Soft Computing for Knowledge Discovery and Data Mining

Soft Computing for Knowledge Discovery and Data Mining

edited by

Oded Maimon
Tel-Aviv University
Israel

and

Lior Rokach
Ben-Gurion University of the Negev
Israel

Springer

Oded Maimon
Tel Aviv University
Dept.of Industrial Engineering
69978 TEL-AVIV
ISRAEL
maimon@eng.tau.ac.il

Lior Rokach
Ben-Gurion University
Dept. of Information System Engineering
84105 BEER-SHEVA
ISRAEL
liorrk@bgu.ac.il

Library of Congress Control Number: 2007934794

Soft Computing for Knowledge Discovery and Data Mining
Edited by Oded Maimon and Lior Rokach

ISBN 978-0-387-69934-9 e-ISBN 978-0-387-69935-6

Printed on acid-free paper.

9 8 7 6 5 4 3 2 1

springer.com

To my family
– O.M.

To my wife Ronit, and my two boys, Yarden and Roy
– L.R.

Preface

The information age has made it easy to store large amounts of data. Data mining is a new and exciting field that tries to solve the crisis of information overload by exploring large and complex bodies of data in order to discover useful patterns. It is extreme importance because it enables modeling and knowledge extraction from abundance data availability. Therefore theoreticians and practitioners are continually seeking techniques to make the process more efficient, cost-effective and accurate. Among the more promising technique that have emerged in recent years are soft computing methods such as fuzzy sets, artificial neural networks, genetic algorithms. These techniques exploit a tolerance for imprecision, uncertainty and partial truth to achieve tractability, robustness and low cost solutions. This book shows that the soft computing methods extend the envelope of problems that data mining can solve efficiently.

This book presents a comprehensive discussion of the state of the art in data mining along with the main soft computing techniques behind it. In addition to presenting a general theory of data mining, the book provides an in-depth examination of core soft computing algorithms.

To help interested researchers and practitioners who are not familiar with the field, the book starts with a gentle introduction to data mining and knowledge discovery in databases (KDD) and prepares the reader for the next chapters. The rest of the book is organized into four parts. The first three parts devoted to the principal constituents of soft computing: neural networks, evolutionary algorithms and fuzzy logic. The last part compiles the recent advances in soft computing and data mining.

This book was written to provide investigators in the fields of information systems, engineering, computer science, statistics and management, with a profound source for the role of soft computing in data mining. In addition, social sciences, psychology, medicine, genetics, and other fields that are interested in solving complicated problems can much benefit from this book. The book can also serve as a reference book for graduate / advanced undergraduate level courses in data mining and machine learning. Practitioners among the readers may be particularly interested in the descriptions of real-world data mining projects performed with soft-computing.

We would like to thank all authors for their valuable contributions. We would like to express our special thanks to Susan Lagerstrom-Fife and Sharon Palleschi of Springer for working closely with us during the production of this book.

Tel-Aviv, Israel *Oded Maimon*
Beer-Sheva, Israel *Lior Rokach*

July 2007

Contents

Introduction to Soft Computing for Knowledge Discovery and Data Mining
Oded Maimon, Lior Rokach 1

Part I Neural Network Methods

Neural Networks For Data Mining
G. Peter Zhang 17

Improved SOM Labeling Methodology for Data Mining Applications
Arnulfo Azcarraga, Ming-Huei Hsieh, Shan-Ling Pan, Rudy Setiono 45

Part II Evolutionary Methods

A Review of Evolutionary Algorithms for Data Mining
Alex A. Freitas 79

Genetic Clustering for Data Mining
Murilo Coelho Naldi André Carlos Ponce de Leon Ferreira de Carvalho Ricardo José Gabrielli Barreto Campello Eduardo Raul Hruschka 113

Discovering New Rule Induction Algorithms with Grammar-based Genetic Programming
Gisele L. Pappa, Alex A. Freitas 133

Evolutionary Design of Code-matrices for Multiclass Problems
Ana Carolina Lorena, André C. P. L. F. de Carvalho 153

Part III Fuzzy Logic Methods

The Role of Fuzzy Sets in Data Mining
Lior Rokach 187

Support Vector Machines and Fuzzy Systems
Yixin Chen 205

KDD in Marketing with Genetic Fuzzy Systems
Jorge Casillas, Francisco J. Martínez-López 225

