the "privacy nightmare of the Web", and retreat from public life. Rather, it should make people competent players in "privacy as social practice", strategically revealing or hiding information [23].

An interesting question in this context is what role(s) tools such as the Wikiscanner can play in this context. Are they socially liberating tools that expose "villains" who sabotage a common good, or are they invading the privacy of authors who – for whatever reason – wanted to stay anonymous? And what kind of dynamics, in the privacy and other "games", do they cause? Specifically, do they lead to the type of "arms race" that can be observed for example in spamming: as soon as a new method for spam detection is published, spamming methods with a new level of sophistication will be employed, which in turn leads to the development of new methods for their detection, etc.?

## 5 Summary and Conclusions

The motivation for this paper was to argue that data mining can do more than help businesses find the nuggets of knowledge in their customer databases, and more than fuel useful applications such as search engines or recommender systems, that instead it can be a means to help people better understand, reflect and influence information and information-producing and -consuming activities that they are surrounded by in today's "knowledge societies". Understanding *and performing* data mining and other data-analysis activities is, therefore, part of the answer to the question what a person needs to know today to be a full-fledged, competent and literate member of the information/knowledge society.

The answer to that question lies in the knowledge and skills that make up information literacy. Within that wide construct, we chose to focus on critical literacy, the "ability to evaluate critically the intellectual, human and social strengths and weaknesses, potentials and limits, benefits and costs of information technologies" [52] or "[habits] of thought, reading, writing, and speaking which go beneath surface meaning, first impressions, dominant myths, official pronouncements, traditional clichés, received wisdom, and mere opinions, to understand the deep meaning, root causes, social context, ideology, and personal consequences of any action,

event, object, process, organization, experience, text, subject matter, policy, mass media, or discourse" [53]. We argued that this way of regarding critical literacy as a social practice rooted in, and dealing with, language, can and should be extended to a social practice involving (technique) and dealing with (object) data analysis, calling the resulting set of skills and knowledge *critical data literacy*. Like other authors, we regard education as a good setting for acquiring/teaching these skills and knowledge.

We therefore proposed a *structuring framework* for describing resources and settings designed to foster critical literacy: whether the predominant technique is language or data analysis, and what the primary object is (language, data analysis, data-related activities, or other data). We also proposed conformance with five principles of successful learning as well as the sophistication of the used data-analysis / data-mining procedures as *criteria for characterising* any resource or setting, in order to have a basis for assessing learning effectiveness and identifying areas for improvement.

We then described a number of examples of (mostly Web-based) tools representing the four types of objects and data analysis as technique. This showed that "data mining for information literacy" exists and is developing fast, and that recent developments call for a new subcategory of information literacy and/or critical literacy: *privacy literacy*, a construct which itself raises many new questions. It became clear that each example has different foci, strengths and weaknesses, and that these individual efforts are far from constituting a field with common goals or procedures. The analysis also showed that important inspiration for our vision can come from the field of Surveillance Studies and in particular sousveillance frameworks and activities.

This paper is, by design, a high-level vision paper. Much exciting work remains to be done. Major open issues include (a) how to combine the strengths of the examples shown without compounding their weaknesses; (b) how to improve the analysis frame in the light of new findings from the learning sciences; (c) how to progress from the tool-centric view of the present paper to the design of educational activities in formal as well as informal settings; (d) how, within such settings, people of different aspirations, prior knowledge and current learning capacities can be addressed; and (e) how to not get lost in a naïve belief in the exclusive merits of knowing ever more ("if only students were to also use method X, they would *really* understand what's going on"), but also consciously and explicitly deal with the limitations of knowledge-centric beliefs and activities.

## References

1. Ackermann, E.: Piaget's constructivism, Papert's constructionism: What's the difference? In: Constructivism: Uses And Perspectives In Education. Conference Proceedings, pp. 85–94. Research Center in Education, Cahier 8, Geneva (2008), `http://learning.media.mit.edu/content/publications/EA.Piaget%20_%20Papert.pdf`
2. Acquisti, A., Grossklags, J.: Privacy and rationality in individual decision making. IEEE Security & Privacy 3(1), 26–33 (2005)

3. Ryan, S.J.D., Baker, Yacef, K.: The state of educational data mining in 2009: A review and future visions. Journal of Educational Data Mining 1(1), 3–17 (2009)
4. Bell, G., Gemmell, J.: Total Recall: How the E-Memory Revolution Will Change Everything. Penguin Group (2009)
5. Berendt, B.: Lernwege und Metakognition [[Learning paths and metacognition]]. In: Berendt, B., Voss, H.-P., Wildt, J. (eds.) Neues Handbuch Hochschullehre [[New Handbook of Higher Education]], vol. D3.11, pp. 1–34. Raabe Fachverlag für Wissenschaftsinformation, Berlin (2006) (in German)
6. Berendt, B., Brenstein, E.: Visualizing individual differences in Web navigation: STRATDYN, a tool for analyzing navigation patterns. Behavior Research Methods, Instruments, & Computers 33, 243–257 (2001)
7. Berendt, B., Günther, O., Spiekermann, S.: Privacy in e-commerce: Stated preferences vs. actual behavior. Communications of the ACM 48(4), 101–106 (2005)
8. Berendt, B., Preibusch, S., Teltzrow, M.: A privacy-protecting business-analytics service for online transactions. International Journal of Electronic Commerce 12, 115–150 (2008)
9. Berendt, B.: You are a document too: Web mining and IR for next-generation information literacy. In: Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., White, R.W. (eds.) ECIR 2008. LNCS, vol. 4956, pp. 3–3. Springer, Heidelberg (2008)
10. Berendt, B., Krause, B., Kolbe-Nusser, S.: Intelligent scientific authoring tools: Interactive data mining for constructive uses of citation networks. Inf. Process. Manage. 46(1), 1–10 (2010)
11. Berry, M.J.A., Linoff, G.S.: Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management. Wiley, Chichester (2004)
12. Bonneau, J., Preibusch, S.: The privacy jungle: On the market for data protection in social networks. In: Proc. WEIS 2009 (2009), http://preibusch.de/publications/social_networks/privacy_jungle_dataset.htm
13. Borra, E.: Repurposing the wikiscanner (2007) (August 13, 2008), http://wiki.issuecrawler.net/twiki/bin/view/Dmi/WikiScanner
14. Brown, J.S., Collins, A., Duguid, P.: Situated cognition and the culture of learning. Educational Researcher 18(1), 32–42 (1989)
15. Browne, P.: Yahoo pipes could do better, O'Reilly Java Blog, http://www.oreillynet.com/onjava/blog/2007/03/yahoo_pipes_could_do_better.html (retrieved July 30, 2007)
16. Cole, R., Stumme, G.: CEM – a conceptual email manager. In: Ganter, B., Mineau, G.W. (eds.) ICCS 2000. LNCS, vol. 1867, pp. 438–452. Springer, Heidelberg (2000)
17. Dewey, J.: Democracy and education. Free Press, New York (1916)
18. Fang, L., LeFevre, K.: Privacy wizards for social networking sites. In: Proc. 19th International Conference on World Wide Web, WWW 2010 (2010)
19. Fortuna, B., Galleguillos, C., Cristianini, N.: Detecting the bias in media with statistical learning methods. In: Ashok, N., Srivastava, Sahami, M. (eds.) Text Mining: Classification, Clustering, and Applications, Chapman & Hall/CRC Press (2007) (in press)
20. Frankowski, D., Cosley, D., Sen, S., Terveen, L.G., Riedl, J.: You are what you say: privacy risks of public mentions. In: Efthimiadis, E.N., Dumais, S.T., Hawking, D., Järvelin, K. (eds.) SIGIR, pp. 565–572. ACM, New York (2006)
21. Gama, C.A.: Integrating Metacognition Instruction in Interactive Learning Environments. PhD thesis, University of Sussex , http://www.dcc.ufba.br/~claudiag/thesis/Thesis_Gama.pdf (retrieved August 09, 2004)
22. Gilbert, E., Karahalios, K.: Predicting tie strength with social media. In: Proc. CHI 2009 (2009)

23. Gürses, S., Berendt, B.: The social web and privacy: Practices, reciprocity and conflict detection in social networks. In: Ferrari, E., Bonchi, F. (eds.) Privacy-Aware Knowledge Discovery: Novel Applications and New Techniques. Chapman & Hall/CRC Press (2010)
24. Hansen, M.: Linkage control - integrating the essence of privacy protection into identity management. In: eChallenges (2008)
25. Huff, D.: How to Lie with Statistics. Norton, New York (1954)
26. Ishii, K., Lutterbeck, B.: Unexploited resources of online education for democracy: Why the future should belong to opencourseware. First Monday 6(11) (2001), http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/896 (retrieved July 30, 2010)
27. Jameson, A., Smyth, B.: Recommendation to Groups. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 596–627. Springer, Heidelberg (2007)
28. Jonassen, D.: Evaluating constructivist learning. Educational Technology 36(9), 28–33 (1991)
29. Lajoie, S.P., Derry, S.J. (eds.): Computers as cognitive tools. Lawrence Erlbaum, Hillsdale (1993)
30. Lederer, S., Hong, J.I., Dey, A.K., Landay, J.A.: Personal privacy through understanding and personal privacy through understanding and action: Five pitfalls for designers. Personal Ubiquitous Computing 8(6), 440–454 (2004)
31. Lin, W.-H., Xing, E. P., Hauptmann, A.G.: A Joint Topic and Perspective Model for Ideological Discourse. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part II. LNCS (LNAI), vol. 5212, pp. 17–32. Springer, Heidelberg (2008)
32. Liu, B.: Web Data Mining. Exploring Hyperlinks, Contents, and Usage Data. Springer, Berlin (2007)
33. Liu, H., Maes, P., Davenport, G.: Unraveling the taste fabric of social networks. International Journal on Semantic Web and Information Systems 2(1), 42–71 (2006)
34. Lyon, D.: Surveillance Studies: an Overview. Polity Press, Cambridge (2007)
35. Mann, S., Nolan, J., Wellman, B.: Sousveillance: Inventing and using wearable computing devices for data collection in surveillance environments. Surveillance & Society 1(3), 331–355 (2010), http://wearcam.org/sousveillance.pdf (retrieved July 30, 2010)
36. McMahon, M.: Social constructivism and the world wide web – a paradigm for learning. Paper Presented at the ASCILITE Conference (1997), http://www.ascilite.org.au/conferences/perth97/papers/Mcmahon/Mcmahon.html
37. Mladenic, D.: Turning yahoo to automatic web-page classifier. In: ECAI, pp. 473–474 (1998)
38. Nguyen, D.H., Mynatt, E.: Privacy mirrors: Understanding and shaping socio-technical ubiquitous computing. Technical report git-gvu-02-16, Georgia Institute of Technology, USA (2002)
39. Niggemeier, S.: Chronisch krank [chronically ill], http://www.stefan-niggemeier.de/blog/chronisch-krank/ (retrieved July 30, 2010)
40. Nürnberger, A., Klose, A.: Improving clustering and visualization of multimedia data using interactive user feedback. In: Proc. of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2002), pp. 993–999 (2002)
41. Ohlsson, S.: Information processing explanations of insight and related phenomena. In: Keane, M., Gilhooly, K. (eds.) Advances in the Psychology of Thinking, vol. 1, pp. 1–44. Harvester-Wheatsheaf, London (1992)

42. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 2(1-2), 1–135 (2008)
43. Papert, S.: Mindstorms. Children, Computers and Powerful Ideas. Basic Books, New York (1980)
44. Peirsman, Y., Heylen, K., Geeraerts, D.: Applying word space models to sociolinguistics. religion names before and after 9/11. In: Geeraerts, D., Kristiansen, G., Peirsman, Y. (eds.) Advances in Cognitive Sociolinguistics, Cognitive Linguistics Research [CLR], pp. 111–137. De Gruyter, New York (2010)
45. Plaisant, C., Rose, J., Yu, B., Auvil, L., Kirschenbaum, M.G., Smith, M.N., Clement, T., Lord, G.: Exploring erotics in Emily Dickinson's correspondence with text mining and visual interfaces. In: Marchionini, G., Nelson, M.L., Marshall, C.C. (eds.) JCDL, pp. 141–150. ACM, New York (2006)
46. Pollack, S.: Exploratory analysis of press articles on Kenyan elections: a data mining approach. In: Proc. SiKDD (2009)
47. Reeder, R.W.: Expandable Grids: A User Interface Visualization Technique and a Policy Semantics to Support Fast, Accurate Security and Privacy Policy Authoring. PhD thesis, Carnegie Mellon University (2008)
48. Richardson, J.E.: Analysing Newspapers. Palgrave Macmillan, Houndmills (2007)
49. Roll, I., Aleven, V., McLaren, B.M., Koedinger, K.R.: Designing for metacognition – applying cognitive tutor principles to the tutoring of help seeking. Metacognition and Learning 2(2-3), 1556–1623 (2007)
50. Romero, C., Ventura, S., Pechenizkiy, M., Baker, R. (eds.): Handbook of Educational Data Mining. CRC Data Mining and Knowledge Discovery Series. Chapman & Hall, Boca Raton (2010)
51. Sennett, R.: The Fall of Public Man. Knopf, New York (1977)
52. Shapiro, J.J., Hughes, S.K.: Information literacy as a liberal art. Enlightenment proposals for a new curriculum. Educom Review 31(2) (1996)
53. Shor, I.: What is critical literacy? Journal for Pedagogy, Pluralism & Practice 4(1) (1999), http://www.lesley.edu/journals/jppp/4/shor.html
54. Singel, R.: Newly declassified files detail massive FBI data-mining project. Wired, http://www.wired.com/threatlevel/2009/09/fbi-nsac/ (retrieved July 30, 2010 )
55. Subramanian, K.: Microsoft kills yahoo pipes competitor (2009), http://www.cloudave.com/link/microsoft-kills-yahoo-pipes-competitor (retrieved July 30, 2010)
56. Tufte, E.R.: Visual Explanations. Images and Quantities, Evidence and Narrative. Graphics Press, Cheshire (1997)
57. van Halteren, H.: Linguistic profiling for author recognition and verification. In: ACL 2004: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, p. 199. Association for Computational Linguistics, Morristown (2004)
58. van Joolingen, W.: Cognitive tools for discovery learning. International Journal of Artificial Intelligence in Education 10, 385–397 (1999)
59. Van Meter, P., Stevens, R.J.: The role of theory in the study of peer collaboration. The Journal of Experimental Education 69, 113–127 (2000)

60. Verbeke, M., Berendt, B., Nijssen, S.: Data mining, interactive semantic structuring, and collaboration: A diversity-aware method for sense-making in search. In: Proceedings of First International Workshop on Living Web, Collocated with the 8th International Semantic Web Conference (ISWC-2009). CEUR Workshop Proceedings, Washington, D.C, USA, vol. 515 (October 26, 2009), http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-515/
61. Vygotsky, L.S.: Mind in society: The development of higher mental processes. Harvard University Press, Cambridge (1978)
62. Wikipedia contributors. Sousveillance (2010), http://en.wikipedia.org/w/index.php?title=Sousveillance&oldid=375690865 (retrieved August 09, 2010)
63. Wikipedia contributors. Surveillance (2010), http://en.wikipedia.org/w/index.php?title=Surveillance&oldid=376166748 (retrieved August 09, 2010)
64. Woolfolk, A.: Educational Psychology, 11th edn. Merrill, Upper Saddle River (2010)
65. Zetter, K.: Wikiscanner creator releases new tools to uncover anonymous edits. Wired (2008), http://www.wired.com/threatlevel/2008/07/wikiscanner-cre/ (retrieved July 30, 2010)

# Chapter 13
# Rule Extraction from Neural Networks and Support Vector Machines for Credit Scoring

Rudy Setiono[1], Bart Baesens[2], and David Martens[2,3]

[1] School of Computing, National University of Singapore,
13 Computing Drive, Singapore 117417
[2] Department of Decision Sciences and Information Management,
Katholieke Universiteit Leuven,
Naamsestraat 69, B-3000, Leuven, Belgium
[3] Department of Business Administration and Public Management,
Hogeschool Ghent, University Ghent,
Voskenslaan, Ghent, Belgium

**Abstract.** In this chapter we describe how comprehensible rules can be extracted from artificial neural networks (ANN) and support vector machines (SVM). ANN and SVM are two very popular techniques for pattern classification. In the business intelligence application domain of credit scoring, they have been shown to be effective tools for distinguishing between good credit risks and bad credit risks. The accuracy obtained by these two techniques is often higher than that from decision tree methods. Unlike decision tree methods, however, the classifications made by ANN and SVM are difficult to understand by the end-users as outputs from ANN and SVM are computed as nonlinear mapping of the input data attributes. We describe two rule extraction methods that we have developed to overcome this difficulty. These rule extraction methods enable the users to obtain comprehensible propositional rules from ANN and SVM. Such rules can be easily verified by the domain experts and would lead to a better understanding about the data in hand.

## 1 Introduction

Artificial neural networks (ANN) and Support Vector Machines (SVM) are two very popular methods for classification and regression. These methods have been applied to solve pertinent problems in diverse application areas such as business intelligence, bioinformatics, social sciences and engineering. They usually outperform traditional statistical methods such as multiple regression, logit regression, naive Bayesian and linear discriminant analysis in their prediction accuracy (see e.g. [4,21]). They also often produce better results than other machine learning techniques such as genetic algorithm and decision tree methods.

However, both ANN and SVM are essentially black-box techniques; their outputs are computed as a complex function of their input variables. For some applications such as medical diagnosis and credit scoring, a clear explanation of

D.E. Holmes, L.C. Jain (Eds.): Data Mining: Found. & Intell. Paradigms, ISRL 25, pp. 299–320.
springerlink.com 

how the decision is reached by models obtained by these techniques could be a critical requirement and even a regulatory recommendation. In order to explain the internal workings of ANN and SVM, we must be able to extract comprehensible rules that mimic their output predictions. The user may then use these rules to validate their hypotheses as well as to uncover any new fascinating insights that other rule generating methods fail to reveal.

In this chapter, we present methods that we have developed to extract classification rules from ANN and SVM. It should be emphasized from the onset that rule extraction from ANN and SVM only makes sense when the trained models perform better than traditional rule induction techniques such as decision list or decision tree methods. Otherwise, the user is better off simply using the rules induced by the traditional rule induction techniques. There is no reason in this case why the user would take the complicated and cumbersome detour that entails finding the best ANN or SVM models and subsequently applying the rule extraction methods we will present here.

In credit scoring, it is well established that both ANN and SVM could predict new data samples not used to build the models with higher accuracy rates than other techniques. Classification rules can then be obtained by our rule extraction techniques from these ANN and SVM. The classification rules elucidate exactly how credit applications should be labeled as "good credit" or "bad credit" according the values of their relevant input data attributes.

The contents of the subsequent sections of this chapter are organized as follows. In Section 2, we present our method Re-RX for extracting rules from ANN. We also present and discuss the rules that are extracted by this algorithm on two credit scoring data sets. In Section 3, we present ALBA – an active learning based approach for SVM rule extraction. The results obtained by this approach on a data set containing Belgian and Dutch credit risk data is presented in this section. Finally, Section 4 concludes the chapter.

## 2 Re-RX: Recursive Rule Extraction from Neural Networks

### 2.1 Multilayer Perceptron

Artificial neural networks (ANN) are mathematical representations inspired by the functioning of the human brain. Many types of neural networks have been suggested in the literature for both supervised and unsupervised learning [9]. Since our focus is on classification, we will discuss the Multilayer Perceptron neural network in more detail here.

A MLP (Multilayer Perceptron) is typically composed of an input layer, one or more hidden layers and an output layer, each consisting of several units or neurons. Each unit processes its inputs and generates one output value which is transmitted to the unit in the subsequent layer. The decision to classify input samples into various groups is made at the output layer based on the output values of the units in this layer. One of the key characteristics of MLPs is that

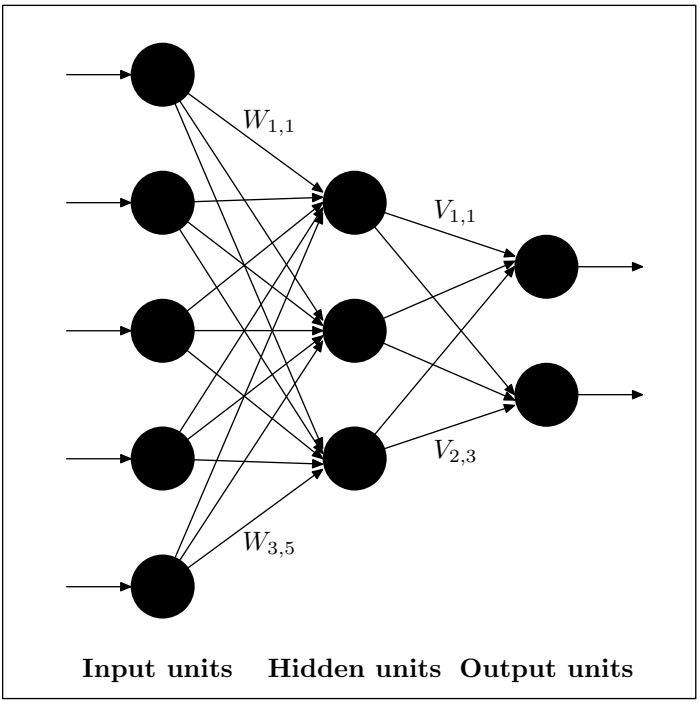



**Fig. 1.** A three-layer feedforward neural network with 5 input units, 3 hidden units and 2 output units. To avoid cluttering, only 2 connections from the input units to the hidden units and 2 connections from the hidden units to the output units are labeled.

all units and layers are arranged in a feedforward manner and no feedback connections are allowed.

Figure 1 depicts an example of an MLP with one hidden layer and two output units for a binary classification problem. Given an $n$-dimensional input data sample $\mathbf{x}_i$, the output of hidden unit $h$ is computed by processing the weighted inputs and its bias term $b_h^{(1)}$ as follows:

$$z_{ih} = f^{(1)}\left(b_h^{(1)} + \sum_{j=1}^{n} \mathbf{W}_{h,j}\mathbf{x}_{ij}\right), \tag{1}$$

where $x_{ij}$ is the $j$-th component of the input vector $\mathbf{x}_i$, $\mathbf{W}$ is a weight matrix whereby $\mathbf{W}_{h,j}$ denotes the weight connecting input $j$ to hidden unit $h$.

Similarly, the output of the output unit $c$ is computed:

$$y_{ic} = f^{(2)}\left(b_c^{(2)} + \sum_{h=1}^{H} \mathbf{V}_{c,h}z_{ih}\right), \tag{2}$$

where $H$ denotes the number of hidden units and $\mathbf{V}$ is a weight matrix whereby $\mathbf{V}_{c,h}$ denotes the weight connecting hidden unit $j$ to output unit $c$.

The bias inputs play a role analogous to that of the intercept term in a classical linear regression model. The class is then assigned according to the output unit with the highest activation value (*winner take all learning*). The transfer functions $f^{(1)}$ and $f^{(2)}$ allow the network to model non-linear relationships in the data. Examples of transfer functions that are commonly used include the sigmoid function:

$$f(x) = \frac{1}{1 + e^{-x}}, \tag{3}$$

the hyperbolic tangent function:

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}, \tag{4}$$

and the linear transfer function:

$$f(x) = x \tag{5}$$

The weights $\mathbf{W}$ and $\mathbf{V}$ are the crucial parameters of a neural network and need to be estimated during a training process which is usually based on gradient-descent learning to minimize some kind of error function over a set of training observations [9]. Note that multiple hidden layers might be used but theoretical works have shown that one hidden layer is sufficient to approximate any continuous function to any desired degree of accuracy (*universal approximation property*) [18].