Knowledge Discovery in a Framework for Modelling with Words
Zengchang Qin, Jonathan Lawry 241

Part IV Advanced Soft Computing Methods and Areas

Swarm Intelligence Algorithms for Data Clustering
Ajith Abraham, Swagatam Das, Sandip Roy 279

A Diffusion Framework for Dimensionality Reduction
Alon Schclar 315

Data Mining and Agent Technology: a fruitful symbiosis
Christos Dimou, Andreas L. Symeonidis,, Pericles A. Mitkas 327

Approximate Frequent Itemset Mining In the Presence of Random Noise
Hong Cheng, Philip S. Yu, Jiawei Han 363

The Impact of Overfitting and Overgeneralization on the Classification Accuracy in Data Mining
Huy Nguyen Anh Pham, Evangelos Triantaphyllou 391

Index 433

List of Contributors

Ajith Abraham
Center of Excellence for Quantifiable
Quality of Service (Q2S),
Norwegian University of Science and
Technology,
Trondheim, Norway
`ajith.abraham@ieee.org`

Arnulfo Azcarraga
College of Computer Studies,
De La Salle University, Manila,
The Philippines
`azcarragaa`
`@canlubang.dlsu.edu.ph`

Ricardo José Gabrielli Barreto Campello
Instituto de Ciências Matemáticas e
de Computação,
Universidade de São Paulo
`campello@icmc.usp.br`

André Carlos Ponce de Leon Ferreira de Carvalho
Instituto de Ciê
ncias Matemá
ticas e de Computação
Universidade de São Paulo
`andre@icmc.usp.br`

Jorge Casillas
Dept. of Computer Science and
Artificial Intelligence,
University of Granada,
Spain
`casillas@decsai.ugr.es`

Yixin Chen
Dept. of Computer and Information
Science
The University of Mississippi
MS 38655
`ychen@cs.olemiss.edu`

Hong Cheng
University of Illinois at Urbana-Champaign
`hcheng3@cs.uiuc.edu`

Swagatam Das
Dept. of Electronics and Telecommunication Engineering,
Jadavpur University,
Kolkata 700032,
India.

Christos Dimou
Electrical and Computer Engineering Dept.
Aristotle University of Thessaloniki,
54 124, Thessaloniki,
Greece
cdimou@issel.ee.auth.gr

Alex A. Freitas
Computing Laboratory,
University of Kent,
Canterbury, Kent, CT2 7NF, UK
A.A.Freitas@kent.ac.uk

Jiawei Han
University of Illinois at Urbana-Champaign
hanj@cs.uiuc.edu

Eduardo Raul Hruschka
eduardo.hruschka
@pesquisador.cnpq.br

Ming-Huei Hsieh
Dept. of International Business,
National Taiwan University,
Taiwan
mhhsieh@management.ntu.edu.tw

Jonathan Lawry
Artificial Intelligence Group,
Department of Engineering Mathematics,
University of Bristol,
BS8 1TR, UK.
j.lawry@bris.ac.uk

Ana Carolina Lorena
Centro de Matemática,
Computação e Cognição
Universidade Federal do ABC
Rua Catequese, 242,
Santo André, SP, Brazil
ana.lorena@ufabc.edu.br

Oded Maimon
Dept. of Industrial Engineering
Tel-Aviv University
Israel
maimon@eng.tau.ac.il

Francisco J. Martínez-López
Dept. of Marketing, University of Granada, Spain
fjmlopez@ugr.es

Murilo Coelho Naldi
Instituto de Ciê
ncias Matemá
ticas e de Computação
Universidade de São Paulo
murilocn@icmc.usp.br

Shan-Ling Pan
School of Computing,
National University of Singapore,
Singapore
pansl@comp.nus.edu.sg

Gisele L. Pappa
Computing Laboratory
University of Kent
Canterbury, Kent, CT2 7NF, UK
glp6@kent.ac.uk

Huy Nguyen Anh Pham
Dept. of Computer Science,
298 Coates Hall,
Louisiana State University,
Baton Rouge, LA 70803
hpham15@lsu.edu

Zengchang Qin
Berkeley Initiative in Soft Computing (BISC),
Computer Science Division,
EECS Department,
University of California,
Berkeley, CA 94720, US.

zqin@eecs.berkeley.edu

Lior Rokach
Dept. of Information System Engineering,
Ben-Gurion University,
Israel
liorrk@bgu.ac.il

Sandip Roy
Dept. of Computer Science and Engineering,
Asansol Engineering College,
Asansol-713304, India.