In order to find the optimal values of the network weights, we minimize the following augmented cross-entropy error function:

$$F(\mathbf{W}, \mathbf{V}) = P(\mathbf{W}, \mathbf{V}) - \sum_{c=1}^{C} \sum_{i=1}^{N} (t_{ic} \log y_{ic} + (1 - t_{ic}) \log(1 - y_{ic})) \tag{6}$$

with

$$P(\mathbf{W}, \mathbf{V}) =$$

$$\epsilon_1 \sum_{h=1}^{H} \sum_{c=1}^{C} \frac{\beta \mathbf{V}_{ch}^2}{1 + \beta \mathbf{V}_{ch}^2} + \sum_{j=1}^{n} \frac{\beta \mathbf{W}_{hj}^2}{1 + \beta \mathbf{W}_{hj}^2} + \epsilon_2 \sum_{h=1}^{H} \sum_{c=1}^{C} \mathbf{V}_{ch}^2 + \sum_{j=1}^{n} \mathbf{W}_{hj}^2, \tag{7}$$

where $C$ is the number of classes and the number of output units, $N$ the number of data points, $n$ the number of inputs, $H$ the number of hidden units, $t_{ic}$ is 1 if observation $i$ belongs to class $c$ and 0 otherwise, and $\epsilon_1$, $\epsilon_2$ and $\beta$ are small positive parameters.

The penalty function $P(\mathbf{W}, \mathbf{V})$ is included in the cost function $F(\mathbf{W}, \mathbf{V})$ to encourage weight decay [17]. Since $f(w) = w^2/(1 + w^2)$ approaches 0 when $w$ is small and 1 when $w$ is large, the first term of $P(\mathbf{W}, \mathbf{V})$ approximates the number of relevant, non-zero weights in the network. The $\beta$ parameter is then added to control how fast the irrelevant weights converge to zero. The larger the $\beta$ parameter, the faster the irrelevant weights will converge to zero. The second

part of $P(\mathbf{W}, \mathbf{V})$ additionally prevents these weights from taking on excessive values. The parameters $\epsilon_1$ and $\epsilon_2$ then reflect the relative importance of the accuracy of the neural network versus its complexity. Typical values for these parameters are: $\beta$=10, $\epsilon_1$=$10^{-1}$ and $\epsilon_2$=$10^{-5}$ [29].

The cost function $F(\mathbf{W}, \mathbf{V})$ is minimized using the BFGS method which is a modified Quasi-Newton algorithm [8,13]. This algorithm converges much faster than the standard backpropagation algorithm and the total error decreases after each iteration step which is not necessarily the case in the backpropagation algorithm [27].

Determining the optimal number of hidden units is not a trivial task. In the literature, two approaches have been suggested to tackle this problem. A growing strategy starts from a network with no or few hidden units and gradually adds hidden units to improve the classification accuracy [27,28]. On the other hand, a pruning strategy starts from an oversized network and removes the irrelevant connections [29]. When all connections to a hidden unit have been removed, the hidden unit can be pruned. The inclusion of the term $P(\mathbf{W}, \mathbf{V})$ into the objective function $F(\mathbf{W}, \mathbf{V})$ of the network allows it to efficiently remove connections based upon the magnitude of the weights. Note that this pruning step plays an important role in both rule extraction algorithms since it will facilitate the extraction of a compact, parsimonious rule set. After having removed one or more connections, the network is retrained and inspected for further pruning.

As we are interested in extracting a rule set having few rules and/or few conditions per rule, it is important that all the irrelevant and redundant network connections are removed from the network. We describe in the next subsection a network pruning method that checks all connections in a network with a small number of hidden units for possible removal.

### 2.2 Finding Optimal Network Structure by Pruning

We start with a network having a fixed number of hidden units $H$. Once this network has been trained, that is, a local minimum of the error function $F(\mathbf{W}, \mathbf{V})$ has been obtained, network pruning may start. Our pruning algorithm identifies network connections for possible removal based on its effect on the network's accuracy if their values are set to zero. The steps of this pruning algorithm are described in Algorithm 1.

The pruning algorithm checks all network connections for possible removal. The candidates for removal are identified by the effect that they have on the accuracy of the network when their individual weights are set to zero. The algorithm first tries to remove the one connection which causes the smallest drop in accuracy. If after retraining the neural network with this particular network connection weight fixed at 0, the accuracy of the network is still acceptable, the iterative pruning process continues. Otherwise, the second candidate is checked for possible removal. This would be the network connection whose weight has the second smallest effect on the overall network accuracy when its weight value is set to 0. If after retraining the network, the network still does not achieve the required accuracy, the next candidate is searched and so on. The pruning

**Algorithm 1.** Neural network pruning algorithm

1: Group the network connections into two subsets: $\mathcal{W}$ and $\mathcal{C}$, these are the sets of network connections that are still present in the network and those that have been checked for possible removal in the current stage of pruning, respectively. Initially, $\mathcal{W}$ corresponds to all the connections in the fully connected trained network and $\mathcal{C}$ is the empty set.
2: Save a copy of the weight values of all connections in the network.
3: Find a connection $w \in \mathcal{W}$ and $w \notin \mathcal{C}$ such that when its weight value is set to 0, the accuracy of the network is least affected.
4: Set the weight for network connection $w$ to 0 and retrain the network.
5: If the accuracy of the network is still satisfactory, then

- Remove $w$, i.e. set $\mathcal{W} := \mathcal{W} - \{w\}$.
- Reset $\mathcal{C} := \emptyset$.
- Go to Step 2.

6: Otherwise,

- Set $\mathcal{C} := \mathcal{C} \cup \{w\}$.
- Restore the network weights with the values saved in Step 2 above.
- If $\mathcal{C} \neq \mathcal{W}$, go to Step 2. Otherwise, Stop.

process continues until all possible network connections have been tested, and the algorithm cannot find one that could be removed.

The reason we design the pruning algorithm to have this "greedy" approach in its search for network connections for possible removal, is that it is possible that the best candidate as measured by its effect on the network's accuracy may not be the best after all. After retraining, it is possible that the network fails to achieve the required accuracy and yet there may still be other network connections that could be removed. The set $\mathcal{C}$ in the algorithm is used to keep track of those connections that have been tested for removal but eventually are not removed at one iteration of the pruning process.

A smaller neural network can be expected to generate rule sets that are simpler, that is, having fewer rules with fewer conditions per rule. In particular, for the credit scoring application we are addressing in this paper, we are interested in finding the smallest subset of data attributes that could still give similar predictive accuracy rates as obtained by other researchers. In order to mitigate the increase in the computation cost for having to retrain the network repeatedly during pruning, we restrict our neural networks to have a small number of hidden unit. As we will be presenting in the next section, it is possible to obtain good accuracy rates for the credit scoring application data sets using networks with as few as one hidden unit.

### 2.3 Recursive Rule Extraction

The algorithm Re-RX is an algorithm for rule extraction from neural network that we have developed recently [30]. This is an algorithm that is specifically designed to handle input data sets having both discrete and continuous attributes.

The novel characteristic of the algorithm lies in its hierarchical nature of considering discrete variables before continuous variables, in a recursive way. Rules are first generated using the relevant discrete attributes, and then refined using the continuous attributes.

The outline of the algorithm is as follows.

**Algorithm 2.** Recursive Rule Extraction (Re-RX) from Neural Networks

1: Train and prune a neural network using the data set $\mathcal{S}$ and all its attributes $\mathcal{D}$ and $\mathcal{C}$.
2: Let $\mathcal{D}^{'}$ and $\mathcal{C}^{'}$ be the sets of discrete and continuous attributes still present in the network, respectively. And let $\mathcal{S}^{'}$ be the set of data samples that are correctly classified by the pruned network.
3: If $\mathcal{D}^{'} = \emptyset$, then generate a hyperplane to split the samples in $\mathcal{S}^{'}$ according to the values of their continuous attributes $\mathcal{C}^{'}$ and stop.
Otherwise using only the values discrete attributes $\mathcal{D}^{'}$, generate the set of classification rules $\mathcal{R}$ for the data set $\mathcal{S}^{'}$.
4: For each rule $\mathcal{R}_i$ generated:
If $support(\mathcal{R}_i) > \delta_1$ and $error(\mathcal{R}_i) > \delta_2$, then

- Let $\mathcal{S}_i$ be the set of data samples that satisfy the condition of rule $\mathcal{R}_i$ and $\mathcal{D}_i$ be the set of discrete attributes that do not appear in rule condition of $\mathcal{R}_i$.
- If $\mathcal{D}_i = \emptyset$, then generate a hyperplane to split the samples in $\mathcal{S}_i$ according to the values of their continuous attributes $\mathcal{C}_i$ and stop.
Otherwise, call Re-RX($\mathcal{S}_i, \mathcal{D}_i, \mathcal{C}_i$).

In Step 1 of the algorithm, any neural network training and pruning method can be employed. The algorithm Re-RX does not make any assumption on the neural network architecture used, but we have restricted ourselves to the backpropagation neural networks with one hidden layer. An effective neural network pruning algorithm is a crucial component of any neural network rule extraction algorithm. By removing the inputs that are not needed for solving the problem, the extracted rule set can be expected to be more concise. In addition, the pruned network also serves to filter noise that might be present in the data. Such noise could be data samples that are outliers or incorrectly labeled. Hence, in Step 2 onward, the algorithm processes only those training data samples that have been correctly classified by the pruned network.

If all the discrete attributes are pruned from the network, then in Step 3 the algorithm generates a hyperplane

$$\sum_{C_i \in \mathcal{C}'} w_i C_i = w_0$$

that separates the two group of samples. The constant $w_0$ and the rest of coefficients $w_i$ of the hyperplane can be obtained by statistical and machine learning methods such as logit regression or support vector machine. In our implementation, we employ a neural network with one hidden hidden unit.

When at least one discrete attribute remains in the pruned network, a set of classification rules involving only the discrete attributes is generated. This step effectively divides the input space into smaller subspaces according to the values of the discrete attributes. Each rule generated corresponds to a subspace, and when the accuracy of the rule is not satisfactory, the subspace is further subdivided by the algorithm Re-RX. The widely-used decision tree generating method C4.5 [25] is applied in to generate the classification rules in this step.

The support of a rule is the number of samples that are covered by that rule. The support and the corresponding error rate of each rule are checked in Step 4. If the error exceeds the threshold $\delta_2$ and the support meets the minimum threshold $\delta_1$, then the subspace of this rule if further subdivided by either calling recursively Re-RX when there are still discrete attributes not present in the conditions of the rule, or by generating a separating hyperplane involving only the continuous attributes of the data.

By handling the discrete and continuous attributes separately, Re-RX generates a set of classification rules that are more comprehensible than rules that have both types of attributes in their conditions. We illustrate the working of Re-RX in detail in the next section.

### 2.4 Applying Re-RX for Credit Scoring

We illustrate how Re-RX works on a publicly available credit approval data set that has often been used in benchmarking studies. The data set is the CARD data set [24], which contains information regarding credit card application with the outcome of the application given as a class label which is either "approved" or "not approved". Three permutations of this data set are available and they are labeled as CARD1, CARD2 and CARD3. Samples in the data set are described by 51 attributes, six of which are continuous and the rest binary. As there is no detailed explanation on what each of the attributes represents, continuous input attributes 4, 6, 41, 44, 49 and 51 are simply labeled $C_4, C_6, C_{41}, C_{44}, C_{49}$, and $C_{51}$, respectively. The remaining binary-valued attributes are $D_1, D_2, D_3, D_5, D_7, \ldots, D_{40}, D_{42}, D_{43}, D_{45}, D_{46}, D_{47}, D_{48}$, and $D_{50}$.

We show how rules are extracted for CARD2 and CARD3.

*A. CARD2 data set*

The pruned neural network for the CARD2 data set depicted in Fig. 2 is selected to illustrate how Re-RX works on a network with more than one hidden unit. We first note that Re-RX falls under the category of pedagogical approaches to neural network rule extraction [34]. A pedagogical algorithm does not explicitly analyze the activation values of the network's hidden units. Instead, it considers the network as a "black box" and directly extracts rules that explain the input-output relation of the network. Hence, Re-RX can be expected to perform efficiently even for more complex networks such as networks with many hidden units or more than one hidden layer.

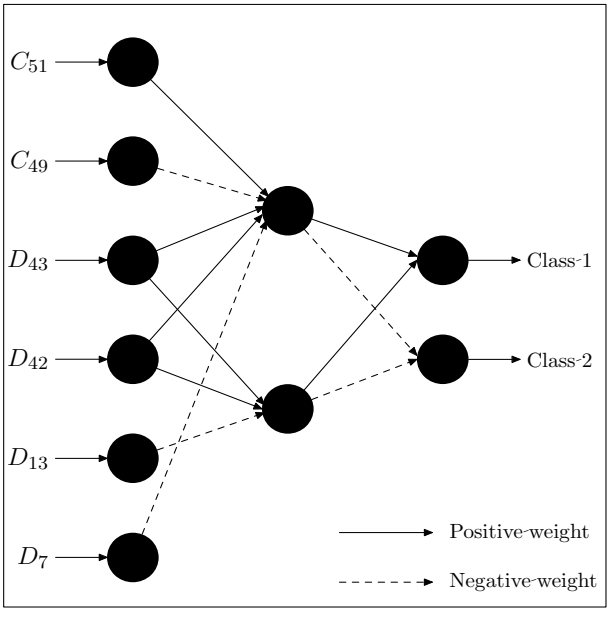



**Fig. 2.** The pruned neural network for the CARD2 data set. Of the initial 3 hidden units and 51 input units in the network, 2 hidden units and 6 input units remain. Its accuracy rates on the training set and test set are 89.38% and 86.05%, respectively.

The neural network obtains accuracy rates of 89.38% and 86.05% on the training and test sets, respectively. Using the 463 correctly classified training samples and the values of the discrete attributes $D_7, D_{12}, D_{13}, D_{42}$ and $D_{43}$, the following rules are obtained:

**Rule $\mathcal{R}_1$:** If $D_7 = 1$ and $D_{42} = 0$, then predict Class 2.
**Rule $\mathcal{R}_2$:** If $D_{13} = 1$ and $D_{42} = 0$, then predict Class 2.
**Rule $\mathcal{R}_3$:** If $D_{42} = 1$ and $D_{43} = 1$, then predict Class 1.
**Rule $\mathcal{R}_4$:** If $D_7 = 1$ and $D_{42} = 1$, then predict Class 1.
**Rule $\mathcal{R}_5$:** If $D_7 = 0$ and $D_{13} = 0$, then predict Class 1.
**Rule $\mathcal{R}_6$:** Default rule, predict Class 2.

All the rules above except rule $\mathcal{R}_4$ achieve 100% accuracy. Of the 57 samples with $D_7 = 1$ and $D_{42} = 1$, 17 samples actually belong to class 2. Applying Re-RX on this subset of 52 samples with input attributes $D_{12}, D_{13}, D_{43}, C_{49}$ and $C_{51}$, we obtain a neural network with one hidden unit and with only the two continuous inputs left unpruned. This network correctly separates all 40 class 1 samples from 17 class 2 samples. Using the connection weights of this network, we refine rule $R_4$ and obtain the following complete set of rules:

**Rule $\mathcal{R}_1$:** If $D_7 = 1$ and $D_{42} = 0$, then predict Class 2.
**Rule $\mathcal{R}_2$:** If $D_{13} = 1$ and $D_{42} = 0$, then predict Class 2.
**Rule $\mathcal{R}_3$:** If $D_{42} = 1$ and $D_{43} = 1$, then predict Class 1.

**Table 1.** Comparison of test set accuracy rates obtained for CARD2

| GA | Prechelt | NeuralWorks | NeuroShell | PNN | Re-RX |
|---|---|---|---|---|---|
| 84.88 % | 81.98 % | 81.98 % | 81.98 % | 86.05 % | 86.63 % |

**Rule** $\mathcal{R}_4$**:** If $D_7 = 1$ and $D_{42} = 1$, then predict Class 1.
**Rule** $\mathcal{R}_{4a}$**:** If $20.23C_{49} - 51.42C_{51} \leq 1.40$, then predict Class 1.
**Rule** $\mathcal{R}_{4b}$**:** Else predict Class 2.
**Rule** $\mathcal{R}_5$**:** If $D_7 = 0$ and $D_{13} = 0$, then predict Class 1.
**Rule** $\mathcal{R}_6$**:** Default rule, predict Class 2.

The accuracy of the rules on the training set is the same as the accuracy of the pruned neural network, that is, 89.38%. On the test data set, the accuracy of the rules is slightly higher at 86.63% compared to the 86.05% rate obtained by the pruned network. In Table 1 we summarize the test set accuracy obtained by our pruned neural network, Re-RX and the other neural network methods as reported by Sexton et al. [31]. In this table, we label our method simply as PNN (Pruned Neural Networks) and Re-RX. The results from GA, NeuralWorks and NeuralShell have been obtained from 10 neural networks, while the Prechelt's results are the best results from training 720 neural networks with various numbers of hidden neurons and hidden layers. GA indicates neural networks trained with the aid of Genetic Algorithm to evolve the best network configuration. NeuralWorks and NeuralShell are commercial neural network packages. We can conclude that our pruned network achieves better predictive accuracy than the other methods, and the rule extracted from this network by Re-RX produces the highest accuracy on this data set.

*B. CARD3 data set*

There are 690 samples in total in the CARD3 data set, consisting of 345 training samples, 173 cross-validation samples, and 172 test samples. The pruned network obtained for the CARD 3 data set depicted in Fig.2.4 has very few connections left and its accuracy rate is better than that achieved by other neural networks reported by Sexton et al. [31]. Of the 518 samples used for training, 87.26% (452 samples) are correctly predicted by the network.

With only the discrete attributes $\mathcal{D}^{'} = \{D_1, D_2, D_{31}, D_{42}, D_{43}\}$ given as input to C4.5, the following set of rules is generated:

**Rule** $\mathcal{R}_1$**:** If $D_{41} = 1$ and $D_{43} = 1$, then predict Class 1.
**Rule** $\mathcal{R}_2$**:** If $D_{31} = 0$ and $D_{42} = 1$, then predict Class 1.
**Rule** $\mathcal{R}_3$**:** If $D_1 = 0$ and $D_{42} = 1$, then predict Class 1.
**Rule** $\mathcal{R}_4$**:** If $D_{42} = 0$, then predict Class 1.
**Rule** $\mathcal{R}_5$**:** If $D_1 = 1$ and $D_{31} = 1$ and $D_{43} = 0$, then predict Class 2.
**Rule** $\mathcal{R}_6$**:** Default rule, predict Class 2.

The number of samples classified by each rule and the corresponding error rates are summarized in Table 2.

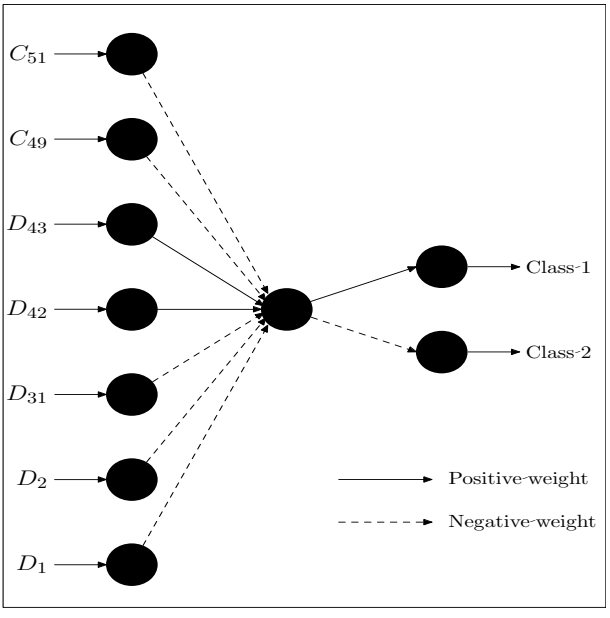



**Fig. 3.** The pruned neural network for the CARD3 data set. It has only one hidden unit, and of the 51 original input units, only 7 units remain. The accuracy rates on the training set and test set are 87.26% and 88.95%, respectively.

As rule $\mathcal{R}_5$ incorrectly classifies the highest number of training data samples, we describe how we refine this rule to improve its accuracy. First, the 23 samples classified by this rule are used to train a new neural network. The input attributes for this network are $D_2, D_{42}, D_{43}, C_{49}$ and $C_{51}$. When the network is pruned, it turns out that only one hidden unit and two inputs $C_{42}$ and $C_{43}$ are still left unpruned. The coefficients of a hyperplane separating class 1 samples from class 2 samples can then be determined from the network connection weights between input units and hidden unit. Based on this, the two classes of samples are separated as follows:

- If $44.65\, C_{49} - 17.29\, C_{51} \leq 2.90$, then predict Class 1,
- Else predict Class 2.

If the thresholds $\delta_1$ and $\delta_2$ were set to 0.05, the algorithm would terminate after generating this rule. For completeness however, let us assume that these parameters are set to zero. This would force Re-RX to generate more rules to refine rules $\mathcal{R}_2, \mathcal{R}_3$ and $\mathcal{R}_4$, and when the algorithm finally terminates, the rules generated would correctly classify all the training samples that have been correctly classified by the original pruned neural network in Fig. 2.4.

**Table 2.** The support level and error rate of the rules generated by C4.5 for the CARD3 data set using only the binary-valued attributes found relevant by the pruned neural network in Figure 2.4

| Rule | # of samples | Correct classification | Wrong classification | Support (%) | Error (%) |
|---|---|---|---|---|---|
| $\mathcal{R}_1$ | 163 | 163 | 0 | 36.06 | 0 |
| $\mathcal{R}_2$ | 26 | 25 | 1 | 5.75 | 3.85 |
| $\mathcal{R}_3$ | 12 | 9 | 3 | 2.65 | 25 |
| $\mathcal{R}_4$ | 228 | 227 | 1 | 50.44 | 0.44 |
| $\mathcal{R}_5$ | 23 | 13 | 10 | 5.09 | 43.48 |
| $\mathcal{R}_6$ | 0 | 0 | 0 | 0 | - |
| All rules | 452 | 437 | 15 | 100 | 3.32 |

The final set of rules generated is as follows:

**Rule $\mathcal{R}_1$:** If $D_{41} = 1$ and $D_{43} = 1$, then predict Class 1.
**Rule $\mathcal{R}_2$:** If $D_{31} = 0$ and $D_{42} = 1$, then
  **Rule $\mathcal{R}_{2a}$:** If $C_{49} \leq 0.339$, then predict Class 1,
  **Rule $\mathcal{R}_{2b}$:** Else predict Class 2.
**Rule $\mathcal{R}_3$:** If $D_1 = 0$ and $D_{42} = 1$, then
  **Rule $\mathcal{R}_{3a}$:** If $C_{49} \leq 0.14$, then predict Class 1,
  **Rule $\mathcal{R}_{3b}$:** Else predict Class 2.
**Rule $\mathcal{R}_4$:** If $D_{42} = 0$, then
  **Rule $\mathcal{R}_{4a}$:** If $D_1 = 0, D_2 = 2, D_{43} = 1$, then predict Class 1,
  **Rule $\mathcal{R}_{4b}$:** Else predict Class 2.
**Rule $\mathcal{R}_5$:** If $D_1 = 1, D_{31} = 1, D_{43} = 0$, then
  **Rule $\mathcal{R}_{5a}$:** If $44.65\, C_{49} - 17.29\, C_{51} \leq 2.90$, then predict Class 1,
  **Rule $\mathcal{R}_{5b}$:** Else predict Class 2.
**Rule $\mathcal{R}_6$:** Default rule, predict Class 2.