Alon Schclar
School of Computer Science,
Tel Aviv University,
Tel Aviv 69978,
Israel
shekler@post.tau.ac.il

Rudy Setiono
School of Computing,
National University of Singapore,
Singapore
rudys@comp.nus.edu.sg

Andreas L. Symeonidis
Electrical and Computer Engineering Dept.
Aristotle University of Thessaloniki,
54 124, Thessaloniki,
Greece
asymeon@iti.gr

Pericles A. Mitkas
Electrical and Computer Engineering Dept.
Aristotle University of Thessaloniki,
54 124, Thessaloniki,
Greece
mitkas@eng.auth.gr

Evangelos Triantaphyllou
Dept. of Computer Science,
298 Coates Hall,
Louisiana State University,
Baton Rouge, LA 70803
trianta@lsu.edu

Philip S. Yu
IBM T. J. Watson Research Center
psyu@us.ibm.com

G. Peter Zhang
Georgia State University,
Dept. of Managerial Sciences
gpzhang@gsu.edu

Introduction to Soft Computing for Knowledge Discovery and Data Mining

Oded Maimon[1] and Lior Rokach[2]

[1] Department of Industrial Engineering, Tel-Aviv University, Ramat-Aviv 69978,
Israel,
maimon@eng.tau.ac.il
[2] Department of Information System Engineering, Ben-Gurion University,
Beer-Sheba, Israel,
liorrk@bgu.ac.il

Summary. In this chapter we introduce the Soft Computing areas for Data Mining and the Knowledge Discovery Process, discuss the need for plurality of methods, and present the book organization and abstracts.

1 Introduction

Data Mining is the science, art and technology of exploring data in order to discover insightful unknown patterns. It is a part of the overall process of Knowledge Discovery in Databases (KDD). The accessibility and abundance of information today makes data mining a matter of considerable importance and necessity.

Soft computing is a collection of new techniques in artificial intelligence, which exploit the tolerance for imprecision, uncertainty and partial truth to achieve tractability, robustness and low solution cost. Given the history and recent growth of the field, it is not surprising that several mature soft computing methods are now available to the practitioner, including: fuzzy logic, artificial neural networks, genetic algorithms, and swarm intelligence. The aims of this book are to present and explain the important role of soft computing methods in data mining and knowledge discovery.

The unique contributions of this book is in the introduction of soft computing as a viable approach for data mining theory and practice, the detailed descriptions of novel soft-computing approaches in data mining, and the illustrations of various applications solved in soft computing techniques, including: Manufacturing, Medical, Banking, Insurance, Business Intelligence and others. The book does not include some of the most standard techniques in Data Mining, such as Decision Trees (the reader is welcome to our new book, from 2007, dedicated entirely to Decision Trees). The book include the leading soft computing methods, though for volume reasons it could not cover all methods, and there are further emerging techniques, such as fractal based data mining (a topic of our current research).

Since the information age, the accumulation of data has become easier and storing it inexpensive. It has been estimated that the amount of stored information doubles less than twenty months. Unfortunately, as the amount of electronically stored information increases, the ability to understand and make use of it does not keep pace with its growth. Data Mining is a term coined to describe the process of sifting through large databases for interesting patterns and relationships. The studies today aim at evidence-based modeling and analysis, as is the leading practice in medicine, finance, intelligence and many other fields. Evidently, in the presence of the vast techniques' repertoire and the complexity and diversity of the explored domains, one real challenge today in the data mining field is to know how to utilize this repertoire in order to achieve the best results. The book shows that the soft computing methods extend the envelope of problems that data mining can solve efficiently. The techniques of soft computing are important for researchers in the fields of data mining, machine learning, databases and information systems, engineering, computer science and statistics.