The accuracy rates of the above rules and the pruned neural network are summarized in Table 3. Note that with smaller values for $\delta_1$ and $\delta_2$, the accuracy of the rules is exactly the same as the accuracy of the neural network on the training data set. However, the accuracy on the test is slightly lower as there is one sample that is correctly predicted by the network but not by the rules.

Table 4 compares our results with the results from other methods as by Sexton et al. [31]. The accuracy rates of Re-RX with $\delta_1 = \delta_2 = 0.05$ and $\delta_1 = \delta_2 = 0$ are shown under Re-RX 1 and Re-RX 2, respectively. From the figures in the table, we can see that our pruned neural network also achieves the highest predictive test set accuracy on this data set. The network has only one hidden unit and seven input units, and with that, Re-RX is able to generate a simple set of rules that preserves the accuracy of the network. In addition, we believe that the rules are easy to understand as the rule conditions involving discrete attributes are disjoint from those involving continuous attributes, thus allowing different types of conditional expressions to be used for discrete and continuous attributes.

**Table 3.** Accuracy of the pruned network and of the rules extracted by Re-RX for the CARD3 data set

| | Neural network | Re-RX | |
|---|---|---|---|
| | | $\delta_1 = \delta_2 = 0.05$ | $\delta_1 = \delta_2 = 0$ |
| Training set | 87.26% | 86.29% | 87.26% |
| Test set | 88.95% | 88.95% | 88.37% |

**Table 4.** Comparison of test set accuracy rates obtained for CARD3

| GA | Prechelt | NeuralWorks | NeuroShell | PNN | Re-RX 1 | Re-RX 2 |
|---|---|---|---|---|---|---|
| 88.37 % | 81.98 % | 87.79 % | 84.88 % | 88.95 % | 88.95 % | 88.37 % |

## 3 ALBA: Rule Extraction from Support Vector Machines

### 3.1 Support Vector Machine

The Support Vector Machine is a learning procedure based on the statistical learning theory [36]. Given a training set of $N$ data points $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ with input data $\mathbf{x}_i \in \mathbb{R}^n$ and corresponding binary class labels $y_i \in \{-1, +1\}$, the SVM classifier, according to Vapnik's original formulation satisfies the following conditions [12,36]:

$$\begin{cases} \mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}_i) + b \geq +1, & \text{if } \; y_i = +1 \\ \mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}_i) + b \leq -1, & \text{if } \; y_i = -1 \end{cases} \tag{8}$$

which is equivalent to

$$y_i[\mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}_i) + b] \geq 1, \qquad i = 1, ..., N. \tag{9}$$

The non-linear function $\boldsymbol{\varphi}(\cdot)$ maps the input space to a high (possibly infinite) dimensional feature space. In this feature space, the above inequalities basically construct a hyperplane $\mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}) + b = 0$ discriminating between the two classes. By minimizing $\mathbf{w}^T \mathbf{w}$, the margin between both classes is maximized.

In primal weight space the classifier then takes the form

$$y(\mathbf{x}) = \text{sign}[\mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}) + b], \tag{10}$$

but, on the other hand, is never evaluated in this form. One defines the convex optimization problem:

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \mathcal{J}(\mathbf{w}, b, \boldsymbol{\xi}) = \tfrac{1}{2}\mathbf{w}^T \mathbf{w} + C \sum_{i=1}^N \xi_i \tag{11}$$

subject to

$$\begin{cases} y_i[\mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}_i) + b] \geq 1 - \xi_i, & i = 1, ..., N \\ \xi_i \geq 0, & i = 1, ..., N. \end{cases} \tag{12}$$

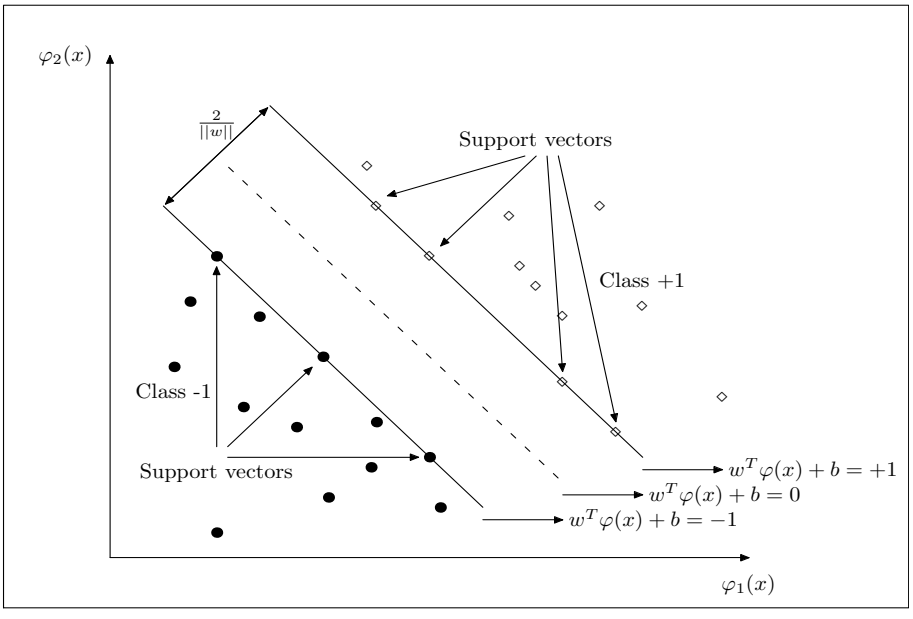



**Fig. 4.** An illustration of SVM optimization of the margin in the feature space. The dashed line is the optimal decision boundary. There are 4 and 3 support vectors from Class +1 and Class -1, respectively.

The variables $\xi_i$ are slack variables which are needed to allow misclassification in the set of inequalities (e.g. due to overlapping distributions). The first part of the objective function tries to maximize the margin between both classes in the feature space, whereas the second part minimizes the misclassification error. The positive real constant $C$ should be considered as a tuning parameter in the algorithm.
The Lagrangian to the constraint optimization problem (11) and (12) is given by

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}; \boldsymbol{\alpha}, \boldsymbol{\nu}) = \mathcal{J}(\mathbf{w}, b, \boldsymbol{\xi}) - \sum_{i=1}^{N} \alpha_i \{y_i[\mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}_i) + b] - 1 + \xi_i\} - \sum_{i=1}^{N} \nu_i \xi_i \qquad (13)$$

with Lagrange multipliers $\alpha_i \geq 0$; $\nu_i \geq 0$ ,$(i = 1, 2, \ldots, N)$.

The solution to the optimization problem is given by the saddle point of the Lagrangian, i.e. by minimizing $\mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}; \boldsymbol{\alpha}, \boldsymbol{\nu})$ with respect to $\mathbf{w}$, $b$, $\boldsymbol{\xi}$ and maximizing it with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{\nu}$.

$$\max_{\boldsymbol{\alpha}, \boldsymbol{\nu}} \min_{\boldsymbol{w}, b, \boldsymbol{\xi}} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}; \boldsymbol{\alpha}, \boldsymbol{\nu}). \qquad (14)$$

This leads to the following classifier:

$$y(\mathbf{x}) = \text{sign}[\textstyle\sum_{i=1}^{N} \alpha_i \, y_i \, K(\mathbf{x}_i, \mathbf{x}) + b], \qquad (15)$$

whereby $K(\mathbf{x}_i, \mathbf{x}) = \boldsymbol{\varphi}(\mathbf{x}_i)^T \boldsymbol{\varphi}(\mathbf{x})$ is taken with a positive definite kernel satisfying the Mercer theorem. The Lagrange multipliers $\alpha_i$ are then determined by means of the following optimization problem (dual problem):

$$\max_{\alpha_i} -\frac{1}{2} \sum_{i,j=1}^{N} y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \alpha_i \alpha_j + \sum_{i=1}^{N} \alpha_i \tag{16}$$

subject to

$$\begin{cases} \sum_{i=1}^{N} \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C, \quad i = 1, ..., N. \end{cases} \tag{17}$$

The entire classifier construction problem now simplifies to a convex quadratic programming (QP) problem in $\alpha_i$. Note that one does not have to calculate $\mathbf{w}$ nor $\boldsymbol{\varphi}(\mathbf{x}_i)$ in order to determine the decision surface. Thus, no explicit construction of the nonlinear mapping $\boldsymbol{\varphi}(\mathbf{x})$ is needed. Instead, the kernel function $K$ will be used. For the kernel function $K(\cdot, \cdot)$, one typically has the following choices:

$$\begin{array}{ll} K(\mathbf{x}, \mathbf{x}_i) = \mathbf{x}_i^T \mathbf{x}, & \text{(linear kernel)} \\ K(\mathbf{x}, \mathbf{x}_i) = (1 + \mathbf{x}_i^T \mathbf{x}/c)^d, & \text{(polynomial kernel)} \\ K(\mathbf{x}, \mathbf{x}_i) = \exp\{-\|\mathbf{x} - \mathbf{x}_i\|_2^2/\sigma^2\}, & \text{(RBF kernel)} \\ K(\mathbf{x}, \mathbf{x}_i) = \tanh(\kappa\, \mathbf{x}_i^T \mathbf{x} + \theta), & \text{(MLP kernel),} \end{array}$$

where $d$, $c$, $\sigma$, $\kappa$ and $\theta$ are constants. Note that for the MLP kernel, the Mercer condition is not always satisfied.

Typically, only few of the training observations will have a non-zero $\alpha_i$ (sparseness property), which are called support vectors and are located close to the decision boundary. This observation is leveraged by the ALBA approach by creating additional data points close to them.

As equation (15) shows, the SVM classifier is a complex, non-linear function. Understanding the logics of the classifications made is very difficult, if not impossible. Comprehensible rules can be extracted from the trained SVM that mimic and hence explain the SVM as much as possible.

### 3.2 ALBA: Active Learning Based Approach to SVM Rule Extraction

In our previously proposed methodology, named ALBA[1], we apply active learning to SVM rule extraction [22]. Active learning entails the control of the learning algorithm over the input data on which it learns [11]. For rule extraction the focus areas are those regions in the input space where most of the noise is present [11]. These regions are found near the SVM decision boundary, which marks the transition of one class to another. The first step in the rule extraction methodology is to change the labels of the data instances by the SVM predicted labels. In this manner the induced rules will mimic the SVM model and all noise

[1] The Matlab code is freely available upon request.

**Algorithm 3.** Active Learning Based Approach (ALBA) for SVM Rule Extraction

Preprocess data $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$
Split data in training data $\mathcal{D}_{tr}$, and test data $\mathcal{D}_{te}$ in a 2/3, 1/3 ratio
Tune SVM parameters with gridsearch on $\mathcal{D}_{tr}$
Train SVM on $\mathcal{D}_{tr} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_{tr}}$, providing an oracle $SVM$ mapping a data input to a class label
Change the class labels of the training data to the SVM predicted class
% *Calculate the average distance* $distance_k$ *of training data to support vectors, in each dimension* $k$
**for** $k$=1 to $n$ **do**
  $distance_k = 0$
  **for all** support vectors $\mathbf{sv_j}$ **do**
    **for all** training data instance $\mathbf{d}$ in $\mathcal{D}_{tr}$ **do**
      $distance_k = distance_k + |d_k - sv_{j,k}|$
    **end for**
  **end for**
  $distance_k = \frac{distance_k}{\#\mathbf{sv} \times N_{tr}}$
**end for**
% *Create 1000 extra data instances*
**for** $i$=1 to 1000 **do**
  Randomly choose one of the support vectors $\mathbf{sv_j}$
  % *Randomly generate an extra data instance* $\mathbf{x_i}$ *close to* $\mathbf{sv_j}$
  **for** $k$=1 to $n$ **do**
    $x_{i,k} = sv(j,k) + \left[(rand - 0.5) \times \frac{distance_k}{2}\right]$ with $rand$ a random number in $[0, 1]$
  **end for**
  Provide a class label $y_i$ using the trained $SVM$ as oracle: $y_i = SVM(\mathbf{x_i})$
**end for**
Run rule induction algorithm on the data set containing both the training data $\mathcal{D}_{tr}$, and newly created data instances $\{(\mathbf{x_i}, y_i)\}_{i=1:1000}$
Evaluate performance in terms of accuracy, fidelity and number of rules, on $\mathcal{D}_{te}$

or class overlap is omitted from the data, removing any apparent conflicts in the data. A second, more intelligent methodology is to incorporate the active learning approach by generating additional, artificial data instances close to the decision boundary. For this we make explicit use of the support vectors, which are typically close to the decision boundary. These are provided with a class label, using the SVM model. So the trained SVM is used both for the data generation process, as well as the data labeling.

This active learning based approach is described formally in Algorithm 3. First, a preprocessing step is needed where the data set is prepared for the data mining algorithm. This includes the removal of all data entries with missing values, the encoding of nominal variables with weights of evidence (WOE). The Weights Of Evidence (WOE) is a logarithmic transformation that allows to

transform a categorical variable into a variable which is monotonically related to the target, and avoids the excessive use of dummy variables [33]. In the case of two classes (Good and Bad), the WOE transformation is defined as follows:

$$\text{Weight Of Evidence}_{attribute} = \ln \frac{\text{P(Good)}_{attribute}}{\text{P(Bad)}_{attribute}} \tag{18}$$

with

$$\text{P(Good)}_{attribute} = \frac{\text{number of good}_{attribute}}{\text{number of good}_{total}} \tag{19}$$

$$\text{P(Bad)}_{attribute} = \frac{\text{number of bad}_{attribute}}{\text{number of bad}_{total}} \tag{20}$$

In the case more data instances for class Good are found for one of the values of *attribute*, the corresponding WOE value will be positive. If more Bads are found, the WOE is negative. By using this transformation, an ordering becomes present (linked to the class variable), with higher weights of evidence meaning less risk.

Two-third of the data samples form the training set and the remaining one-third the test set. Next the SVM model is trained on the training data. We have chosen a RBF kernel for the SVM model, as it is shown to achieve good overall performance [2]. The regularization parameter $C$, and bandwith parameter $\sigma$ are chosen using a gridsearch mechanism [32].

Next, the most crucial step takes place: the generation of extra data instances. A simple, naive method is to generate extra data randomly throughout the input space. Many of these random data instances will provide little to no extra information, as they are generated in regions where no noise was present before. Only the ones close to the decision boundary will further improve the discriminatory performance of the induced rule models. Therefore, we will only add extra data around the support vectors. These extra generated data instances are provided with a class label by the trained SVM model.

Of course, the concept '*near*' needs to be clearly defined. We will measure the average distance of the training data to the support vectors to come to an objective measure of how far a data instance typically lies from a support vector. As this may vary across different dimensions, this average distance is measured for each dimension separately (two in this case). This provides us a ***distance*** vector with the two average distances. Now the '*near*' concept can be defined as half this average distance.

As we generate 1000 extra data instances, an iterative process is repeated 1000 times where each time a new data instance is generated. This is done by first randomly selecting a support vector, in the neighborhood of which (defined by the ***distance*** vector) a new data instance is randomly created. This data instance is provided with a class label by the trained SVM model.

In a final step, a rule induction technique, C4.5 [25] or RIPPER [10] for instance, is applied to the set of training data and extra generated data, all provided with a class label by the trained SVM model. To assess the out-of-sample performance, the performance is evaluated in terms of accuracy, fidelity

and number of rules on the test set. The computational complexity of ALBA still largely remains with the training of the SVM model. The calculation of the distance metric, and the generation and classification of extra data instances merely take a fraction of the SVM training duration.

Employing commonly used rule induction algorithms provides us with many advantages. For instance, other rule extraction techniques infer rules that are overlapping [23], that are not complete [16], depends on the initial setup [20,23], or extract a large number of rules [16,23]. Using mature rule induction techniques overcome these issues, and for instance the trade-off between accuracy and comprehensibility can be tuned simply by adjusting the pruning factor. We will demonstrate this further in the experimental section that follows next. Finally, note that although we use a RBF kernel, the algorithm is just as valid with other kernels (as opposed to for instance the technique by Fung et al., which is only valid for linear kernels).

Although our ALBA methodology relies upon the observation that support vectors are typically found near the decision boundary, we are not only limited to the traditional Vapnik SVM. The support vectors of other variants, such as least-squares SVM [32] and RVM [35] are not located near the decision boundary, but by using the training data for which $\boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}) + b \approx 0$ instead of the support vectors, the same approach is still possible.

### 3.3 Applying ALBA for Credit Scoring

We have experimented ALBA on many benchmark data sets [22]. One of these data sets is the Bene data set containing Belgian and Dutch credit risk data [3]. The date set consists of 3123 samples, each described by 24 attributes. To compare the performance of ALBA with other approaches, we have applied the rule induction techniques (1) on the original data, (2) on the data with class labels changed to the SVM predicted class labels, (3) with the random generation of extra data instances, and finally (4) our ALBA methodology: with the generation of extra data instances, close to the support vectors. Each time 10 runs are conducted of which the average test set performance measures are described in Table 5. The accuracy measures the percentage of correctly classified test instances, fidelity measures the percentage of test instances on which the trained SVM model and the induced rule set agree and thus provides a measure for the extent to which the rule set mimics the SVM model. The number of rules gives an indication of the comprehensibility of the rule sets. In case of multiclass data, a one versus all setup is used for the training of the SVM [19].

In order to empirically determine a proper value for the number of extra data instances to generate, we have conducted the same experiments with 100, 250, 500, 1000, 1500 and 2000 extra data instances. Each time 10 runs are conducted for each data set, the average accuracy and fidelity are computed. Clearly the more data instances that are generated the longer is the computational duration. Therefore a good value should be chosen, such that good performance is achieved,

**Table 5.** Comparison of ALBA with other classification methods

| Appr. | original | | | SVM predicted | | | random extra | | | ALBA | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Acc | Nb R | Fid | Acc | Nb R | Fid | Acc | Nb R | Fid | Acc | Nb R | Fid |
| 1 | *70.19* | 88.00 | *83.94* | **72.07** | 65.90 | **96.50** | **72.10** | 71.10 | **96.32** | **72.14** | 82.20 | **96.63** |
| 2 | *70.52* | 3.60 | *87.93* | **71.74** | 11.60 | **96.07** | **71.82** | 12.40 | **96.11** | **71.94** | 13.20 | **95.45** |
| 3 | *70.19* | 88.00 | *83.77* | **72.01** | 69.20 | **96.32** | **72.01** | 73.90 | 96.15 | **72.14** | 102.10 | **96.76** |
| 4 | *70.52* | 3.60 | *88.11* | 71.61 | 11.50 | **96.25** | **71.88** | 14.10 | **96.18** | 71.89 | 15.10 | **96.11** |

yet with minimal computational requirements. From our experiments, 500 or 1000 seems to be reasonable values, as they provide a good balance between predictive performance and computational cost [22].

The four sets of results shown in Table 5 are obtained from two variants on the number of data instances generated and two rule induction techniques:

- Approach 1: 500 extra instances generated and C4.5 rule induction.
- Approach 2: 500 extra instances generated and Ripper rule induction.
- Approach 3: 1000 extra instances generated and C4.5 rule induction.
- Approach 4: 1000 extra instances generated and Ripper5 rule induction.

The best average test set performance over the 10 randomizations is underlined and denoted in bold face. We then use a paired t-test to test the performance differences. Performances that are not significantly different at the 5% level from the top performance with respect to a one-tailed paired t-test are tabulated in bold face. Statistically significant underperformances at the 1% level are emphasized in italics. Performances significantly different at the 5% level but not at the 1% level are reported in normal script.

Looking at the results, one may conclude that on average the best results are obtained by ALBA, both in term of average accuracy and fidelity as in term of average ranking. The performance improvement is more pronounced for the generation of 1000 extra data instances. One can also observe that the 'SVM predicted' approach, where the class labels of training data are changed according to the SVM predictions, already achieves a better result than when applying the induction technique straight on the original data. This confirms the view that changing class labels can clean up the data, and thereby improve the performance. Once more, we emphasize that this is the case in our setup with data sets where the SVM models are superior in term of accuracy.

Comparing the 'SVM predicted' and 'random extra data' approach, we see rather similar results, with a slight improvement for the 'extra random data' approach. This backs up our statement that generating additional data instances in areas of little or no noise is not useful. Therefore our experiments show that the random generation of data is not suitable, and our methodology of generating data close to the support vectors proves to be a sensible one. Previous SVM rule extraction techniques that also generate extra data, such as Trepan and Iter, do not provide an explanation for the reason of their choice of sampling procedure.

The techniques proposed by Barakat et al. [5,7] only use the support vectors as training data, not employing the other training data, but a specific reason for this approach is missing. The reason is that the most useful (extra) data is located near the decision boundary, in the noisy areas. We provide a clear motivation for this choice that is empirically validated, and take explicit advantage of this insight.

To summarize, our results clearly show that ALBA performs best, followed by the approach with generation of random data, than the approach where class labels are changed by the SVM model, and finally the approach where the techniques are applied directly to the original data.

## 4 Conclusion

Both neural networks and support vector machines have proven to be classification techniques with excellent predictive performance. In many application, the black-box property of the model obtained by these techniques is a major barrier to their wider acceptance. This barrier could be breached if more comprehensible solutions could be found. The most comprehensible classification models being rule sets, ANN and SVM rule extraction tries to combine the predictive accuracy of the trained model with the comprehensibility of the rule set format.

In this chapter, we have presented methods for extracting rules from neural networks and support vector machines. Our pedagogical approach to rule extraction is shown to be able to extract comprehensible classification rule with high fidelity. It is important that the extracted rules achieve high fidelity as we should only extract rules from neural networks or support vector machines that perform better than rule generating methods such as C4.5. Rules with high fidelity would then preserve the high accuracy of the neural network or support vector machine from which the rules are extracted. These rules could be verified by domain experts and may provide new insights into the data.

## References

1. Andrews, R., Diederich, J., Tickle, A.B.: A survey and critique of techniques for extracting rules from trained neural networks. Knowledge Based Systems 8(6), 373–389 (1995)
2. Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., Vanthienen, J.: Benchmarking state-of-the-art classification algorithms for credit scoring. Journal of the Operational Research Society 54(6), 627–635 (2003)
3. Baesens, B., Setiono, R., Mues, C., Vanthienen, J.: Using neural network rule extraction and decision tables for credit risk evaluation. Management Science 49(3), 312–329 (2003)
4. Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., Vanthienen, J.: Benchmarking state-of-the-art classification algorithms for credit scoring. Journal of the Operational Research Society 54(6), 627–635 (2003)
5. Barakat, N., Diederich, J.: Eclectic rule extraction from support vector machines. International Journal of Computational Intelligence 2(1), 59–62 (2005)

6. Barakat, N.H., Bradley, A.P.: Rule extraction from support vector machines: Measuring the explanation capability using the area under the ROC curve. In: Proc. of ICPR, vol. (2), pp. 812–815. IEEE Computer Society, Los Alamitos (2006)
7. Barakat, N.H., Bradley, A.P.: Rule extraction from support vector machines: A sequential covering approach. IEEE Transactions on Knowledge and Data Engineering 19(6), 729–741 (2007)
8. Battiti, R.: First- and second-order methods for learning: Between steepest descent and Newton's method. Neural Computation 4, 141–166 (1992)
9. Bishop, C.M.: Neural networks for pattern recognition. Oxford University Press, Oxford (1995)
10. Cohen, W.W.: Fast effective rule induction. In: Proc. of the 12th International Conference on Machine Learning, pp. 115–123 (1995)
11. Cohn, D., Atlas, L., Ladner, R.: Improving generalization with active learning. Machine Learning 15(2), 201–221 (1994)
12. Cristianini, N., Shawe-Taylor, J.: An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, New York (2000)
13. Dennis Jr., J.E., Schnabel, R.E.: Numerical methods for unconstrained optimization and nonlinear equations. Prentice Halls, Englewood Cliffs (1983)
14. Downs, T., Gates, K.E., Masters, A.: Exact Simplification of support vector solutions. Journal of Machine Learning Research 2, 293–297 (2001)
15. Fawcett, T.: PRIE: A system for generating rulelists to maximize ROC performance. Data Mining and Knowledge Discovery 17(2), 207–224 (2008)
16. Fung, G., Sandilya, S., Rao, R.B.: Rule Extraction from linear support vector machines. In: Proc. 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, pp. 32–40 (2005)
17. Hertz, J., Krogh, A., Palmer, R.G.: Introduction to the theory of neural computation. Addison-Wesley, Redwood City (1991)
18. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Networks 2, 359–366 (1989)
19. Hsu, C.-W., Lin, C.-J.: A comparison of methods for multi-class support vector machines. IEEE Transactions on Neural Networks 13, 415–425 (2002)
20. Huysmans, J., Baesens, B., Vanthienen, J.: ITER: An algorithm for predictive regression rule extraction. In: Tjoa, A.M., Trujillo, J. (eds.) DaWaK 2006. LNCS, vol. 4081, pp. 270–279. Springer, Heidelberg (2006)
21. Lessmann, S., Baesens, B., Mues, C., Pietsch, S.: Benchmarking classification models for software defect prediction: A proposed framework and novel findings. IEEE Transactions Software Engineering 34(4), 485–496 (2008)
22. Martens, D., Van Gestel, T., Baesens, B.: Decompositional rule extraction from support vector machines by active learning. IEEE Transactions on Knowledge and Data Engineering 21(2), 178–191 (2009)
23. Nùñez, H., Angulo, C., Català, A.: Rule extraction from support vector machines. In: Proc. European Symposium on Artificial Neural Networks (ESANN), pp. 107–112 (2002)
24. Prechelt, L.: PROBEN1 - A set of benchmarks and benchmarking rules for neural network training algorithms. Technical Report 21/94, Fakultät für Informatik. Universität Karlsruhe, Germany. Anonymous ftp, `ftp://pub/papers/techreports/1994/1994021.ps.gz` on `ftp.ira.uka.de`
25. Quinlan, R.: C4.5: Programs for machine learning. Morgan Kaufman, San Mateo (1993)

26. Saar-Tsechansky, M., Provost, F.: Decision-centric active learning of binary-outcome models. Information Systems Research 18(1), 4–22 (2007)
27. Setiono, R.: A neural network construction algorithm which maximizes the likelihood function. Connection Science 7(2), 147–166 (1995)
28. Setiono, R., Hui, L.C.K.: Use of quasi-Newton method in a feedforward neural network construction algorithm. IEEE Transactions on Neural Networks 6(2), 326–332 (1995)
29. Setiono, R.: A penalty function approach for pruning feedforward neural networks. Neural Computation 9(1), 185–204 (1997)
30. Setiono, R., Baesens, B., Mues, C.: Recursive neural network rule extraction for data with mixed attributes. IEEE Transactions on Neural Networks 19(2), 299–307 (2008)
31. Sexton, R.S., McMurtrey, S., Cleavenger, D.J.: Knowledge discovery using a neural network simultaneous optimization algorithm on a real world classification problem. European Journal of Operational Research 168, 1009–1018 (2006)
32. Suykens, J.A.K., Van Gestel, T., De Brabanter, J., De Moor, B., Vandewalle, J.: Least squares support vector machines. World Scientific, Singapore (2003)
33. Thomas, L., Edelman, D., Crook, J.: Credit scoring and its applications. SIAM, Philadelphia (2002)
34. Tickle, A.B., Andrews, R., Golea, M., Diederich, J.: The truth will come to light: Directions and challenges in extracting the knowledge embedded within trained artificial neural networks. IEEE Transactions on Neural Networks 9(6), 1057–1068 (1998)
35. Tipping, M.: Sparse bayesian learning and the relevance vector machine. Journal of Machine Learning Research 1, 211–244 (2001)
36. Vapnik, V.N.: The nature of statistical learning theory. Springer, New York (1995)

# Chapter 14
# Using Self-Organizing Map for Data Mining: A Synthesis with Accounting Applications

Andriy Andreev and Argyris Argyrou

Hanken School of Economics, Arkadiankatu 22, 00101 Helsinki, Finland
{andriy.andreev,argyris.argyrou}@hanken.fi

**Abstract.** The self-organizing map (i.e. SOM) has inspired a voluminous body of literature in a number of diverse research domains. We present a synthesis of the pertinent literature as well as demonstrate, via a case study, how SOM can be applied in clustering accounting databases. The synthesis explicates SOM's theoretical foundations, presents metrics for evaluating its performance, explains the main extensions of SOM, and discusses its main financial applications. The case study illustrates how SOM can identify interesting and meaningful clusters that may exist in accounting databases. The paper extends the relevant literature in that it synthesises and clarifies the salient features of a research area that intersects the domains of SOM, data mining, and accounting.

## 1 Introduction

This paper aims to present a coherent synthesis of the literature pertinent to self-organizing map (SOM; Kohonen, 1982, 1997) as well as demonstrate how SOM can be applied as a data-mining tool in the domain of accounting. The motivation behind the paper emanates from the considerable academic interest and body of literature SOM has inspired in a multitude of research domains; until 2005, SOM has paved the way for 7,718 publications[1]. The paper differs from previous reviews of SOM (Kohonen, 1998, 2008), and (Yin, 2008), in that it addresses SOM from a data-mining perspective, and places much emphasis on the main financial applications of SOM. The contribution and novelty of the paper lie in it synthesising an expansive and fragmented literature pertinent to SOM, focusing on how SOM can perform certain data-mining tasks, and demonstrating the performance of such tasks via a case study that considers the clustering of accounting databases.

In essence, SOM performs a non-linear projection of a multi-dimensional input space to a two-dimensional regular grid that consists of spatially-ordered neurons, and preserves the topology of the input space as faithfully as possible. SOM has been applied successfully in numerous research domains for clustering, data visualization, and feature extraction.

To carry out the synthesis, we adopt the following four organizing principles. First, we select and review papers that explain the SOM algorithm in sufficient details, and exclude those papers that delve into the mathematical complexities and subtleties of

[1] A complete bibliography is available at: http://www.cis.hut.fi/research/som-bibl/

D.E. Holmes, L.C. Jain (Eds.): Data Mining: Found. & Intell. Paradigms, ISRL 25, pp. 321–342.
springerlink.com 

SOM. Further, we review only the most prevalent criteria that evaluate the performance of SOM, because there is neither an accepted global criterion nor a consensus about which criteria are the most informative. As the literature abounds with extensions of SOM, we delimit the synthesis to those extensions that enhance the ability of SOM to perform data-mining tasks (e.g. clustering of non-vectorial data). Finally, to review the financial applications of SOM, we pay particular attention to the subjects of bankruptcy prediction, financial benchmarking, and clustering of hedge funds.

To conduct the case study, the paper uses a set of accounting transactions that describe the economic activities of an international shipping company for fiscal year 2007. It first pre-processes the accounting transactions for SOM-based processing, and then uses bootstrap to select random samples with replacement from the empirical distribution of the transactions. For each bootstrapped sample, the paper trains a SOM, and subsequently evaluates the performance of each SOM by calculating three metrics: (i) quantization error, (ii) topographic error, and (iii) Davies-Bouldin index. Finally, it estimates the two-sided 95% confidence interval of the mean and standard deviation of the foregoing metrics.

The rest of the paper is organized as follows. Section 2 introduces data pre-processing; an activity that precedes most of data-mining tasks. Section 3 elaborates on the SOM algorithm and its main constituents, and Section 4 presents three metrics for evaluating the performance of SOM as well as a criterion for assessing the internal validity of clustering. Section 5 discusses the main extensions of SOM, and Section 6 reviews the main financial applications of SOM. Section 7 demonstrates, via a case study, how SOM can be applied in identifying and visualizing meaningful clusters that may exist in accounting databases.

## 2 Data Pre-processing

For SOM to operate efficiently and yield meaningful results, a researcher must pay attention to data pre-processing; an activity that involves three main tasks: (i) understanding the different types of variables, (ii) selecting an appropriate and valid distance metric, and (iii) rescaling input variables.

### 2.1 Types of Variables

Understanding the various types of variables guides a researcher in selecting mathematical and statistical operations that are valid for each type as well as in choosing permissible transformations that preserve the original meaning of variables.

Four types of variables can be identified, as follows (Stevens, 1946): (i) nominal, (ii) ordinal, (iii) interval, and (iv) ratio. The order is cumulative, which means that each type subsumes the properties of its predecessor. Nominal variables take as values different names or labels (e.g. name of employees), and hence they are not amenable to any mathematical operation other than a simple function of equality. Further, ordinal or hierarchical variables can be ranked by order (e.g. examination grades, hardness of minerals). Interval variables take values whose differences are meaningful, but ratios are not. The reason being that interval variables lack a “true” zero point; a zero value does not entail the absence of a variable (e.g. temperature in Celsius). In contrast, ratio

variables (e.g. length) take values whose differences, and ratios are meaningful. For completeness, nominal and ordinal variables are collectively described as categorical data, whereas interval and ratio variables as numerical data.

A further distinction can be made between discrete and continuous data; the former is associated with counting and can take a value from a finite set of real integers, whereas the latter is associated with physical measurements and thus it can take any numerical value within an interval.

### 2.2 Distance Metrics

Selecting a valid distance metric takes on added importance in an unsupervised-learning task (e.g. clustering), because in performing such task, a researcher has no recourse to information concerning the class labels of input data. On the other hand, in a supervised-learning task (e.g. classification), the classification error can form an external criterion that can be optimised to yield a valid distance metric. A distance metric over $\mathbb{R}^n$ is considered to be valid only if it can assign distances that are proportionate to the similarities between data points.

In particular, given three vectors $x$, $y$, $z \in \mathbb{R}^n$, a distance metric $d(x,y)$ must satisfy the conditions of a metric space, as follows (Jungnickel, 2002, p.65):

1. $d(x,y) > 0$ for all $x \neq y$ (non-negativity),
2. $d(x,y) = 0$ if and only if $x = y$ (distinguishability),
3. $d(x,y) = d(y,x)$ for all $x$ and $y$ (symmetry),
4. $d(x,z) \leq d(x,y) + d(y,z)$ for all $x$, $y$, $z$ (triangular inequality).

In a metric space, the most prevalent distance metric is the Euclidean distance defined as:

$$d_E(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \ . \tag{1}$$

Further, Mahalanobis distance takes into account the correlations between data, and hence it can yield more accurate results than Euclidean distance does. However, any benefits derived from the improved accuracy may be outweighed by the computational costs involved in calculating the covariance matrix.

$$d_M(x,y) = \sqrt{(x-y)\,S^{-1}\,(x-y)^T} \ . \tag{2}$$

Mahalanobis distance reduces to the Euclidean if the covariance matrix, $S$, is equal to the identity matrix, and to the normalized Euclidean if $S$ is diagonal.

### 2.3 Rescaling Input Variables

The rationale behind rescaling input variables is threefold (Bishop, 1995, p.298): (i) to ensure that input variables reflect their relative importance, (ii) different variables may have different units of measurement, and hence their typical values may differ considerably, and (iii) to facilitate the initialisation of codevectors. For example, consider a

dataset having two variables: "weight" and "height". The former takes values in the range $\{30Kg, 40Kg, \ldots, 100Kg\}$, whereas the latter in the range $\{1.3m, 1.4m, \ldots, 2m\}$; without rescaling, the "weight" variable is going to dominate the distance calculations, and thus it would bias SOM. Linear and non-linear operations can rescale input variables so that they could take values in the interval $[0, 1]$, or values having zero mean and unit variance. The latter is particularly appropriate when the Euclidean distance is used as a distance metric.

## 3 Self-Organizing Map

### 3.1 Introduction to SOM

SOM performs two closely-coupled operations: (i) vector quantization, and (ii) non-linear projection. First, SOM regresses a set of codevectors into input data in a non-parametric and non-linear manner. In doing so, SOM partitions input data into a finite number of disjoint and contiguous spaces each of which is represented by a codevector. The codevectors are estimated in such a way that minimises the distance between an input vector and its closest codevector; this estimation process continues iteratively until it converges to a stationary state at which codevectors are not updated any further. The vector-quantization operation reduces input data to a much smaller, albeit representative, set of codevectors.

Second, SOM projects the codevectors onto a regular two-dimensional grid of neurons; a grid can be either hexagonal or rectangular depending on whether a neuron has either six or four neighbours. In either case, the neurons are spatially-ordered, and thus the grid can preserve the neighbourhood relations or topology between input data as faithfully as possible. A neuron $k$ is described by the tuple $(m_k, p_k)$, where $m_k \in \mathbb{R}^{\mathrm{d}}$ is a codevector, $p_k \in \mathbb{R}^2$ is a location vector on the SOM grid, $d$ denotes the dimensionality of input space, and $k = 1, 2, \ldots, K$ represents the number of neurons. The codevectors are used for vector-quantization, and their corresponding locations on the SOM grid for vector-projection.

To elaborate, given an input dataset, $X = (x_{ij})_{nd}$, SOM constructs a set of codevectors, $M = (m_{kj})_{Kd}$, where $x_i$, $m_k \in \mathbb{R}^{\mathrm{d}}$ are row vectors of $X$ and $M$ respectively, and $n$ represents the number of input vectors. The number of neurons can be estimated by using the heuristic formula: $K = 5\sqrt{n}$ (Vesanto et al, 2000, p.30). As a first approximation to input data, SOM can initialise the codevectors by using either random vectors or vectors derived from the hyperplane that is spanned by the two largest principal components of input data (Kohonen, 1998). Although SOM is very robust to either of the two initialisation approaches, the latter approach is preferable, because it enables SOM to converge more efficiently than random initialisation does. The reason for this is twofold: (i) SOM gets organized approximately from the beginning, and (ii) a researcher can start with a narrower neighbourhood kernel and a smaller learning rate (Kohonen, 1998).

### 3.2 Formation of SOM

The formation of SOM involves the following three iterative processes: (i) competition, (ii) co-operation, and (iii) adaptation (Haykin, 1999, p.447). First, in the competition

process, each input vector, $x_i \in \mathbb{R}^d$, is compared with all codevectors, $m_k \in \mathbb{R}^d$, and the best match in terms of the smallest Euclidean distance, $\| x_i - m_k \|$, is mapped onto neuron $k$ that is termed the best-matching unit (i.e. BMU) and denoted by the subscript $c$ (Kohonen, 1997, p.86):

$$\| x_i - m_c \| = \min_k \{ \| x_i - m_k \| \} \quad , \tag{3}$$

or equivalently:

$$c = \underset{k}{\operatorname{argmin}} \{ \| x_i - m_k \| \} \quad . \tag{4}$$

Second, in the co-operation process, the BMU locates the centre of a neighbourhood kernel, $h_{ck}(t)$, which is usually a Gaussian function defined as:

$$h_{ck}(t) = exp\left[ -\frac{\| p_c - p_k \|^2}{2\sigma^2(t)} \right] \quad , \tag{5}$$

where $p_c$, $p_k \in \mathbb{R}^2$ are the location vectors of BMU and neuron $k$ respectively, $t$ denotes discrete time, and $\sigma(t)$ defines the width of the kernel; $\sigma(t)$ is a monotonically decreasing function of time (Kohonen, 1997, p.87). The motivation behind the neighbourhood kernel is that an activated neuron, BMU, excites adjacent neurons to a greater degree than that to which it excites distant neurons. It can be deduced from Eq. 5 that the neighbourhood kernel decays proportionately with discrete time $t$ as well as with the lateral distance between the BMU and the "excited" neurons, that is: $h_{ck}(t) \to 0$ as $\| p_c - p_k \|$ increases.

The neighbourhood kernel enables the SOM grid to preserve the topology or neighbourhood relations between input data by allowing the codevectors to be updated according to their respective proximity to the BMU; the closer to the BMU a codevector is, the greater the extent of its updating is, whereas codevectors lying outside the neighbourhood of BMU are not updated at all. As a result, the neurons on the SOM grid become spatially-ordered in the sense that neighbouring neurons have similar codevectors, and thus they represent similar areas in the input space.

A SOM grid enjoys two main properties (Ritter and Kohonen, 1989): (i) it preserves the neighbourhood or topological relations between input data as faithfully as possible, and (ii) it represents a mapping of input vectors that is determined by their density function, whereby more frequent vectors are mapped to a larger area on the grid. Because of these properties, SOM can map similar and high-frequent input vectors to a localised area, capture the overall topology of cluster arrangements, and perform nonlinear dimensionality-reduction.

Third, in the adaptive process, the sequence-training SOM updates recursively codevector $m_k$ as follows:

$$m_k(t+1) = m_k(t) + a(t) h_{ck}(t) [x_i(t) - m_k(t)] \quad , \tag{6}$$

where $0 < a(t) \leqslant 1$ is a learning rate at discrete time $t$; $a(t)$ is a non-increasing function of time, for example: $a(t) = a_0 \left(1 - \frac{t}{T}\right)$ or $a(t) = a_0 \left(\frac{0.005}{a_0}\right)^{\frac{t}{T}}$, where $a_0$ is the initial

learning rate, and $T$ is the training length. The updating rule, described in Eq. 6, is motivated by the Hebbian law for synaptic modification, and includes a non-linear forgetting process for synaptic strength (Ritter and Kohonen, 1989).

Batch-SOM is a variant of the updating rule, described in Eq. 6. It estimates the BMU according to Eq. 3, but updates the codevectors only at the end of each epoch, which is a complete presentation of input data, rather than recursively, as follows (Vesanto et al, 2000, p.9):

$$m_k(t+1) = \frac{\sum_{i=1}^{n} h_{ck}(t)x_i}{\sum_{i=1}^{n} h_{ck}(t)} \quad . \tag{7}$$

Batch-SOM does not contain a learning rate, and estimates codevectors as averages of input vectors being weighed by neighbourhood kernel $h_{ck}(t)$. It can also be expressed in terms of Voronoi cells: $V_k = \{x_i \mid \| x_i - m_k \| < \| x_i - m_j \| \; \forall k \neq j\}$, as follows (Vesanto et al, 2000, p.11):

$$m_k(t+1) = \frac{\sum_{k=1}^{K} h_{ck}(t)s_k(t)}{\sum_{k=1}^{K} h_{ck}(t)N_k} \quad , \tag{8}$$

where $K$ is the number of codevectors, and hence that of Voronoi cells, $N_k$ and $s_k(t) = \sum_{x \in V_k} x$ denote the number and sum of input vectors that belong in Voronoi cell $V_k$, respectively.

Batch-SOM enjoys certain advantages vis-a-vis sequence-SOM (Lawrence et al, 1999): (i) it is less computationally expensive, (ii) because the updating of codevectors is not recursive, there is no dependence on the order in which input vectors appear, and (iii) it mitigates concerns that input vectors that appear at a later iteration may affect the result disproportionately.

SOM converges to a stationary state when the codevectors do not get updated any further. This entails that at a stationary state we require that $E\{m_k(t+1)\}$ must be equal to $E\{m_k(t)\}$ for $t \to \infty$, even if $h_{ck}(t)$ was non-zero (Kohonen, 1998). The foregoing leads to the convergence criterion: $E\{h_{ck}(x_i - \lim_{t\to\infty} m_k(t))\} = 0$, where $E\{.\}$ denotes the expectation function (Kohonen, 1997, p.113).

## 4 Performance Metrics and Cluster Validity

In the general case, the updating rule, described in Eq. 6, is not the gradient of any cost function that can be optimised for SOM to converge to at least a local optimum (Erwin et al, 1992). The lack of a cost function has hindered researchers in their efforts to derive a global criterion that can assess the quality and performance of SOM. In the absence of a global criterion, we confine the discussion to the following three criteria, each evaluating a distinct facet of SOM: (i) Quantization error, Q.E, (ii) Topographic error, T.E, and (iii) Distortion error, D.E.

First, Q.E quantifies the resolution of a SOM grid. A som grid exhibits high resolution if the input vectors that are near to one another are projected nearby on the SOM grid, and the input vectors that farther apart are projected farther apart on the SOM grid. The average Q.E for a SOM grid is defined as: $Q.E = \frac{1}{n}\sum_{i=1}^{n} \| x_i - m_c \|$; and the lower its value is, the higher the SOM's resolution is. Second, T.E quantifies the topology preservation of SOM mapping as follows: $T.E = \frac{1}{n}\sum_{i=1}^{n} \varepsilon(x_i)$; if the first and

second BMUs of $x_i$ are adjacent, then $\varepsilon(x_i) = 0$, otherwise $\varepsilon(x_i) = 1$ (Kiviluoto, 1996). Further, T.E indicates whether a mapping from $\mathbb{R}^d$ to $\mathbb{R}^2$ is continuous; if the first and second BMUs of an input vector are not adjacent on the SOM grid, then the mapping is not continuous near that input vector.

Although Q.E and T.E are the most prevalent performance metrics, a number of caveats apply to their interpretation. First, the two metrics are not independent, instead there is a trade-off between them being moderated by the size of the kernel width (i.e. $\sigma(t)$). Second, Q.E and T.E are dependent on input data; a feature that precludes their use from comparing SOMs that are trained on different datasets. Third, good results may be achieved by SOM overfitting the data rather than it enjoying high resolution and preserving topology faithfully. Finally, Q.E takes its minimum value when the neighbourhood kernel, $h_{ck}(t)$, becomes equivalent to Kronecker's delta, that is: $h_{ck}(t) = \begin{cases} 1 & for \quad c = k, \\ 0 & for \quad c \neq k, \end{cases}$. However, in that case SOM reduces to the classical k-means algorithm, and thus it does not possess any self-organizing capabilities.

Further, the distortion error can act as a local cost function, provided that the input data are discrete and the neighbourhood kernel is fixed (Vesanto et al, 2003). Given these assumptions, the sequence-training rule in Eq.6 estimates approximately the gradient of the distortion error (Graepel et al, 1997). The distortion error is defined as:

$$D.E = \sum_{i=1}^{n} \sum_{j=1}^{m} h_{cj} (x_i - m_j)^2 \quad , \tag{9}$$

and it can be decomposed into the following three terms (Vesanto et al, 2003):

$$D.E = E_{qx} + E_{nb} + E_{nv} \quad . \tag{10}$$

The term $E_{qx}$ measures the quantization quality of SOM in terms of the variance of input data that belong in each Voronoi set; $E_{nb}$ can be interpreted as the stress or link between the quantizing and ordering properties of SOM; and $E_{nv}$ quantifies the topological quality of SOM. Analytically,

1. $E_{qx} = \sum_{j=1}^{m} N_j H_j Var\{x|j\}$ ,
2. $E_{nb} = \sum_{j=1}^{m} N_j H_j \parallel n_j - \bar{m}_j \parallel^2$ ,
3. $E_{nv} = \sum_{j=1}^{m} N_j H_j Var_h\{m|j\}$ ,

where $Var\{x|j\} = \sum_{x \in V_j} \parallel x - n_j \parallel^2 / N_j$ is the local variance, $Var_h\{m|j\} = \sum_k h_{jk} \parallel m_k - \bar{m}_j \parallel^2 / H_j$ is the weighed variance of the codevectors, and $\bar{m}_j = \sum_k h_{jk} m_k / H_j$ is the weighed mean of the codevectors.