This book was written to provide investigators in the fields of information systems, engineering, computer science, statistics and management, with a profound source for the role of soft computing in data mining. In addition, social sciences, psychology, medicine, genetics, and other fields that are interested in solving complicated problems can much benefit from this book. Practitioners among the readers may be particularly interested in the descriptions of real-world data mining projects performed with soft computing.

The material of this book has been taught by the authors in graduate and undergraduate courses at Tel-Aviv University and Ben-Gurion University. The book can also serve as a reference book for graduate and advanced undergraduate level courses in data mining and machine learning.

In this introductory chapter we briefly present the framework and overall knowledge discovery process in the next two sections, and then the logic and organization of this book, with brief description of each chapter.

2 The Knowledge Discovery process

This book is about methods, which are the core of the Knowledge Discovery process. For completion we briefly present here the process steps. The knowledge discovery process is iterative and interactive, consisting of nine steps.

Note that the process is iterative at each step, meaning that moving back to previous steps may be required. The process has many "artistic" aspects in the sense that one cannot present one formula or make a complete taxonomy for the right choices for each step and application type. Thus it is required to understand the process and the different needs and possibilities in each step.

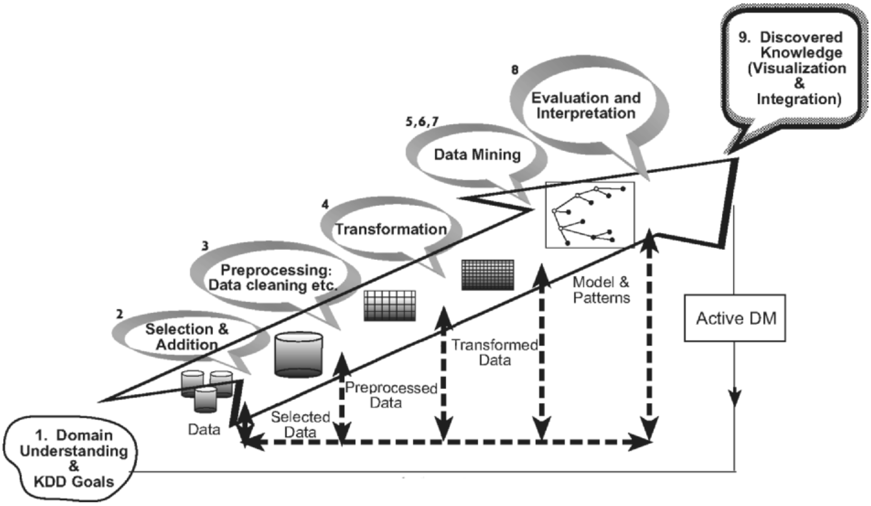
9. Discovered Knowledge (Visualization & Integration)
8
Evaluation and Interpretation
5,6,7
Data Mining
4
Transformation
3
Preprocessing: Data cleaning etc.
2
Selection & Addition
Model & Patterns
Active DM
Transformed Data
Preprocessed Data
Selected Data
Data
1. Domain Understanding & KDD Goals

Fig. 1. The Process of Knowledge Discovery in Databases.

The process starts with determining the KDD goals, and "ends" with the implementation of the discovered knowledge. Then the loop is closed - the Active Data Mining part starts. As a result, changes can be made in the application domain (such as offering different features to mobile phone users in order to reduce churning). This closes the loop, and the effects are then measured on the new data repositories, and the KDD process is launched again.

Following is a brief description of the nine-step KDD process, starting with a managerial step:

1. Developing an understanding of the application domain: This is the initial preparatory step. It prepares the scene for understanding what should be done with the many decisions (about transformations, algorithms, representation, etc.). The people who are in charge of a KDD project need to understand and define the goals of the end-user and the environment in which the knowledge discovery process will take place (including relevant prior knowledge). As the KDD process proceeds, there may be even a revision of this step.
   Having understood the KDD goals, the preprocessing of the data starts, defined in the next three steps.
2. Selecting and creating a data set on which discovery will be performed: Having defined the goals, the data that will be used for the knowledge discovery should be determined. This includes finding out what data is available, obtaining additional necessary data, and then integrating all the data for the knowledge discovery into one data set, including the attributes

that will be considered for the process. This process is very important because the Data Mining learns and discovers from the available data. This is the evidence base for constructing the models.