Although the aforesaid criteria have been used extensively in evaluating the performance of SOM, they do not provide any information about the validity of clustering. For this reason, we describe the Davies-Bouldin index (Davies and Bouldin, 1979) that evaluates the internal validity of clustering as follows:

$$DBI = \frac{1}{C} \sum_{i=1}^{C} \max_{i \neq j} \left\{ \frac{\Delta(C_i) + \Delta(C_j)}{\delta(C_i, C_j)} \right\} \quad , \tag{11}$$

where $C$ is the number of clusters produced by SOM, $\delta(C_i, C_j)$ denotes inter-cluster distances, and $\Delta(C_i)$ and $\Delta(C_j)$ represent intra-cluster distances. A small value indicates highly-compact clusters whose centroids are well-separated.

# 5 Extensions of SOM

SOM has been extended so that it could address data-mining tasks in the following domains: (i) non-metric spaces, (ii) temporal sequence processing, (iii) clustering, and (iv) visualizing high-dimensional data.

## 5.1 Non-metric Spaces

In the domain of non-metric spaces, extensions of SOM include models for clustering text documents, symbol strings, categorical data, and hierarchical data.

### 5.1.1 WEBSOM

WEBSOM extends SOM to clustering, visualizing, and filtering a large collection of text documents; tasks that are essential in the domains of text-mining and information retrieval. WEBSOM consists of two steps: (i) vector space model (Salton et al, 1975), and (ii) dimensionality reduction. In the first step, WEBSOM sets up a word-document matrix whose elements are the weighed frequencies of a word in each document. The frequencies are weighed by using the inverse document frequency (i.e. IDF), so that rare words could get a higher weight than frequent words. The justification for this scoring mechanism being that words that appear rarely enjoy more discriminatory power than words that appear frequently do (Manning et al, 2009, p.119). If a body of documents is classified into known topics, then a word can be weighed by using entropy (Shannon, 1948). For example, let the entropy of word "$\omega$" be: $H_{(\omega)} = -\sum_g \frac{N_g(\omega)}{N_g} log_2 \frac{N_g(\omega)}{N_g}$, and the total entropy of a body of documents be: $H_{max} = log_2 N_g$, where $N_g(\omega)$ denotes the number of times word "$\omega$" occurs in topic $g$, and $N_g$ represents the number of topics. Then, $W_{(\omega)} = H_{max} - H_{(\omega)}$ becomes the weight for word "$\omega$".

However, the vector space model increases dimensionality, because each word has to be represented by a dimension in the document vector. To reduce dimensionality, WEBSOM uses random mapping (Kaski, 1998) rather than the more established technique of latent semantic indexing (i.e. LSI) (Deerwester et al, 1990). The reason is the computational complexity of the former, $O(Nl) + O(n)$, is much lower than that of the latter, $O(Nld)$; where $N$ is the number of documents, $l$ is the average number of different words in a document, $n$ represents the original dimensionality, and $d$ denotes the dimensionality that results from LSI performing singular value decomposition. Further, experimental results have suggested that the classification accuracy of random mapping is very close to that of LSI (Kohonen et al, 2000).

WEBSOM has been applied[2] in clustering and visualizing collections of text documents, such as: articles from Encyclopaedia Britannica (Lagus et al, 2004), a collection of 7 million patent abstracts that are available in electronic form and written in the English language (Kohonen et al, 2000), a number of scientific abstracts[3] (Lagus, 1997), and articles from Usenet newsgroups (Honkela et al, 1996a,b).

[2] Some of these applications are demonstrated at: `http://websom.hut.fi/websom/`.

[3] The WEBSOM of scientific abstracts is available at: `http://www.cis.hut.fi/wsom97/abstractmap/`.

### 5.1.2 SOM for Symbol Strings

Unlike numerical data, symbol strings and other non-vectorial data lack intrinsic quantitative information for a distance metric (e.g. Euclidean) to be used as a similarity metric. To extend SOM to symbol strings and other non-vectorial data, Kohonen (1996), and Kohonen and Somervuo (1998) put forward the following principles: (i) define learning as a succession of conditional averages over subsets of strings, (ii) use batch-SOM, Eq. 7, (iii) define a valid similarity measure over strings (e.g. Levenshtein distance), and (iv) averages over such data are computed as generalized means or medians (Kohonen, 1985). Based on these principles, SOM for symbol strings has been applied in constructing a pronunciation dictionary for speech recognition (Kohonen and Somervuo, 1997), clustering protein sequences (Kohonen and Somervuo, 2002), and identifying novel and interesting clusters from a selection of human endogenous retroviral sequences (Oja et al, 2005).

### 5.1.3 SOM for Categorical Data

To extend SOM to categorical data, Hsu (2006) proposed the Generalized SOM. In brief, a domain expert first describes input data in terms of a concept hierarchy, and then extends it to a distance hierarchy by assigning a weight to each link that exists on the concept hierarchy. For example, a data point $X$ is represented on a distance hierarchy by the tuple $(N_x, d_x)$, where $N_x$ denotes the leaf node corresponding to $X$, and $d_x$ stands for the distance between the root node and $N_x$. The distance between two points, $X$ and $Y$, can then be defined as:

$$\mid X - Y \mid = d_x + d_y - 2d_{LCP(X,Y)} \quad , \tag{12}$$

where $d_{LCP(X,Y)}$ is the distance between the root node and the least common point of $X$ and $Y$.

### 5.1.4 SOM for Hierarchical Data

The graph-theoretical approach (Argyrou, 2009) is founded on graph theory and is decoupled from SOM. It functions as a data pre-processing step that transforms hierarchical data into a numerical representation, and thereby renders them suitable for SOM-based processing. In essence, it operates in two steps. First, it encodes hierarchical data in the form of a directed acyclic graph (i.e. DAG), whereby the root vertex represents the complete set of data, and all other vertices are ordered in such a way that each vertex is a subset of its parent vertex. In doing so, the graph-theoretical approach preserves the semantical relationships, (i.e. $child \prec parent$ relationship), that exist between hierarchical data. Second, it uses Dijkstra's algorithm (Dijkstra, 1959) in order to calculate all pairwise distances between vertices. The calculation yields a distance matrix that satisfies the conditions of a metric space, specified in Section 2.2, and thus it can form the input dataset to SOM.

## 5.2 SOM for Temporal Sequence Processing

SOM can be applied only to static data, because it ignores the temporal ordering of input data. To elaborate, we draw on the updating rules, Eq. 6 and Eq. 7, that represent the output of a neuron at iteration $t$; iteration $t$ acts as a surrogate for discrete time.

At each iteration, the updating rules update codevector $m_k$ towards input vector $x_i$ by considering the value of $m_k$ at the previous iteration, but ignoring any changes to the value of $x_i$. As a result, SOM can not capture any temporal context that may exist between consecutive input vectors.

### 5.2.1 Recurrent SOM

Recurrent SOM (i.e. RSOM) incorporates leaky integrators to maintain the temporal context of input vectors, and thus it extends SOM to temporal sequence processing (Koskela et al, 1998); it also constitutes a significant improvement to Temporal Kohonen Map (Chappell and Taylor, 1993) in terms of learning and convergence (Varsta et al, 2001).

RSOM modifies SOM's updating rule, Eq. 6, and models the leaky integrators as follows:

$$m_k(t+1) = m_k(t) + a(t)h_{ck}(t)\psi_k(t,\beta) \quad , \tag{13}$$

$$\psi_k(t,\beta) = (1-\beta)\psi_k(t-1,\beta) + \beta\left(x_i(t) - m_k(t)\right) \quad , \tag{14}$$

where $\psi_k(t,\beta)$ is the leaked difference vector of neuron $k$ at iteration $t$, and $0 < \beta \leqslant 1$ is the leaky coefficient that determines how quickly memory decays. A large $\beta$ entails fast memory decay, i.e. short memory, whereas a small $\beta$ a slower memory loss, i.e. long memory. The leaked difference vector forms the feedback to RSOM; and because the feedback is a vector rather than a scalar, it allows the updating rule, Eq. 13, to capture information about changes in both the magnitude and direction of input vectors.

### 5.2.2 Recursive SOM

Recursive SOM (i.e. RecSOM) uses the output of the entire SOM as its feedback to the next iteration (Voegtlin, 2002). RecSOM defines neuron $k$ by means of two codevectors: (i) a feed-forward codevector, $m_k \in \mathbb{R}^{\mathsf{d}}$, which is the same as the codevector defined by SOM, and (ii) a recurrent codevector, $w_k(t) \in \mathbb{R}^{|\mathsf{N}|}$ that represents the output of the entire SOM at iteration $t$, where $\mid N \mid$ denotes the number of iterations. The distance calculation becomes:

$$d_k(t) = \alpha \parallel x_i(t) - m_k(t) \parallel^2 + \beta \parallel y(t-1) - w_k(t) \parallel^2 \quad , \tag{15}$$

where $\alpha$,$\beta > 0$, $y(t) = [\exp(-d_1(t), \exp(-d_2(t), \ldots, \exp(-d_K(t)]$, and $k = 1, 2, \ldots, K$ represents the number of neurons. It follows form Eq.15 that the best-matching unit is given by $c = \underset{k}{\operatorname{argmin}}\{d_k(t)\}$.

The updating rule for the feed-forward codevector is the same as SOM's updating rule described in Eq. 6, that is:

$$m_k(t+1) = m_k(t) + a(t)h_{ck}(t)\left[x_i(t) - m_k(t)\right] \quad , \tag{16}$$

and the updating rule for the recurrent codevector is:

$$w_k(t+1) = w_k(t) + a(t)h_{ck}(t)\left[y(t-1) - w_k(t)\right] \quad . \tag{17}$$

### 5.3 SOM for Cluster Analysis

Cluster analysis aims at identifying subsets or clusters in data for which no information about their class-labels is known in advance. Cluster analysis groups data into subsets so that a distance metric (e.g. Euclidean) within a subset is minimized, and that between one subset and another is maximized. This process ensures that the intra-cluster similarity is maximized, and the inter-cluster similarity is minimized, as the distance between two data points is inversely proportional to their similarity.

While k-means algorithm clusters data directly, SOM performs a two-step clustering: first, it regresses a set of codevectors to input data in a non-parametric, and non-linear manner; second, the codevectors can be clustered by using either a partitive or an agglomerative algorithm. An input vector belongs to the same cluster as its nearest codevector does.

SOM enjoys a lower computational complexity than the algorithms that cluster data directly (Vesanto and Alhoniemi, 2000). In particular, the computational complexity of k-means algorithm is proportionate to $\sum_{C=2}^{Cmax} NC$, where $Cmax$ denotes the maximum number of clusters, $N$ stands for the number of input vectors, and $C$ represents the number of clusters. In contrast, when codevectors are used as as intermediary step, the computational complexity becomes proportionate to $NM + \sum_C MC$, where $M$ represents the number of codevectors. Further, an agglomerative algorithm starts with either $N$ or $M$ clusters depending on whether it is applied directly on input data or on codevectors. The latter is much less computationally expensive than the former, because the number of codevectors is usually chosen to be approximately equal to the square root of the number of input vectors, $M \approx \sqrt{n}$.

SOM has an additional benefit in that it is tolerant to outliers and noise that may be present in input data; the reason is SOM estimates codevectors as weighed averages of input data, as shown in Eq. 7. In addition, SOM is robust to missing values, as it performs the distance calculations, $\| x_i - m_k \|$, by excluding them. This approach yields a valid solution, because the same variables are excluded at each distance calculation (Vesanto et al, 2000, p.7)

For SOM to be able to identify clusters that may exist in input data, the probability density function (i.e. pdf) of codevectors must approximate that of input data; in this case, small and large distances between codevectors indicate dense and sparse areas in the input data, respectively.

For vector quantization algorithms (e.g. k-means), the pdf of codevectors (i.e. $p(m_k)$) approximates asymptotically that of the input data, (i.e. $p(x)$), thus $p(m_k) \propto p(x)^{\frac{d}{d+r}}$, where $d$ denotes the dimensionality of the input space, and $r$ is the distance norm. Such an approximation has been derived only for the one-dimensional SOM (Ritter, 1991):

$$p(m_k) \propto p(x)^{\frac{2}{3} - \frac{1}{3\sigma^2 + 3(\sigma+1)^2}} \quad . \tag{18}$$

The approximation, Eq. 18, is valid only if the neighbourhood width (i.e. $\sigma$) is very large and the number of codevectors tends to infinity. Nonetheless, experimental results have suggested that $p(m_k)$ approximates $p(x)$ or some monotone function of $p(x)$ (Kaski and Kohonen, 1996), and (Kohonen, 1999).

### 5.3.1 Unified-Distance Matrix

The Unified-distance matrix (i.e. U-matrix) enables SOM to identify and visualize clusters that may be present in the input data (Ultsch and Siemon, 1990). U-matrix first calculates the average distance between a neuron's codevector and that of its immediate neighbouring neurons:

$$h(k) = \frac{1}{\mid N_k \mid} \sum_{i \in N_k} d(m_k, m_i) \quad , \tag{19}$$

where $N_k$ stands for the neighbourhood of neuron $k$, and $\mid N_k \mid$ represents the number of neurons in that neighbourhood. It then superimposes distance $h(k)$ as a height on the SOM grid, and thereby transforms the latter into a three-dimensional landscape of a multi-dimensional input space. To represent the additional information (i.e. distances), U-matrix augments the SOM grid by inserting an additional neuron between each pair of neurons. The distances are depicted on the SOM grid as shades of grey; dark and light areas indicate large and small distances respectively, and denote cluster boundaries and clusters in that order.

U-matrix may not be sufficient to delineate clusters that are either overlapping or not well-separated, because it does not take into account the density of input data. To overcome this limitation, U*-matrix combines both distances and densities in order to improve the visualization and clustering of input data (Ultsch, 2003a,c). It uses Pareto Density Estimation (i.e. PDE) to estimate the local density of input data as follows (Ultsch, 2003b):

$$p(k) = \{x_i \in \mathbb{R}^{\mathrm{d}} \,|\, d(x_i, m_k) < r\} \quad , \tag{20}$$

where $p(k)$ denotes the density of input data in the vicinity of codevector $m_k$, and $r$ denotes the Pareto radius of a hypersphere. If $r$ is kept constant, then the number of data points that are included in a hypersphere is proportionate to the underlying density of the data.

### 5.3.2 U-Matrix Refined

To refine the resolution of U-matrix, Kaski et al (2003) proposed the following modification:

$$G_{ki} = \parallel (m_k - m_i) - (c_k - c_i) \parallel \quad . \tag{21}$$

The first term calculates the distance between codevectors $m_k$ and $m_i$; the second term is estimated from the input data, where $c_k = \frac{1}{|N_k|} \sum_{x \in N_k} x$ denotes the centroid of the input data that belong in the neighbourhood of neuron $k$. The magnitude of $G_{ki}$ is inversely proportional to the density of the corresponding area, the lower the density the larger the magnitude of $G_{ki}$ would be, and vice versa. In a manner similar to U-matrix, the values of $G_{ki}$ can be converted to colour by using index-colour coding, and subsequently be depicted on the SOM grid in order to delineate inter-cluster boundaries. Experimental results have suggested that this visualization method can identify important and meaningful clusters the U-matrix can not identify (Kaski et al, 2003).

### 5.3.3 SOM Interpretation

In order to evaluate how well a variable can explain the formation of cluster borders, Kaski et al (1998) suggested the following metric:

$$\Phi_j = \frac{\| (m_{ij} - m_{kj}) \|}{\| (m_i - m_k) \|} \quad , \tag{22}$$

where the denominator represents the distance between codevectors $m_i$ and $m_k$, $m_{ij}$ and $m_{kj}$ denote the $j^{th}$ variable of $m_i$ and $m_k$, respectively. A large $\Phi_j$ value means that variable $j$ explains well the formation of the cluster border, and data points that belong on either side of the border differ predominantly in the value of variable $j$.

Two further measures assess the relative importance a variable possesses within, and between clusters (Siponen et al, 2001).

$$S_{ij} = \frac{mean_{ij} - min_j}{max_j - min_j} \quad . \tag{23}$$

The term $S_{ij}$ measures the weight variable $j$ has within cluster $i$ relative to that variable's range, $mean_{ij}$ is the mean value of variable $j$ in cluster $i$, $min_j$ and $max_j$ denote the minimum and maximum values of variable $j$, respectively.

The quantity $Q_{ij}$ measures the weight of variable $j$ in cluster $i$ with respect to the values of that variable in clusters other than $i$:

$$Q_{ij} = \frac{S_{ij}}{\frac{1}{C-1}\sum_{k \neq i} S_{kj}} \quad , \tag{24}$$

where $C$ represents the number of clusters.

## 5.4 SOM for Visualizing High-Dimensional Data

As we elaborated in Section 3, SOM preserves the topology or neighbourhood relations between input data as faithfully as possible. However, SOM does not perform a point-to-point mapping, and hence it can not reproduce the pairwise distances between input data. Thus, the distances between neurons on the SOM grid are not proportionate to their corresponding distances in the input space.

ViSOM (Yin, 2002) modifies SOM's updating rule, described in Eq. 6, so that SOM could preserve not only the topology but also the pairwise distances between the input data. It decomposes the term $F_{ik} = \| x_i(t) - m_k(t) \|$ into two parts: (i) an expansion force, $F_{ic} = \| x_i(t) - m_c(t) \|$, where $m_c(t)$ stands for the BMU codevector of input vector $x_i(t)$, and (ii) a lateral force $F_{ck} = \| m_c(t) - m_k(t) \|$. ViSOM's updating rule is defined as:

$$m_k(t+1) = m_k(t) + a(t)h_{ck}(t)[\| x_i(t) - m_c(t) \| + \| m_c(t) - m_k(t) \| (\frac{d_{ck}}{\lambda \Delta_{ck}} - 1)] \forall k \in N_c \quad , \tag{25}$$

where $N_c$ is the neighbourhood of neuron $c$, $k$ denotes the neurons that belong in $N_c$, $d_{ck}$ represents the distance between codevectors $m_c$ and $m_k$, $\Delta_{ck}$ represents the distance between their location vectors, $p_c$ and $p_k$, on the SOM grid, and $\lambda$ is a resolution parameter. The aim of Eq. 25 is to adjust inter-neuron distances on the SOM grid, (i.e. $\| p_c - p_k \|$), so that they could be proportionate to their corresponding distances in the input space, (i.e. $\| m_c - m_k \|$).

## 6 Financial Applications of SOM

To demonstrate how a researcher can apply SOM in order to understand complex data sets, Kaski and Kohonen (1996) applied SOM to 39 statistical indicators that described aspects of the welfare of a number of countries. Based on the 39 statistical indicators, the resulting SOM grid revealed a clustering of countries as well as maintained the similarity between one country and another. For example, OECD and most African countries were clustered on opposite corners of the map indicating extreme welfare and poverty, respectively.

SOM has been applied in predicting the event of bankruptcy as well as in identifying those financial characteristics that are positively correlated with bankruptcy. In order to gain insight into the Spanish banking crisis of 1977-1985, Martn-del-Bro and Serrano-Cinca (1993) selected a sample of 37 solvent banks and 29 bankrupt banks. For each bank, they calculated the following nine ratios: (i) Current Assets/Total Assets, (ii) (Current Assets - Cash and Banks)/Total Assets, (iii) Current Assets/Loans, (iv) Reserves/Loans, (v) Net Income/Total Assets, (vi) Net Income/Total Equity Capital, (vii) Net Income/Loans, (viii) Cost of Sales/Sales, and (ix) Cash-Flow/Loans. Based on these ratios, the study developed a SOM that was able to cluster solvent banks separately from their bankrupt counterparts, and identify how well each ratio could discriminate between the two sets of banks. As expected, small profitability and large debts were positively correlated with the event of bankruptcy. A closer examination revealed that SOM grouped the seven biggest banks as a sub-cluster of the solvent banks, although no information about the sizes of the banks was available.

In addition, Serrano-Cinca (1996) selected 64 solvent, and 65 bankrupt companies from the Moody's Industrial Manual as well as five ratios that described the financial performance of the companies from 1975 to 1985. The ratios were: (i) Working Capital/Total Assets, (ii) Retained Earnings/Total Assets, (iii) Earnings before Interest and Tax/Total Assets, (iv) Market value of Equity, and (v) Sales/Total Assets. SOM clustered the two sets of companies into separate clusters, and also revealed that solvent and bankrupt companies were characterised by high and low levels of earnings, respectively.

Further, to predict the event of bankruptcy, Kiviluoto (1998) applied SOM in a semi-supervised manner in that the input vector, and by extension the codevector, consisted of two parts. The first part contained four financial indicators: (i) Operating Margin, (ii) Net Income before Depreciation and Extraordinary items, (iii) the same as (ii) for the previous year, and (iv) Equity ratio. The second part was a binary indicator, $\{1,0\}$, denoting bankrupt and solvent companies, respectively. The first part was used only for finding the BMU, whereas the whole codevector got updated. The results suggested that companies having low profitability were more likely to go bankrupt.

To investigate the potential of SOM in performing financial benchmarking, Back et al (1996, 1997) applied SOM to a set of financial ratios that described the performance of a number of international companies operating in the pulp and paper industry from 1985 to 1989. The results suggested that SOM could cluster companies according to their financial similarities, and identify regularities and patterns that were present in the data. This line of research was applied by Karlsson et al (2001) in order to analyse the quarterly financial performance of a number of international companies that operated in the telecommunication industry.

A similar methodology was adopted by Eklund et al (2003). They first developed two sets of SOM: the first was based on financial ratios of international companies operating in the pulp and paper industry, and the second was based on the financial ratios having been averaged on a country by country basis. They then compared the two sets in order to identify which financial characteristics were specific to a country. For example, the results revealed the propensity of Finnish companies to finance their activities via debt as well as the economic downturn the Japanese companies experienced in 1997 as a result of the Asian financial crisis. Lansiluoto et al (2004) extended the foregoing body of literature by using SOM to perform financial benchmarking at both company and industry levels; they did so in order to find out how changes at the industry level (e.g. labour and material costs) might affect the financial performance of companies.

SOM was used by (Baghai-Wadj et al, 2005) to derive a taxonomy of hedge funds based on their monthly returns in order to address the following three issues: (i) whether hedge funds change their self-declared styles over time, (ii) whether self-declared styles are useful or misleading, and (iii) which types of hedge funds are prone to misclassify their own styles. The study selected the monthly returns of 2,442 hedge funds from the Centre for International Securities and Derivatives Markets (i.e. CISDM). The monthly returns covered the period from April 1995 to April 2004, and the sample included both active and inactive hedge funds. SOM was able to group hedge funds into nine clusters that were labelled as follows: (i) convertible arbitrage and fixed income, (ii) emerging markets, (iii) futures, (iv) merger arbitrage and distressed securities, (v) sector financial, (vi) sector health care, (vii) sector technology, (viii) short-selling, and (ix) other hedge funds. The study found that 23% of the examined hedge funds changed their self-declared style over the period under investigation, and that hedge funds having high consistency in their self-declared styles were less prone to change their styles over time.

Further financial applications of SOM include: selecting shares suitable for portfolio management (Deboeck and Ultsch, 2000), and (Khan et al, 2009); assessing the creditworthiness of loan applicants (Tan et al, 2002), and (Huysmans et al, 2006); and performing market segmentation as part of customer relationship management (Hung and Tsai, 2008).

## 7 Case Study: Clustering Accounting Databases

The case study illustrates how SOM can be applied in clustering accounting databases, and discusses potential applications such clustering may have in the discipline of accounting. The motivation behind the case study is twofold. First, existing literature has paid insufficient attention to the clustering of accounting databases as an internal control procedure; and second, the importance and relevance of this issue are underlined by a number of statutory and professional pronouncements that have proliferated in the wake of several cases of financial malfeasance (e.g. Enron, WorldCom).

### 7.1 Data Description

The data were provided by an international shipping company as a text-dump of its accounting database, and consisted of 16,300 journal entries covering fiscal year 2007.

**Table 1.** Journal Entry

| Account Number | Account Description | Date | Debit-Credit | US$ Amount | Details | Account Class |
|---|---|---|---|---|---|---|
| 60000 | Salaries | 27/Aug/2007 | Debit | 56,070 | Payment of salaries | Expenses |
| 30000 | Bank | 27/Aug/2007 | Credit | 56,070 | Payment of salaries | Cash & Cash Equivalents |

Journal entries implement the system of double-entry bookkeeping by recording accounting transactions in a "Debit" and "Credit" format; it entails that an accounting transaction is recorded twice: as a "Debit" and as an equivalent "Credit", hence the name "double-entry". This system of recording accounting transactions serves to preserve the accounting equality: $Assets = Liabilities + Equity$. To be considered as an accounting transaction and thus be recorded, an economic event must cause either an increase or a decrease in a company's assets, liabilities, or equity.

For example, Table 1 depicts the aforesaid company's journal entry that consists of two accounting transactions, and records the payment of salaries for August 2007. Each accounting transaction is described by the following seven variables: (i) Account Number (hierarchical), (ii) Account Description (text), (iii) Date (date), (iv) Debit-Credit (binary), (v) US$ Amount (numerical), (vi) Details (text), and (vii) Account Class (hierarchical). The types of variables are shown in parentheses.

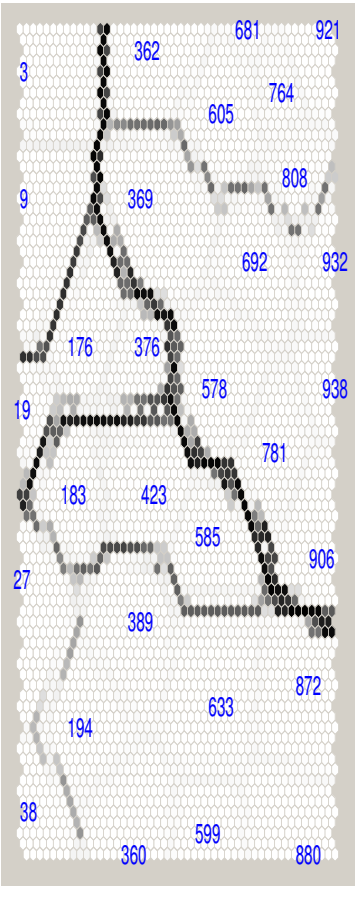
To carry out the case study, we select three variables: (i) Debit-Credit, (ii) US$ Amount, and (iii) Account Class. We incorporate "Debit-Credit" into "US$ Amount" by expressing the latter as positive and negative amounts denoting "Debit" and "Credit" balances, respectively.

### 7.2 Data Pre-processing

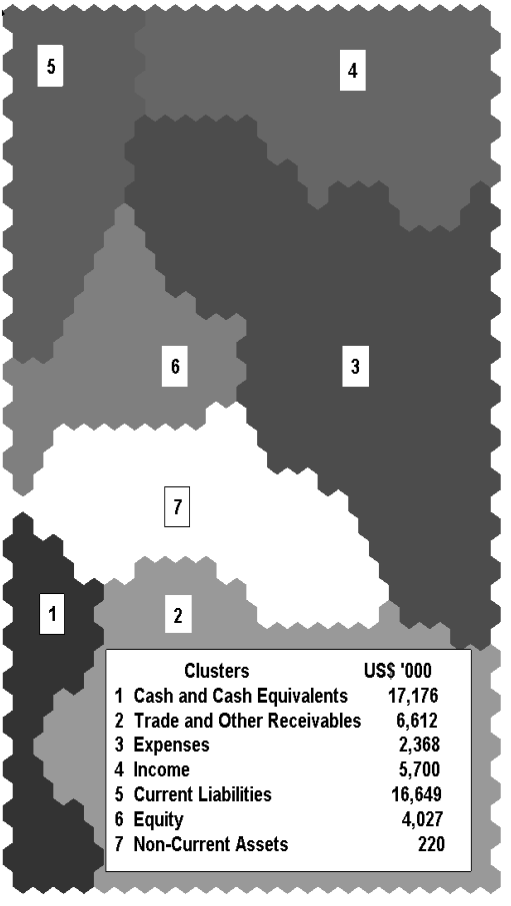
As we discussed in Section 2.2, we must define a valid similarity metric over "Account Class" for SOM to yield meaningful results; "Account Class" is a hierarchical variable and as such it lacks intrinsic quantitative information, and hence SOM can not calculate the Euclidean distance. To this end, we opt for the graph-theoretical approach, described in Section 5.1.4, in order to transform the values of "Account Class" into a numerical representation, and thus render them amenable for SOM-based processing.

To elaborate, we observe that the values of "Account Class" follow a hierarchy, the IASB XBRL Taxonomy (IASCF, 2009); it describes accounting concepts and their semantical relationships as $child \prec parent$ links according to the International Financial Reporting Standards (IASB, 2009). Based on the aforementioned taxonomy, we first encode "Account Class" as a directed acyclic graph (i.e. DAG), whereby the root vertex represents the complete set of "Account Class", and all other vertices are ordered in such a way that each vertex represents a subset of its parent vertex. Consequently, the DAG can preserve the $child \prec parent$ relationships specified in the taxonomy. For example, "Salaries"$\prec$ "Expenses" $\prec$ "Income Statement" forms a path on the DAG, and serves to induce the corresponding $child \prec parent$ relationships.

Second, we calculate all pairwise distances between the vertices by using Dijkstra's algorithm (Dijkstra, 1959). This operation yields distance matrix $X = (x_{ij})_{nd}$, where $d = 30$ denotes the dimensionality, and $n = 36,510$ represents the number of accounting transactions. The distance matrix can form the input dataset to SOM, as it satisfies the conditions




**(a)** SOM grid: the numbers denote those neurons that are BMU, and U-matrix identifies seven clusters. The dark areas represent the borders of the clusters.




**(b)** SOM grid: it labels the seven clusters, and also conveys their respective total US$ amounts for fiscal year 2007.

**Fig. 1.** The SOM grids

of a metric space specified in Section 2.2. There are more accounting transactions than journal entries, because a journal entry can be composed of any number of accounting transactions provided that the "Debit" and "Credit" balances are always equal.

### 7.3 Experiments

To illustrate how SOM can be applied in clustering accounting databases, the paper adopts a five-step approach. First, it uses bootstrap to draw two hundred random samples with replacement from the empirical distribution of the input dataset, $X = (x_{ij})_{nd}$.

Second, for each bootstrapped sample, it uses SOM-Toolbox for Matlab[4] (Vesanto et al, 2000) to train a SOM in batch mode, Eq. 7, with hexagonal topology, and Gaussian neighbourhood, Eq. 5. Third, to identify clusters on the SOM grid, it uses U-matrix, described in Section 5.3.1. Fourth, to evaluate the performance of SOM, it calculates the quantization and topographic errors, and to assess the internal validity of clustering, it calculates the Davies-Bouldin index; these metrics were explained in Section 4. Finally, the paper estimates the two-sided 95% confidence intervals of the mean and standard deviation of the foregoing metrics.

### 7.4 Results Presentation and Discussion

The mean and standard deviation of the topographic error exhibit small values and narrow 95% confidence intervals: $CI_{0.95} = 0.3336 - 0.3984$, and $CI_{0.95} = 0.1478 - 0.1880$, respectively; and the corresponding values for the quantization error are zero to the $4^{th}$ decimal point. These results suggest that the SOM grid can map the input data accurately as well as preserve their topology faithfully.

The results of the case study are depicted in the form of two SOM grids in Fig. 1. A visual inspection of Fig. 1a reveals how U-matrix can identify seven clusters. As we discussed in Section 5.3.1, dark and light areas indicate long and short distances, respectively; the former delineates cluster borders, whereas the latter represents clusters. The internal validity of clustering is evaluated by statistical analysis; both the mean and standard deviation of Davies-Bouldin index have small values and narrow 95% confidence intervals: $CI_{0.95} = 1.4905 - 1.5988$, and $CI_{0.95} = 0.2445 - 0.3563$, in that order. The statistical results imply that the seven clusters thus identified are highly-compact, and their centroids are well-separated.

More importantly, the seven clusters can be interpreted within an accounting context, as shown in Fig. 1b. Specifically, clusters 3 and 4 represent expense and income items respectively, and the Income Statement collectively. Further, clusters 7, 2, and 1 correspond to "Non-Current Assets", "Trade and Other Receivables", and "Cash and Cash Equivalents", in that order. These three clusters make up the "Assets" side of the Balance Sheet Statement. In addition, clusters 6 and 5 stand for "Equity" and "Current Liabilities" respectively, and their total forms the "Equity and Liabilities" side of the Balance Sheet Statement. Finally, for each cluster, Fig. 1b conveys the total US$ amount in thousands for fiscal year 2007. This piece of information may serve useful purposes, as it provides a user with aggregated US$ amounts for each category of the financial statements. Indeed, the SOM grid, Fig. 1b, preserves the accounting equality: $Assets(220+6,612+17,176) = Liabilities(16,649) + Equity(4,027+3,332)$; the amount of US$ 3,332 (i.e. 5,700 -2,368) represents the profit for the fiscal year 2007.

## References

Argyrou, A.: Clustering hierarchical data using self-organizing map: A graph-theoretical approach. In: Príncipe, J., Miikkulainen, R. (eds.) WSOM 2009. LNCS, vol. 5629, pp. 19–27. Springer, Heidelberg (2009)

[4] SOM-Toolbox for Matlab and its documentation are available at: `http://www.cis.hut.fi/somtoolbox`.

Back, B., Sere, K., Vanharanta, H.: Data mining accounting numbers using self-organizing maps. In: Alander, J., Honkela, T., Jakobsson, M. (eds.) Genes, Nets and Symbols (STeP 1996). Finnish Artificial Intelligence Society, pp. 35–47. University of Vaasa, Vaasa (1996)

Back, B., Sere, K., Vanharanta, H.: Analyzing financial performance with self-organizing maps. In: Proceedings of the First International Workshop on Self-Organizing Maps (WSOM 1997), Espoo, Finland, pp. 356–361 (1997)

Baghai-Wadj, R., El-Berry, R., Klocker, S., Schwaiger, M.: The Consistency of Self-Declared Hedge Fund Styles - A Return-Based Analysis with Self-Organizing Maps. Central Bank of Austria: Financial Stability Report (9), 64–76 (2005)

Bishop, C.M.: Neural networks for pattern recognition. Oxford University Press, Oxford (1995)

Chappell, G.J., Taylor, J.G.: The temporal kohonen map. Neural Networks 6(3), 441–445 (1993)

Davies, D., Bouldin, D.: A cluster separation measure. IEEE Transactions on Pattern Analysis and Machine Intelligence 1(2), 224–227 (1979)

Deboeck, G., Ultsch, A.: Picking stocks with emergent Self-Organizing value maps. Neural Networks World 10(1-2), 203–216 (2000)

Deerwester, S., Dumais, S., Furnas, G., Thomas, Harshman, R.: Indexing by latent semantic analysis. Journal of the American Society for Information Science 41, 391–407 (1990)

Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische Mathematik 1, 269–271 (1959)

Eklund, T., Back, B., Vanharanta, H., Visa, A.: Financial benchmarking using self-organizing maps studying the international pulp and paper industry. In: Data Mining: Opportunities and Challenges, pp. 323–349. IGI Publishing (2003)

Erwin, E., Obermayer, K., Schulten, K.: Self-Organizing maps: Ordering, convergence properties and energy functions. Biological Cybernetics 67, 47–55 (1992)

Graepel, T., Burger, M., Obermayer, K.: Phase transitions in stochastic self-organizing maps. Physical Review E 56(4), 3876–3890 (1997)

Haykin, S.: Neural Networks. A Comprehensive Foundation, 2nd edn. Prentice Hall International, Upper Saddle River (1999)

Honkela, T., Kaski, S., Lagus, K., Kohonen, T.: Exploration of full-text databases with self-organizing maps. In: Proceedings of the International Conference on Neural Networks (ICNN 1996), vol. I, pp. 56–61. IEEE Service Center, Piscataway (1996a)

Honkela, T., Kaski, S., Lagus, K., Kohonen, T.: Newsgroup exploration with WEBSOM method and browsing interface. Tech. Rep. A32. Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland (1996b)

Hsu, C.: Generalizing self-organizing map for categorical data. IEEE Transactions on Neural Networks 17(2), 294–304 (2006)

Hung, C., Tsai, C.: Market segmentation based on hierarchical self-organizing map for markets of multimedia on demand. Expert Systems with Applications 34(1), 780–787 (2008)

Huysmans, J., Baesens, B., Vanthienen, J., van Gestel, T.: Failure prediction with self organizing maps. Expert Systems with Applications 30(3), 479–487 (2006)

IASB, International Financial Reporting Standards (IFRS), International Accounting Standards Committee Foundation (IASCF), London, United Kingdom (2009)

IASCF, IFRS Taxonomy Guide 2009 (XBRL). International Accounting Standards Committee Foundation (IASCF), London, United Kingdom (2009)

Jungnickel, D.: Graphs, Networks and Algorithms, English edn. In: Algorithms and Computation in Mathematics, vol. 5. Springer, Berlin (2002)

Karlsson, J., Back, B., Vanharanta, H., Visa, A.: Analysing financial performance with quarterly data using Self-Organising Maps. TUCS Technical Report No 430, Turku Centre for Computer Science, Turku, Finland (2001), `http://tucs.fi/publications/attachment.php?fname=TR430.pdf`

Kaski, S.: Dimensionality reduction by random mapping: Fast similarity computation for clustering. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN 1998), vol. 1, pp. 413–418. IEEE Service Center, Piscataway (1998)

Kaski, S., Kohonen, T.: Exploratory data analysis by the self-organizing map: Structures of welfare and poverty in the world. In: Refenes, A.N., Abu-Mostafa, Y., Moody, J., Weigend, A. (eds.) Neural Networks in Financial Engineering. Proceedings of the Third International Conference on Neural Networks in the Capital Markets, London, England, October 11-13, pp. 498–507. World Scientific, Singapore (1996)

Kaski, S., Nikkilä, J., Kohonen, T.: Methods for interpreting a Self-Organized map in data analysis. In: Proceedings of ESANN 1998, 6th European Symposium on Artificial Neural Networks, D-Facto, Brussels, Belgium, pp. 185–190 (1998)

Kaski, S., Nikkilä, J., Kohonen, T.: Methods for exploratory cluster analysis. In: Intelligent Exploration of The Web. Studies In Fuzziness And Soft Computing, pp. 136–151. Physica-Verlag GmbH, Heidelberg (2003)

Khan, A.U., Bandopadhyaya, T.K., Sharma, S.: Classification of stocks using self organizing map. International Journal of Soft Computing Applications (4), 19–24 (2009)

Kiviluoto, K.: Topology preservation in self-organizing maps. In: Proceeding of the International Conference on Neural Networks (ICNN 1996), vol. 1, pp. 294–299. IEEE Service Center, Piscataway (1996)

Kiviluoto, K.: Predicting bankruptcies with the self-organizing map. Neurocomputing 21(1-3), 191–201 (1998)

Kohonen, T.: Self-organized formation of topologically correct feature maps. Biological Cybernetics 43(1), 59–69 (1982)

Kohonen, T.: Median strings. Pattern Recognition Letters 3(5), 309–313 (1985)

Kohonen, T.: Self-Organizing Maps of symbol strings. Tech. Rep. A42. Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland (1996)

Kohonen, T.: Self-Organizing Maps, 2nd edn. Springer Series in Information Sciences, vol. 30. Springer, Heidelberg (1997)

Kohonen, T.: The self-organizing map. Neurocomputing 21(1-3), 1–6 (1998)

Kohonen, T.: Comparison of SOM point densities based on different criteria. Neural Computation 11(8), 2081–2095 (1999)

Kohonen, T.: Data Management by Self-Organizing Maps. In: Zurada, J.M., Yen, G.G., Wang, J. (eds.) Computational Intelligence: Research Frontiers. LNCS, vol. 5050, pp. 309–332. Springer, Heidelberg (2008)

Kohonen, T., Somervuo, P.: Self-organizing maps of symbol strings with application to speech recognition. In: Proceedings of the First International Workshop on Self-Organizing Maps (WSOM 1997), Espoo, Finland, pp. 2–7 (1997)

Kohonen, T., Somervuo, P.: Self-organizing maps of symbol strings. Neurocomputing 21(1-3), 19–30 (1998)

Kohonen, T., Somervuo, P.: How to make large self-organizing maps for nonvectorial data. Neural Networks 15(8-9), 945–952 (2002)

Kohonen, T., Kaski, S., Lagus, K., Salojarvi, J., Honkela, J., Paatero, V., Saarela, A.: Self organization of a massive document collection. IEEE Transactions on Neural Networks 11(3), 574–585 (2000)

Koskela, T., Varsta, M., Heikkonen, J., Kaski, K.: Temporal sequence processing using recurrent SOM. In: Proceedings of the Second International Conference on Knowledge-Based Intelligent Electronic Systems (KES 1998), vol. 1, pp. 290–297 (1998)

Lagus, K.: Map of WSOM 1997 Abstracts Alternative Index. In: Proceedings of the First International Workshop on Self-Organizing Maps (WSOM 1997), Espoo, Finland, pp. 368–372 (1997)

Lagus, K., Kaski, S., Kohonen, T.: Mining massive document collections by the WEBSOM method. Information Sciences 163(1-3), 135–156 (2004)

Lansiluoto, A., Eklund, T., Back, B., Vanharanta, H., Visa, A.: Industry-specific cycles and companies' financial performance comparison using self-organizing maps. Benchmarking: An International Journal 11(3), 267–286 (2004)

Lawrence, R., Almasi, G., Rushmeier, H.: A scalable parallel algorithm for Self-Organizing Maps with applications to sparse data mining problems. Data Mining and Knowledge Discovery 3(2), 171–195 (1999)

Manning, C.D., Raghavan, P., Shutze, H.: An Introduction to Information Retrieval, Online edn. Cambridge University Press, New York (2009), `http://nlp.stanford.edu/IR-book/pdf/irbookprint.pdf`

Martn-del-Bro, B., Serrano-Cinca, C.: Self-organizing neural networks for the analysis and representation of data: Some financial cases. Neural Computing & Applications 1(3), 193–206 (1993)

Oja, M., Sperber, G., Blomberg, J., Kaski, S.: Self-organizing map-based discovery and visualization of human endogenous retroviral sequence groups. International Journal of Neural Systems 15(3), 163–179 (2005)

Ritter, H.: Asymptotic level density for a class of vector quantization processes. IEEE Transactions on Neural Networks 2(1), 173–175 (1991)

Ritter, H., Kohonen, T.: Self-Organizing Semantic Maps. Biological Cybernetics 61(4), 241–254 (1989)

Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Communications of ACM 18(11), 613–620 (1975)

Serrano-Cinca, C.: Self organizing neural networks for financial diagnosis. Decision Support Systems 17(3), 227–238 (1996)

Shannon, C.E.: A mathematical theory of communication. The Bell System Technical Journal 27, 379–423, 623–656 (1948)

Siponen, M., Vesanto, J., Simula, O., Vasara, P.: An approach to automated interpretation of SOM. In: Allinson, N., Yin, H., Allinson, L., Slack, J. (eds.) Proceedings of Workshop on Self-Organizing Map (WSOM 2001), pp. 89–94. Springer, Heidelberg (2001)

Stevens, S.S.: On the theory of scales of measurement. Science 103(2684), 677–680 (1946)

Tan, R., van den Berg, J., van den Bergh, W.: Credit rating classification using Self-Organizing Maps. In: Neural Networks in Business: Techniques and Applications, pp. 140–153. Idea Group Publishing, USA (2002)

Ultsch, A.: Maps for the visualization of high-dimensional data spaces. In: Proceedings Workshop on Self-Organizing Maps (WSOM 2003), Hibikino, Kitakyushu, Japan, pp. 225–230 (2003a)

Ultsch, A.: Pareto density estimation: A density estimation for knowledge discovery. In: Innovations in Classification, Data Science, and Information Systems - Proceedings of 27th Annual Conference of the German Classification Society (GfKL 2003), pp. 91–100. Springer, Heidelberg (2003b)

Ultsch, A.: U*-Matrix: a tool to visualize clusters in high dimensional data. Tech. Rep. 36, Department of Mathematics and Computer Science. University of Marburg, Germany (2003c)

Ultsch, A., Siemon, H.: Kohonen's self organizing feature maps for exploratory data analysis. In: Proceedings International Neural Networks, pp. 305–308. Kluwer Academic Press, Dordrecht (1990)

Varsta, M., Heikkonen, J., Lampinen, J., Millán, J.D.: Temporal kohonen map and the recurrent self-organizing map: analytical and experimental comparison. Neural Processing Letters 13(3), 237–251 (2001)

Vesanto, J., Alhoniemi, E.: Clustering of the self-organizing map. IEEE Transactions on Neural Networks 11(3), 586–600 (2000)

Vesanto, J., Himberg, J., Alhoniemi, E., Parhankangas, J.: SOM Toolbox for Matlab 5. Tech. Rep. A57, SOM Toolbox Team. Helsinki University of Technology, Espoo, Finland (2000)

Vesanto, J., Sulkava, M., Hollmén, J.: On the decomposition of the Self-Organizing Map distortion measure. In: Proceedings of the Workshop on Self-Organizing Maps (WSOM 2003), Hibikino, Kitakyushu, Japan, pp. 11–16 (2003)

Voegtlin, T.: Recursive self-organizing maps. Neural Networks 15(8-9), 979–991 (2002)

Yin, H.: Data visualisation and manifold mapping using the ViSOM. Neural Networks 15(8-9), 1005–1016 (2002)

Yin, H.: The Self-Organizing Maps: Background, theories, extensions and applications. In: Fulcher, J., Jain, L.C. (eds.) Computational Intelligence: A Compendium. Studies in Computational Intelligence (SCI), vol. 115, pp. 715–762. Springer, Heidelberg (2008)

# Chapter 15
# Applying Data Mining Techniques to Assess Steel Plant Operation Conditions

Khan Muhammad Badruddin, Isao Yagi, and Takao Terano

Tokyo Institute of Technology, Japan

**Abstract.** The improvement in the operation of melting the scrap metal in electric arc furnace, to make various types of steel products, requires complex expertise. This work discusses data mining approach to this problem. We flattened the time series data of the whole operation into the form which is suitable for conventional data mining methods. This paper describes the methodology for transformation of the time series data and discusses the possible applicability of different classification methods in this domain.

## 1 Introduction

Electric arc furnace (EAF) is a furnace that uses electric arc to heat charged material. Energy-efficient electric arc furnaces (EAF) can lead to economical steel production that will naturally result in the profits of industry to increase. The steelmaking operation in electric arc furnace is a complex process and lots of variables are involved in this manufacturing. Usage of shaft is one of the innovations intended to preheat the scrap, resulting in saving of energy [1] [2].ECOARC$^{TM}$ (ECOlogically friendly and ECOnomical ARC furnace) is a Japanese-developed modern EAF with specially designed shaft [3] with continuous operation [4]. We used operational data from one of ECOARC$^{TM}$ for our work.