3. Preprocessing and cleansing: In this stage, data reliability is enhanced. It includes data clearing, such as handling missing values and removal of noise or outliers. There are many methods explained in the handbook, from doing almost nothing to becoming the major part (in terms of time consumed) of a KDD project in certain projects. It may involve complex statistical methods or using a Data Mining algorithm in this context. For example, if one suspects that a certain attribute is of insufficient reliability or has many missing data, then this attribute could become the goal of a data mining supervised algorithm, or finding the centroids of clustering. A prediction model for this attribute will be developed, and then missing data can be predicted. The extension to which one pays attention to this level depends on many factors. In any case, studying the aspects is important and often revealing by itself, regarding complex information systems.
4. Data transformation: In this stage, the generation of better data for the data mining is prepared and developed. Methods here include dimension reduction (such as feature selection and record sampling), and attribute transformation (such as discretization of numerical attributes and functional transformation). This step can be crucial for the success of the entire KDD project, and it is usually very project-specific. For example, in medical examinations, the quotient of attributes may often be the most important factor, and not each one by itself. In marketing, we may need to consider effects beyond our control as well as efforts and temporal issues (such as studying the effect of advertising accumulation). However, even if we do not use the right transformation at the beginning, we may obtain a surprising effect that hints to us about the transformation needed (in the next iteration). Thus the KDD process reflects upon itself and leads to an understanding of the transformation needed.

   Having completed the above four steps, the following four steps are related to the Data Mining part, where the focus is on the algorithmic aspects employed for each project:
5. Choosing the appropriate Data Mining task: We are now ready to decide on which type and approach of Data Mining to use, for example, classification, regression, or clustering. This mostly depends on the KDD goals, and also on the previous steps. There are two major goals in Data Mining: prediction and description. Prediction is often referred to as supervised Data Mining, while descriptive Data Mining includes the unsupervised and visualization aspects of Data Mining. Most Data Mining techniques are based on inductive learning, where a model is constructed explicitly or implicitly by generalizing from a sufficient number of training examples. The underlying assumption of the inductive approach is that the trained

model is applicable to future cases. The strategy also takes into account the level of meta-learning for the particular set of available data.

6. Choosing the Data Mining algorithm: Having the strategy, we now decide on the tactics. This stage includes selecting the specific method to be used for searching patterns (including multiple inducers). For example, in considering precision versus understandability, the former is better with neural networks, while the latter is better with decision trees. For each strategy of meta-learning there are several possibilities of how it can be accomplished. Meta-learning focuses on explaining what causes a Data Mining algorithm to be successful or not in a particular problem. Thus, this approach attempts to understand the conditions under which a Data Mining algorithm is most appropriate. Each algorithm has parameters and tactics of learning (such as ten-fold cross-validation or another division for training and testing).
7. Employing the Data Mining algorithm: Finally the implementation of the Data Mining algorithm is reached. In this step we might need to employ the algorithm several times until a satisfied result is obtained, for instance by tuning the algorithm's control parameters, such as the minimum number of instances in a single leaf of a decision tree.
8. Evaluation: In this stage we evaluate and interpret the mined patterns (rules, reliability, etc.), with respect to the goals defined in the first step. Here we consider the preprocessing steps with respect to their effect on the Data Mining algorithm results (for example, adding features in Step 4 and repeating from there). This step focuses on the comprehensibility and usefulness of the induced model. In this step the discovered knowledge is also documented for further usage.

   The last step is the usage and overall feedback on the patterns and discovery results obtained by the Data Mining:
9. Using the discovered knowledge: We are now ready to incorporate the knowledge into another system for further action. The knowledge becomes active in the sense that we may make changes to the system and measure the effects. Actually the success of this step determines the effectiveness of the entire KDD process. There are many challenges in this step, such as loosing the "laboratory conditions" under which we have operated. For instance, the knowledge was discovered from a certain static snapshot (usually sample) of the data, but now the data becomes dynamic. Data structures may change (certain attributes become unavailable), and the data domain may be modified (such as, an attribute may have a value that was not assumed before).