The data logged in historical/operational database is a precious resource that can be used to analyze *heats.* The number of variables stored can vary for different electric arc furnaces [5]. The operational database that we used contains values of numerous parameters collected by different sensors for every second throughout the hundreds of *heats* of ECOARC$^{TM}$ . The word "Heat" will be used in this chapter in different contexts with different meanings and in order to avoid confusion, we feel it appropriate to present brief explanation in this section. Heat is form of energy but when we will use word *heat* in italic typeface, we will mean entire operation of ECOARC$^{TM}$.

As a result of recent advances of steel engineering technology, EAFs have evolved for better performance. With the reduced duration of *heat*, it is becoming considerably more difficult to control EAF. Automation management is the EAF operation control in which human operator is supported and guided by automated systems [6]. We want to apply data mining techniques on the operational data to find the rules that can be used to develop intelligent automated systems. Fig. 1. depicts the goal of our work. The future vision is to develop an intelligent automated management system that will

D.E. Holmes, L.C. Jain (Eds.): Data Mining: Found. & Intell. Paradigms, ISRL 25, pp. 343–361.
springerlink.com 

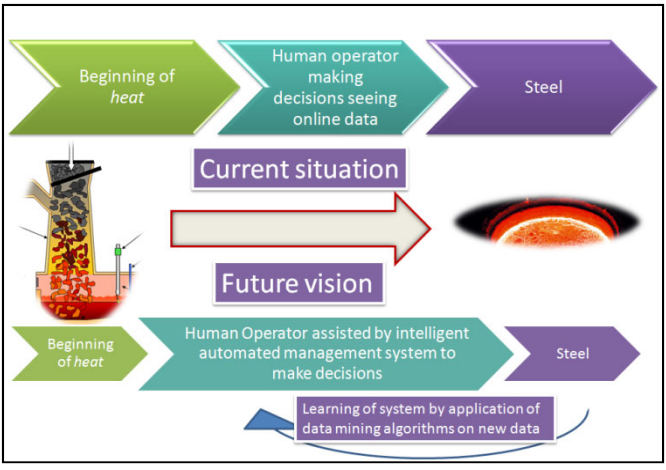



**Fig. 1.** Depiction of goal of the work by outlining current situation and the future vision

be continuously improved by application of data mining algorithms on growing operational data due to newly accumulated experiences. Thus the main theme of this work is *to describe the methodology to discover rules for good energy-efficient operation(s)*. The discovered rules will form rule engine that will be a part of the intelligent automated management system. The system will guide operator to apply different rules seeing the situation through which the *heat* is passing. Fig. 2. illustrates the hypothetical activations of different discovered rules by the operator to improve performance in different stages of hypothetical *heat*. Each column refers to stage of a *heat* and activated rules are illustrated by red color. It is possible that more than one rule can be activated for a particular stage. If operator deems it inappropriate to activate any rule for particular stage, it is also possible as shown for stage 2,7,8 and 11 in Fig. 2.

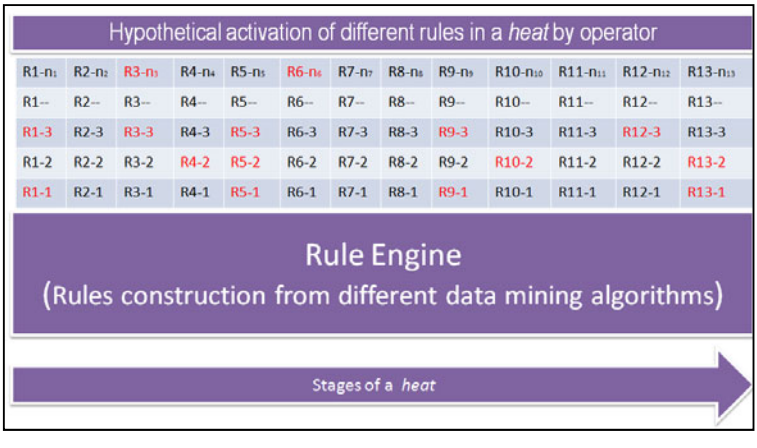



**Fig. 2.** The hypothetical activations of different rules by operator during hypothetical *heat* to improve performance. The rule engine is part of intelligent automated management system.

Since this theme requires understanding of different performance evaluation criteria of an EAF plant, we will mention them, after introducing some basic terminologies and description of part of EAF operation which is relevant to our study in Section 2. In the same section, we will also describe different innovations introduced to EAFs during previous fifty years and their impact on performance of EAFs.

The remaining paper is organized as follows: Section 3 discusses the problem description followed by Section 4 in which data mining process is mentioned. The description of data mining process includes data, its preprocessing, attribute(s) pruning for feature selection, nature of experiments and brief account of different data mining techniques. Section 5 describes the nature of results achievable by application of different data mining techniques. For this purpose, one experiment is used as illustration. In Section 6, the conclusion is made in which description of possible future investigation is also included.

## 2 Brief Description of EAF

Electric arc is used to heat charged material in electric arc furnace (EAF). The scrap is the major component of the charged material that is described by weight in tons. The operating cycle of an EAF is called *tap-to-tap* cycle. The time duration for this whole cycle is called *tap-to-tap time* and is usually denoted by minutes. The different operations performed in this cycle can be placed in either of following two components.

1. Power-on furnace operation time (POT)
2. Power-off furnace operation time

As the name suggests, POT of a *heat* includes the operations which are performed when the electric arcs are on. The following two main operations are carried out at EAF during the POT:

1. Melting of solid charge materials
2. Heating of liquid bath to refine impurities from molten steel(superheating)

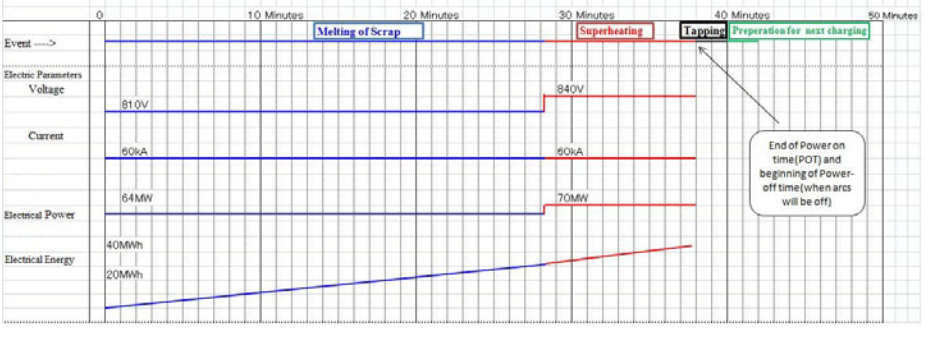



**Fig. 3.** Example Pattern for a *heat* of Electric Arc Furnace

Both of these operations require heat which is provided by electrical and chemical energy. Fig. 3. depicts an example pattern of plot of four electrical parameters (Voltage, Electrical Current, Electrical Power, and Electrical Energy) with respect to passage of time for a *heat* of EAF. The multicolor *Event* line depicts the different stages or events through which the *heat* passes. As can be seen, the most of the time of POT comprises of scrap melting operation (blue line).When the superheating operation (red line) finishes, the electric arcs become off and hence four electrical parameters vanishes. Tapping (black line) and preparation for next *heat* (Green line) are the main components of power-off furnace operation time that is and should be small as compared to POT.

### 2.1 Performance Evaluation Criteria

Performance evaluation criteria of EAF or key performance indicators (KPI) help us to understand the real worth of a furnace. One can compare and judge the performance of different furnaces from same or different vendors by considering values of different criteria. It should be noted that different criteria have different significances and they are not equal in importance.

Few major performance evaluation criteria for EAF are mentioned below:

a. Tapping weight (usually described in tons and depends on furnace capacity)
b. Tap-to-Tap time
c. Power-on Time (POT)
d. Electrical energy consumption
e. Oxygen consumption
f. Graphite electrode consumption

One term that encapsulates the concept with respect to best practices is **benchmarking** that can be defined as *continuous process of comparing products, services and practices against the best competitors or enterprises that are considered as industry leaders* [7].

SGL [8] performance analysis provides the comparison of top 10 performers (10 AC furnaces operating in different regions of the world having all KPIs best-in-class) with respect to different KPIs [9]. According to their study, the best performance in terms of electrical energy consumption by electric arc furnaces that uses 100% scrap as their input is 331kWh/ton.

### 2.2 Innovations in Electric Arc Furnaces

The performance of EAFs is improving rapidly with the passage of time. During previous 50 years, different innovations were introduced to EAFs. These technological developments caused significant improvements in different KPIs. Fig. 4. depicts the whole story of this evolution. It also tells us that what types of innovations were introduced in which period of time. The result of these innovations can be seen in:

1. Reduction of electrical energy consumption from 630 kWh/t to 345 kWh/t
2. Shortening of tap-to-tap time from 180 minutes to 40 minutes
3. Electrode consumption was reduced almost 6 times from 6.5 kg/t to 1.1 kg/t

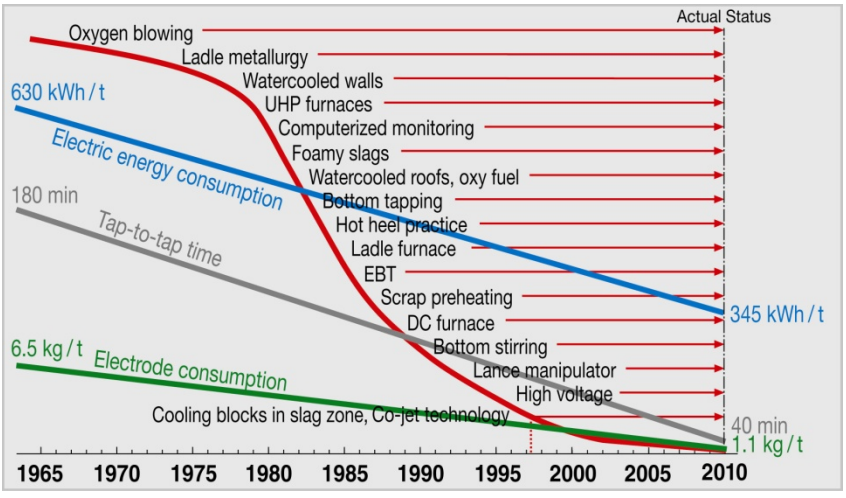



**Fig. 4.** EAF evolution until 2010 (source VDEh) [10]

The further improvement of performance is what industry is now looking at. ECOARC$^{TM}$, the electric arc furnace that is the subject of our study, used different innovations but its main feature is especially designed shaft to preheat the scrap. Recently EAF QUANTUM$^{TM}$ is claimed to be future furnace with electrical energy consumption of less than 280 kWh/ton due to its innovative trapezoidal shaped shaft design for optimum scrap distribution and efficient preheating [11]. It seems that shaft usage for scrap preheating will find more eminence in the design of future EAFs. The intelligent automated management system for such electric arc furnaces will assist in further improvement of their performance.

### 2.3 Details of the Operation

In this section, we describe only that part of operation that is relevant to our study and the data of which was used for data mining purpose. We will refer to the design of EAF having *shaft for preheating purpose* with continuous operation. Fig. 5. illustrates the sketch of a typical "ECOARC$^{TM}$", an electric arc furnace with shaft.

Scrap metal is the input to the operation of steel making and this input is to be melted for subsequent process. The scrap passes through operation of preheating in shaft before it charges the furnace [12]. Scrap preheating is carried out by thermal energy of off-gases [13]. In order to maintain certain scrap height in the shaft, scrap is charged by shaft-charging car called *"skip"*, approximately 10-13 times a *heat*.

The batches of scrap loaded on skip car are off-loaded into shaft and after preheating they are charged into furnace one by one. Getting inspiration from the skip car, we will use the word "SCIP"(SCrap In Preheating) in this paper to describe *the stage of melting period of POT during which a batch of scrap, charged by skip car, remains in lowest region of shaft before charging into furnace*. The slight change of spelling from skip to scip should be noticed.

As we have noted from Fig. 3. that melting of scrap in liquid bath is most time-consuming phase of POT. Scrap preheating makes it possible to reduce the time of this phase and hence before the immersion of scrap in liquid bath, scrap is already preheated to degree that it takes much lesser time to melt completely in liquid bath as compared to the condition when scrap is not preheated.

In order to provide required heat to melt the scrap, the electrodes are lowered onto the scrap and the scrap is subjected to high electric arc. For rapid formation of molten pool, proper adjustments of voltages and arc length throughout the melting process are required. To further accelerate the melting process, the chemical heat from combustion of oxygen plays important role but it also results in the oxidization of iron. Carbon powder injected into the bath reduces this oxidized iron and hence unacceptable drop of yield is avoided. Moreover the Carbon injection has another advantage; the *foaming of slag* (generated by small bubbles of CO floating in upward direction) that is another important innovation to increase the efficiency of electrical energy usage.

The foaming of slag also results in enhanced thermal efficiency and better arc stability. Several batches of preheated scrap are brought at the end of their respective scips from shaft into the furnace for melting in every *heat*. Before the molten steel goes through the process of tapping, it is superheated for further refinement. The *Hot Heel* [14] comprising some molten steel and slag is not tapped so that this untapped material helps to preheat and to accelerate the meltdown of the next charge of scrap.

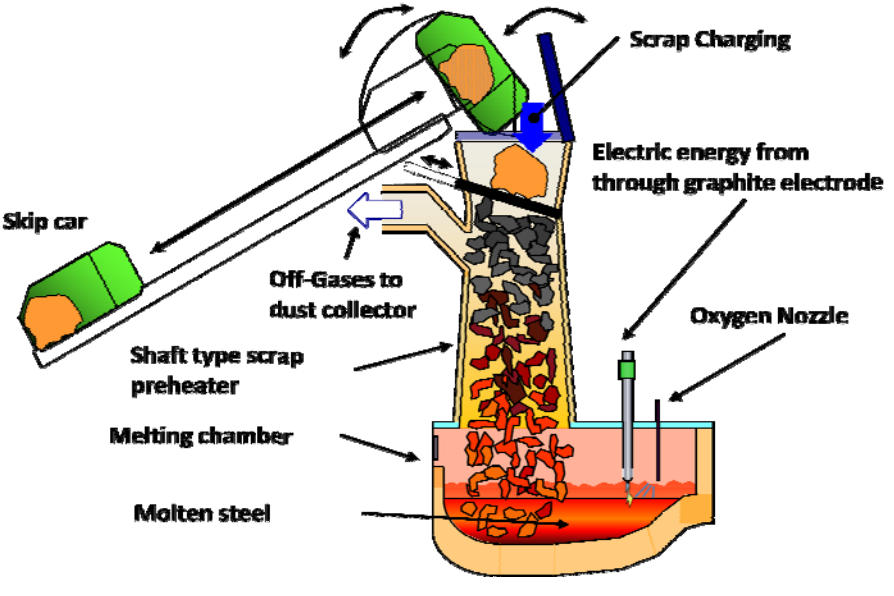



**Fig. 5.** The sketch of ECOARC ™

## 2.4 Understanding SCIPs and Stages of a *Heat*

Figure 6 illustrates the typical movement of scrap from shaft to melting chamber of the furnace. It can be seen that the shaft consists of three parts or regions through which the scrap flows before entering into melting chamber of the furnace. Off-Gases pass upward through all the three regions charged with scrap. When a new *heat* starts, portion of the scrap heated during the previous *heat* is already in the shaft. The scrap in the upper region has lowest temperature. The *heat* starts with scrap moving from lowest region into the furnace. Consequently the scrap is transferred from each region into the next region below and the emptied upper region is further charged with cold scrap.

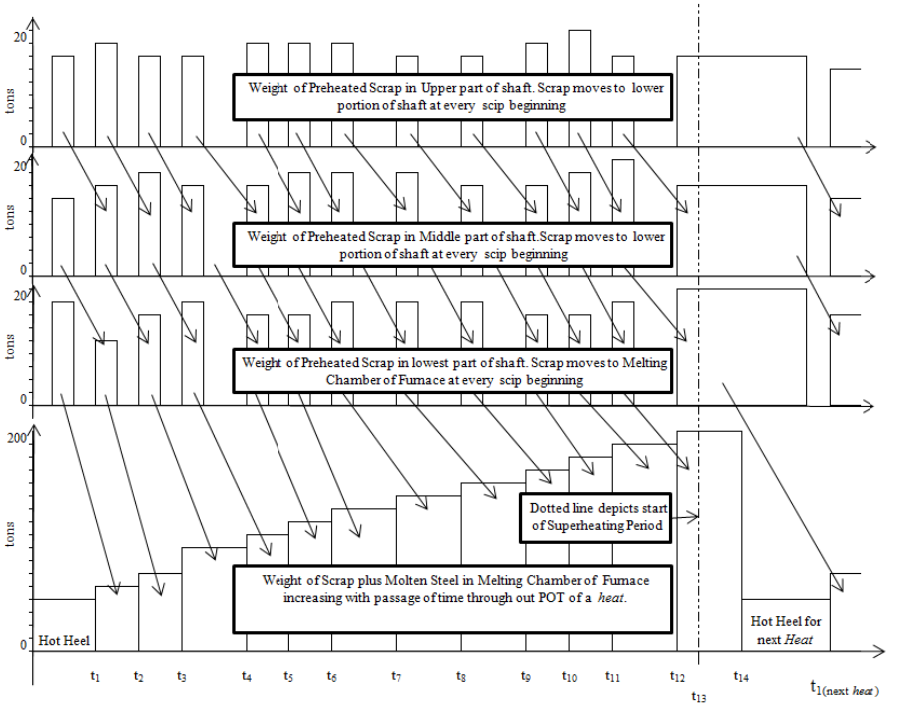



**Fig. 6.** The illustration of movement of scrap from shaft to melting chamber of the furnace in ECOARC [TM] during the POT of hypothetical *heat* comprising of 13 Stages

The time when portion of scrap enters into the melting chamber is ending time of the scip. In Fig. 6., $t_i$ (i ranges from 1 to 12) denotes the timing of ending of 12 scips. Time instance $t_{13}$ marks the beginning of superheating period and $t_{14}$ shows the time when most of molten steel is tapped and portion of molten steel is left as hot heel for the next *heat*. For this work, values of $t_{12}$ and $t_{13}$ are equal. In other words, as soon as last scip ends, the superheating period starts. Hence a *heat* with 12 scips comprises of 13 stages. Alternatively, in a *heat* with 12 scips, superheating period is the 13$^{th}$ or last stage preceded by 12 stages in melting period of POT.

## 3 Problem Description

Since we want the data mining techniques to be used to find rules for energy-efficient *heats*, we will have to establish some criterion to call a *heat* good or bad. If "O" is a vector representing operation of POT of a *heat* and "t" is the variable to be used as criterion to call a *heat* as good or bad then classification problem can be stated as follows:

*The Operation "**O**" to produce the molten steel is "Good" if the target variable "**t**" lies below the certain limit.*

Fig. 7. depicts the scenario when the criterion established for "Good and Bad Operation" is based on the value of its corresponding target variable **t**. When **t** does not exceed the threshold $\mathbf{t_{minGood}}$, the operation vector will be regarded as good operation denoted by $\mathbf{O_{Good}}$ .On the other side, if **t** becomes equal to or exceed the threshold $\mathbf{t_{minGood}}$, the operation vector will be considered as bad operation denoted by $\mathbf{O_{Bad}}$.

If the operation vector consists of **m** attributes, then our data mining problem is:

a. To identify **n** attributes (**n** ≤**m**) which have some relationship with the target variable.
b. To find the acceptable ranges of these **n** attributes so that operation vector can be classified as good.

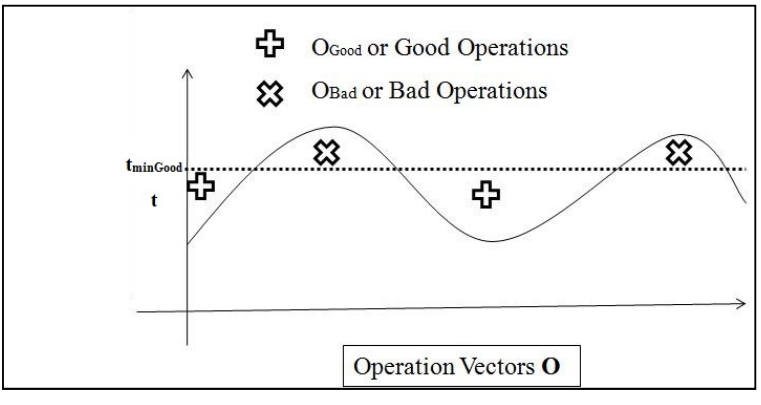



**Fig. 7.** Good and Bad Operation Vectors on the basis of a particular threshold $t_{minGood}$ described by dotted line

It is evident from the above description that *we are looking for best classifiers for this two-class problem.*

*Toulouevski*, *Yuri* N. *et al.* [13] discusses three possible ways of reducing the cost incurred at electric arc furnace described number wise as:

*1. By cutting down specific consumption of charge materials, energy carriers, refractory materials, etc., per ton of steel.*

*2. By increasing output and thus reducing specific manufacturing costs, such as maintenance staff costs, etc.*
*3. By replacing expensive charge materials and energy carriers with cheaper ones.*

In this work, we try to address the first of the above mentioned three directions by applying data mining techniques to minimize the objective variable(or the target variable **t**) *of electrical energy consumption per ton* (described by unit of *kilowatt-hours/ton*). The threshold $t_{minGood}$, considered for this work, of the objective variable for which steel making operation can be termed as *Good* is 310 kWh/ton.

In section 2.1, we have already discussed that electrical energy consumption is one of important key performance indicators (KPI) to evaluate the performance of an electric arc furnace. The top performers in EAF industry that are using pure scrap as input are consuming electrical energy in range of 331 kWh/t to 420 kWh/t in a *heat*. Thus the criterion of consumption of 310 kWh/t of electrical energy in a *heat* to evaluate the performance of operations of ECOARC$^{TM}$ can be considered as fairly competitive. It should be noted that the selection of this value depends on the suggestion of domain expert and can vary for different conditions and/or for further investigation.

Subsequently the objective variable depends on the other variables involved in the steel making operation. Our main goal in this paper will be *to describe the process to discover the acceptable ranges of these independent variables so that the objective variable's value remains below the given threshold.* In this context, we will discuss details of data mining process in coming section. Moreover it is also important to understand that what type of output this process will yield so few results will also be given as sample product of the process.

# 4 Data Mining Process

## 4.1 Data

The work in this paper was performed with 263 *heats*. While the data of 263 *heats* comprises of more than 0.5 million rows in the database, it was not enough to depict all complicated scenarios of EAF operation. The flattening of data for the purpose of data mining yielded 263 rows with one row corresponding to one *heat*. However, although limited, the available data was enough to describe the purpose of this work.

## 4.2 Data Preprocessing

Each *heat* is described by thousands of rows of time series data in the database. One row represents each second that is passed in melting the scrap or superheating the molten steel. In other words, the number of rows or tuples, describing a particular *heat*, actually denotes the duration of POT of that *heat* in seconds. Fig. 8. represents five features of time series data of POT of one sample *heat*.

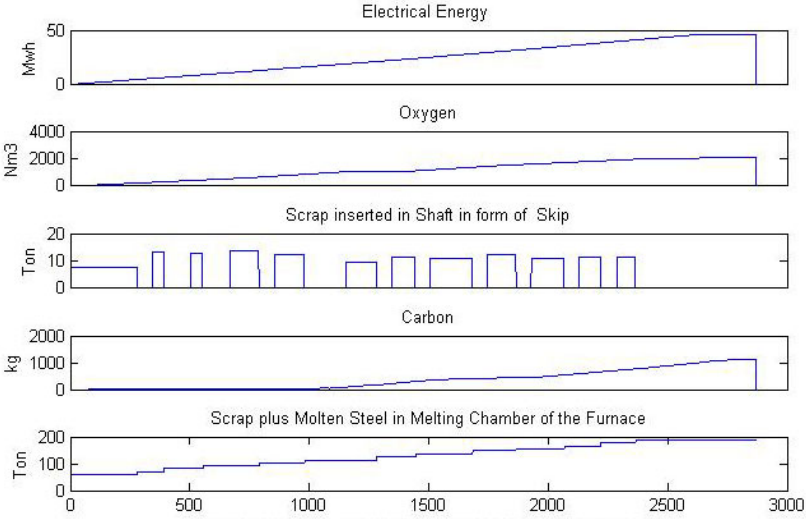



**Fig. 8.** Time Series Data with respect to five features shown in five subplots

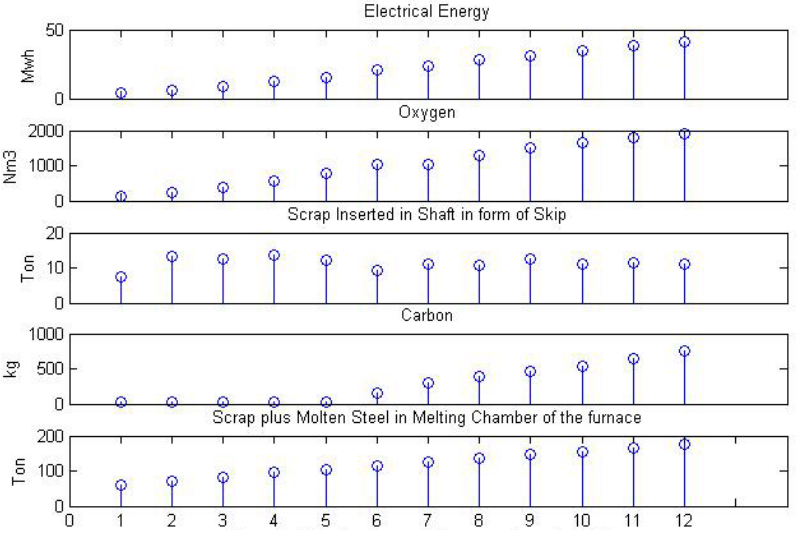



**Fig. 9.** Scip wise Data of five features shown in five subplots

Since, we are interested in transforming this time series data into form [15] appropriate for application of different data mining techniques on it; we have to identify the *important* rows for each *heat*. By important, we mean those rows that can describe the milestones of the operation. The important milestone for the *heat* is the ending time of scip when the batch of the preheated scrap to be molten is transferred

from the shaft to the melting chamber of the furnace. One *heat* usually comprises of 12 or more scips followed by superheating stage. Hence the whole heat operation can be summarized into one row with numerous columns expressing different features of every scip and superheating stage. This one row summarizing the POT of the *heat* can be classified as *good* or *bad* depending on the objective variable value. Fig. 9. demonstrates the transformation of time series data plotted in Fig. 8. into discrete stage wise data.

We used MySQL as the repository for the whole data. The data in required form was retrieved using SQL queries and then further processing was made in the spreadsheet like MS-Excel. The processed data was then given as input to WEKA [16] and different data mining methods were tested on this data. It should be noted that data of all 263 *heats* used for data mining purpose had 13 stages. It means that even if we have a single feature to be tested for checking its relation with objective variable, our input dataset will comprise of 14 columns (13 columns to describe feature value for all stages and one column to denote objective variable).

### 4.3 Attribute Pruning

Each row in input data for data mining process represents single *heat.* Every row comprises of many columns (or attributes) and every attribute belongs to certain stage, the attributes pruning was done on either of following two levels:

1. Stage pruning (the data of various stage(s) was removed to check the remaining stages impact on good or bad operation).
2. Feature Pruning(Certain features were removed of every stage to check the feature impact on good and bad operation).

If there are $n$ stages and $k$ features (i.e. total number of attributes is $n$ x $k$, then the number of possible classifiers with different combination of stages each having all features would be $\sum_{i=1}^{n} \binom{n}{i}$ or $2^n$*-1* and if use all $n$ stages but use different features, then the number of possible classifiers will be $\sum_{i=1}^{k} \binom{k}{i}$ or $2^k$ *-1*. If all classifiers with different combinations of stages and their respective features are to be discovered then the figure goes to $\sum_{i=1}^{n} \binom{n}{i}$ *x* $\sum_{i=1}^{k} \binom{k}{i}$ or *(*$2^n$ *-1)(*$2^k$*-1)*. If we treat every attribute equally with no differentiation of stage for any feature, then the number of discoverable classifiers will be $\sum_{i=1}^{n\text{x}k} \binom{n \text{ x } k}{i}$or $2^{n \text{ x } k}$ *-1*, a figure which is difficult to cope for even a small number of features of 13 or more stages.

To attain clear understanding of the problem complexity, we will give the example of input data set with single feature. If our data set contains only one feature or k =1 and n(number of stages) = 13 and we do not use any attribute selection algorithm then we have to check $2^{13}$-1 or 8383 combinations of attributes to check which combination suits best to attain "Good" *heat*.

### 4.4 The Experiments

We have carried out numerous experiments to find the relationship between different attributes of supplied data and the given result of the *heat*. From thousands of classifiers resulting from thousand of experiments, few are described below as examples.

1. To check the relationship between the *heat* and the pattern of supplied electrical energy in each stage.
2. To check the relationship between the *heat* and the following properties of each stage:
   a. Electrical power supplied during each stage
   b. Change of electrical power for each stage
   c. Duration of each stage
   d. Weight of the preheated scrap present in the shaft during each stage
   e. Weight of the steel and scrap present in the furnace during each stage
3. To check the relationship between the *heat* and the following properties of each stage:
   a. Weight of the preheated scrap that moves from the shaft into the furnace at the end of scip
   b. Oxygen in the system during each stage
   c. CO in the system during each stage
   d. $CO_2$ in the system during each stage
   e. Combustion level of the system during each stage

### 4.5 Data Mining Techniques

Mainly four data mining techniques were applied on the preprocessed data in order to find the good classifiers. J48 (Decision Tree learner [17]), JRip [18], Ridor [19] and NNge [20] were applied on different sets of data. In order to check the accuracy of the classifier, 10-folds cross-validation [21] was performed. Please note that the classifier should not only be accurate but also should be *meaningful*. By meaningful, we mean that the classifier can be incorporated into the existing system. Due to this requirement, we didn't use classifiers like Neural Networks or SVM that come out with complex answers which are difficult to be explained [22] and hard to be incorporated into the system. The brief description of the four data mining techniques is given in order to gain better understanding of results they generate.

J48 is a weka implementation to generate an unpruned or a pruned C4.5 decision tree [23]. JRip in Weka is implementation of a propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction (RIPPER), for fast and effective rule induction. The "Grow Phase" starts with empty rule set and the repeated addition of newly grown rules to this empty rule set and their pruning in "Prune Phase" results in construction of a rule set covering all positive examples. A rule is grown by greedily adding conditions to the antecedent of a rule (starting with empty antecedent) until the rule is perfect (i.e. 100% accurate). After a rule set is constructed, the algorithm goes through an optimization stage to reduce the rule set size and to improve its fitness to the training data. Over fitting is avoided by the application of combination of cross-validation and minimum-description length techniques.

RIDOR(RIpple-DOwn Rule learner) starts by generating the default rule and then the exceptions for the default rule are produced with the least (weighted) error rate followed by generation of the exceptions for each exception and this iteration continues until such exceptions are produced that have no exception.

NNge (Nearest-neighbor method of generating rules using non-nested generalized exemplars) is another rule learner algorithm meant to improve the performance of Nearest-Neighbor by grouping together those examples that share the same class in way that they are not nested(i.e. formation of non-nested generalized exemplars). It classifies new examples by determining the nearest neighbor in the exemplar database using Euclidean distance function modified to compute the distances from exemplars.

## 5 Results

We have used one feature of two stages (Electrical Energy used during stage#4 and stage#5) from the input of experiment#1 described in section 4.4 in order to illustrate the methodology to discover the classifiers that can suggest change of values of required parameters for on-going *heat* so that objective variable remains below the given threshold thus resulting in good operation. The selection of only two attributes will help us to illustrate results in two dimensional space thus ensuring better understanding of this work.

Table 1 shows the accuracy of the results obtained for the four data mining algorithm using the two attributes as input using test options of "10-folds cross-validation" and "user training set".

**Table 1.** The Accuracies shown for four Data Mining Algorithms using two test options

| Test Option \ Data Mining Algorithm | J48 | Ridor | Jrip | NNge |
|---|---|---|---|---|
| **10-folds Cross Validation** | 62.3574 % | 57.4144 % | 60.076 % | 53.6122 % |
| **User training set** | 62.3574 % | 62.3574 % | 62.3574 % | 90.1141 % |

The steel making operation is very complicated one therefore the accuracy should be high. One of the reasons to find less accurate results is lack of data. 263 rows of input data are not enough to make comparatively accurate energy-efficient rules discovery possible. However, as previously stated, even less-accurate results like given above will be enough to fulfill the purpose of this work that is the description of *methodology*. The rules discovered by each data mining algorithm are described below.

### a. J48

Fig-10 depicts a simple rule discovered by J48. The algorithm selects electrical energy used during stage#4 (EE-Scip-4) as best attribute and its value of 7.2 MW-h for split. The bad *heats* according to J48 is 175 but actually 72 good *heats* were also

misclassified as bad operation by this split. If this rule would have got a higher accuracy, then this simple rule can assist the EAF operator in the following way:

1. When the *heat* is about to enter 4th stage, the electrical energy to be supplied should be less than or equal to 7.2 MW-h
2. When the *heat* goes through 4th stage and the electrical energy exceeded 7.2 MW-h due to operational requirements then it is more likely that the on-going *heat* will be a bad operation. In the next stages, remedial actions should be taken.

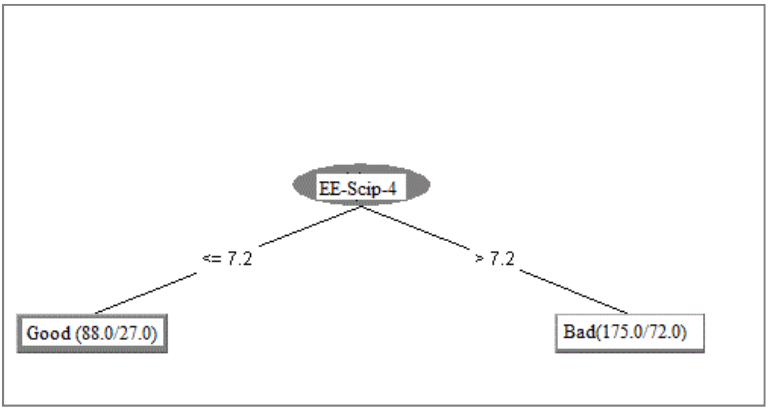



**Fig. 10.** Illustration of simple decision tree formed when J48 algorithm was applied on the data

### b. RIDOR

Following 3 rules were discovered by application of RIDOR on the same data.
Quality = Bad (263.0/133.0)

Except (EE-Scip-4 <= 7.25) and (EE-Scip-5 > 9.95) and (EE-Scip-5 <= 10.05) => Quality = Good

(4.0/0.0) [2.0/0.0]

Except (EE-Scip-4 <= 7.25) => Quality = Good (56.0/20.0) [26.0/7.0]

Out of three rules, the first one is default rule that is "Every *heat* is Bad". This default rule would have misclassified all 133 good operations to Bad ones but the exceptions give two rules for identifying good *heat*. The first rule correctly classify 6(4+2-0-0) good operations where as the second rule correctly classifies 55(56+26-20-7) good operations but misclassifies 27(20+7) bad operations to good *heats*. Fig. 11. depicts the correctly classified and misclassified charge operation of the result of RIDOR algorithm.

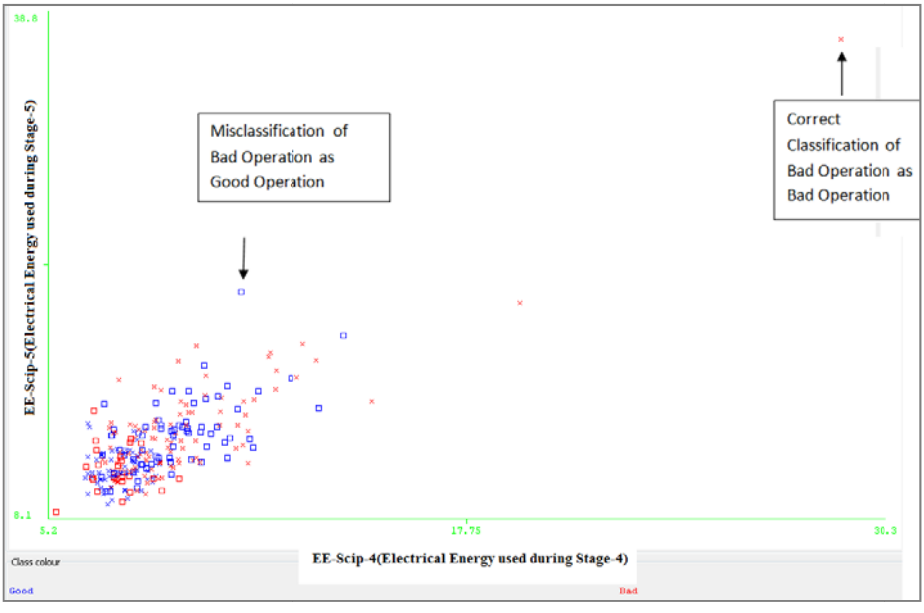



**Fig. 11.** Plot showing Correctly Classified (shown by 'x') and misclassified (shown by '□') instances when RIDOR was applied

### c. JRip

JRip algorithm produced the result very similar to J48 and is given below:

(EE-Scip-4 >= 7.3) => Quality=Bad (175.0/72.0)
=> Quality=Good (88.0/27.0)

The result can be interpreted and explained in the same manner as we have described earlier while explaining J48 result.

### d. NNge

Even though NNge result was the least accurate for 10-folds cross validation, however it came out with the most comprehensive rule set and hence for test option of supplied user training set , NNge algorithms seems to perform best, coming out with the accuracy of 90.1141%. NNge for the training data produced following number of exemplars and feature weights vector:

class Good : 50 exemplar(s) including 33 Hyperrectangle(s) and 17 Single(s).
class Bad : 49 exemplar(s) including 30 Hyperrectangle(s) and 19 Single(s).
Total : 99 exemplars(s) including 63 Hyperrectangle(s) and 36 Single(s).
Feature weights : [0.022403613664333227 0.022277589056530017]

Some of the rules discovered by NNge, that form hyperrectangles(Exemplars covering more than one instance) are given in Table 2. Fig. 12. illustrates the nature of NNge's result when exemplars were plotted in two dimensional space. In order to clearly illustrate, we have given only zoomed part of the plot where most of the exemplars exist. Even though NNge has tendency towards over-fitting, yet it can be seen as major potential rule supplier for improving the operations of EAF.

**Table 2.** The Hyperrectangles generated by NNge for two classes along with their coverage

| Rules | No of *Heats* covered |
|---|---|
| class Good IF : 7.2<=EE-Scip-4<=7.3 ^ 10.0<=EE-Scip-5<=10.6 | 11 |
| class Good IF : 6.3<=EE-Scip-4<=6.8 ^ 8.7<=EE-Scip-5<=9.6 | 7 |
| class Good IF : 7.0<=EE-Scip-4<=7.5 ^ 11.1<=EE-Scip-5<=11.3 | 7 |
| class Bad IF : 7.4<=EE-Scip-4<=7.6 ^ 10.0<=EE-Scip-5<=10.2 | 6 |
| class Good IF : 6.7<=EE-Scip-4<=7.1 ^ EE-Scip-5=10.0 | 6 |
| class Good IF : 7.1<=EE-Scip-4<=7.2 ^ 9.5<=EE-Scip-5<=9.7 | 6 |
| class Bad IF : 6.8<=EE-Scip-4<=7.3 ^ 11.9<=EE-Scip-5<=12.8 | 5 |
| class Bad IF : 7.1<=EE-Scip-4<=7.3 ^ 9.8<=EE-Scip-5<=9.9 | 5 |
| class Bad IF : 9.1<=EE-Scip-4<=9.7 ^ 11.1<=EE-Scip-5<=11.9 | 5 |
| class Bad IF : 9.7<=EE-Scip-4<=12.3 ^ 16.4<=EE-Scip-5<=17.8 | 5 |

### 5.1 Discussion

We have tried to illustrate the process to discover good classifiers by using only two attributes. The application of four data mining algorithm on the given data reveals that except NNge, all three data mining algorithms were unable to fit the training set even though they performed better than NNge when they were tested using 10-folds cross validation. It seems that in higher dimensions, in order to fit data, NNge will come out with too many exemplars however if exemplars with high coverage are selected and are carefully analyzed, they may become useful starting rule-set that can evolved further if the rules from other data mining algorithms are also incorporated.

Before the advent of computers, instructional manuals were the guide for operators to take different steps during different operating phases of *heat*. The rules from different data mining algorithms will give rise to much more detailed computerized instruction manuals for operators. Such online guide will be milestone is development of intelligent automated management system that will be able to learn and improve this online guide after getting data from each passing *heat*.

In order to develop intelligent automated management system, the different challenges that must be answered are as follows:

1. How to present the different rules during different stages of *heat* in front of operator in the way that it become convenient for him/her to make swift decisions rather than causing confusion.
2. How to develop new classifiers containing combination of rules generated from application of different data mining algorithms.

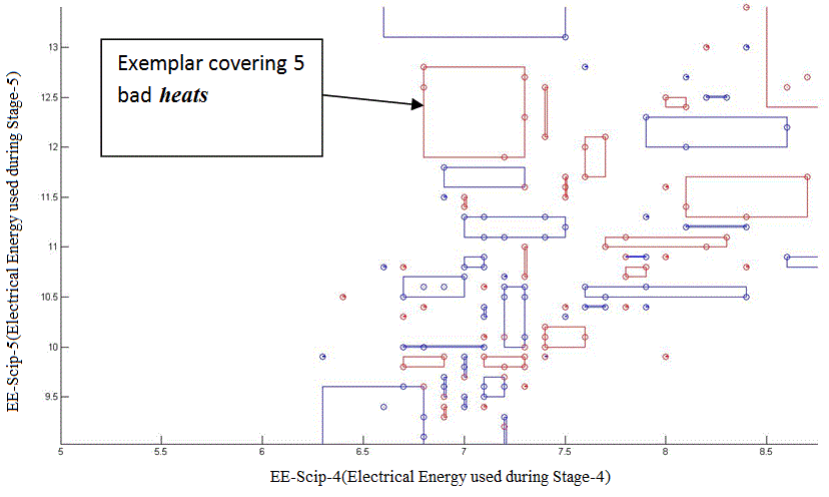



**Fig. 12.** Plot of NNge's exemplars (Red color denotes Bad Heat where as Blue Color denotes Good Heat)

## 6 Concluding Remarks

This paper has described methodology on the basis of which good classifiers are expected to be discovered. The rules from the highly accurate classifiers can become part of intelligent automated management system that can :

1. Assist EAF operators in assessing the situation of on-going *heat.*
2. Guide EAF operators to take required steps to improve on-going *heat*
3. Learn and improve by applying data mining techniques on newly accumulated experiences.

The application of data mining techniques on the flattened time series data has potential for such classifiers discovery. As for our future investigation, we would like to include the following topics:

*Number of Classes*: Rather than treating each *heat* as either good or bad, further classes can be formed and tested on the basis of different values of objective variable.

*Time Series Data Mining*: In this work, we have transformed the whole time series data into rectangular static table with one row for each *heat*. However, there is a room for directly applying different time series mining algorithms too.

**Acknowledgements.** We would like to thank the members of Steel Plantech Co., who allowed us to use the steel plant operation data and gave valuable suggestions for the *work*.

## References


[1] Toulouevski, Y.N., Zinurov, I.Y.: Preheating of Scrap by Burner and Off-Gases. In: Innovation in Electric Arc Furnaces, Scientific Basis for Selection, pp. 110–111. Springer, Heidelberg (2010)

[2] Bell, S., et al.: Final Report on Energy Efficiency and Energy Savings in Recycling, p. 8 (2006); 2005-43(CF), 8

[3] Ao, H., et al.: 'ECOARC' Technology. In: 58th Electric Arc Furnace Conference, Iron and Steel Society, Warrendale, Pa, USA, pp. 325–336 (2000)

[4] Peaslee, K.D., et al.: Final Technical Report:Development of a Process to Continuously Melt, Refine, and Cast High Quality Steel, DE-FC36-03ID14279, p. 12 (2006)

[5] Sandberg, E.: Section 2.2: Variable availability.Thesis: Energy and Scrap Optimisation of Electric Arc Furnaces by Statistical Analysis of Process Data (2005)

[6] Toulouevski, Y.N., Zinurov, I.Y.: Principles of Automation of Heat Control. In: Innovations in Electric Arc Furnaces; Scientific Basis for Selection, pp. 227–228. Springer, Heidelberg (2010)

[7] Parker, S.: Measuring up: Size is No Obstacle to Benchmarking for Competitive. Rochester Business Journal 8 (1996)

[8] SGL Group; The Carbon Company, http://www.sglgroup.com/cms/international/home/index.html?__locale=en (Cited: June 15, 2011)

[9] Schult, H.: Benchmark Analysis - A Tool To Define The Next Development Steps? SEAISI, Singapore (2011)

[10] Lüngen, B., Harste, K.: Quo Vadis Stahlindustrie. Stahl und Eisen (January 2011)

[11] Dorndorf, M., et al.: Simetal Eaf Quantum – The Future Approach For Efficient Scrap Melting. SEAISI, Singapore (2011)

[12] Mizukami, H., et al.: Off-gas Treatment Technology of ECOARC. NKK Tehnical Report, No.176, 2002(3), pp. 1–5 (2002) (in Japanese)

[13] Toulouevski, Y.N., Zinurov, I.Y.: Modern Steelmaking in Electric Arc Furnaces: History and Prospects for Development. In: Innovation in Electric Arc Furnaces, Scientific Basis for Selection, p. 2. Springer, Heidelberg (2010)

[14] Toulouevski, Y.N., Zinurov, I.Y.: Furnace Operation with Hot Heel. In: Innovation in Electric Arc Furnaces, Scientific Basis for Selection, pp. 16–17. Springer, Heidelberg (2010)

[15] Jiawei, H., Micheline, K.: Data Mining Concepts and Techniques, p. 7. Morgan Kaufmann, San Francisco

[16] Hall, M., et al.: The WEKA Data Mining Software: An Update. SIGKDD Explorations 11(1) (2009)

[17] Jiawei, H., Micheline, K.: Classification by Decision Tree Induction. Data Mining Concepts and Techniques, 6.3

[18] Cohen, W.W.: In Fast effective rule induction. In: Prieditis, A., Russell, S.J. (eds.) Proceedings of the Twelfth International Conference on Machine Learning In Fast effective rule induction, Tahoe City, California, July 9-12, pp. 115–123. Morgan Kaufmann, San Francisco (1995)

[19] Gaines, B.R., Compton, P.: Induction of ripple-down rules applied. J. Intell. Inf. Syst. 5, 211–228 (1995)

[20] Martin, B.: Thesis: Instance based Learning; Nearest Neighbour with Generalization (1995)
[21] Bishop, C.M.: Model Selection. Pattern Recognition and Machine Learning, 1.3
[22] Jiawei, H., Micheline, K.: Data Mining Concepts and Techniques, vol. 2, pp. 334–335. Morgan Kaufmann, San Francisco
[23] Quinlan, R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo (1993)

# Author Index