3 The need for plurality of methods

Data Mining methods are becoming part of general purpose Integrated Information Technology (IIT) software packages. Starting from the data sources (such as operational databases, semi- and non-structured data and reports, Internet sites etc.), then the tier of the data warehouse, followed by OLAP (On Line Analytical Processing) servers and concluding with analysis tools, where Data Mining tools are the most advanced.

We can naively distinguish among three levels of analysis. The simplest one is achieved by report generators (for example, presenting all claims that occurred because of a certain cause last year, such as car theft). We then proceed to OLAP multi-level analysis (for example presenting the ten towns where there was the highest increase of vehicle theft in the last month as compared to with the month before). Finally a complex analysis is carried out for discovering the patterns that predict car thefts in these cities, and what might occur if anti theft devices were installed. The latter is based on modeling of the phenomena, where the first two levels are ways of data aggregation and fast manipulation.

Empirical comparison of the performance of different approaches and their variants in a wide range of application domains has shown that each performs best in some, but not all, domains. This phenomenon is known as the selective superiority problem, which means, in our case, that no induction approach or algorithm can be the best in all possible domains. The reason is that each algorithm contains an explicit or implicit bias that leads it to prefer certain generalizations over others, and it will be successful only as long as this bias matches the characteristics of the application domain.

Results have demonstrated the existence and correctness of this “no free lunch theorem”. If one inducer is better than another in some domains, then there are necessarily other domains in which this relationship is reversed. This implies in KDD that for a given problem a certain approach can yield more knowledge from the same data than other approaches.

In many application domains, the generalization error (on the overall domain, not just the one spanned in the given data set) of even the best methods is far above the training set, and the question of whether it can be improved, and if so how, is an open and important one. Part of the answer to this question is to determine the minimum error achievable by any classifier in the application domain (known as the optimal Bayes error). If existing classifiers do not reach this level, new approaches are needed. Although this problem has received considerable attention, no generally reliable method has so far been demonstrated. This is one of the challenges of the DM research – not only to solve it, but even to quantify and understand it better. Heuristic methods can then be compared absolutely and not just against each other.

A subset of this generalized study is the question of which approach and inducer to use for a given problem. To be even more specific, the performance measure need to be defined appropriately for each problem. Though there are some commonly accepted measures it is not enough. For example, if the analyst is looking for accuracy only, one solution is to try each one in turn, and by estimating the generalization error, to choose the one that appears to perform best. Another approach, known as multi-strategy learning, attempts to combine two or more different paradigms in a single algorithm.

The dilemma of what method to choose becomes even greater if other factors such as comprehensibility are taken into consideration. For instance, for a specific domain, neural networks may outperform decision trees in accuracy. However from the comprehensibility aspect, decision trees are considered superior. In other words, in this case even if the researcher knows that neural network is more accurate, the dilemma of what methods to use still exists (or maybe to combine methods for their separate strength).

Induction is one of the central problems in many disciplines such as machine learning, pattern recognition, and statistics. However the feature that distinguishes Data Mining from traditional methods is its scalability to very large sets of varied types of input data. Scalability means working in an environment of high number of records, high dimensionality, and a high number of classes or heterogeneousness. Nevertheless, trying to discover knowledge in real life and large databases introduces time and memory problems.

As large databases have become the norms in many fields (including astronomy, molecular biology, finance, marketing, health care, and many others), the use of Data Mining to discover patterns in them has become potentially very beneficial for the enterprise. Many companies are staking a large part of their future on these "Data Mining" applications, and turn to the research community for solutions to the fundamental problems they encounter.

While a very large amount of available data used to be the dream of any data analyst, nowadays the synonym for "very large" has become "terabyte" or "pentabyte", a barely imaginable volume of information. Information-intensive organizations (like telecom companies and financial institutions) are expected to accumulate pentabyte of raw data every one to two years.

High dimensionality of the input (that is, the number of attributes) increases the size of the search space in an exponential manner (known as the "Curse of Dimensionality"), and thus increases the chance that the inducer will find spurious classifiers that in general are not valid. There are several approaches for dealing with a high number of records including: sampling methods, aggregation, massively parallel processing, and efficient storage methods. This book presents some of the approaches in this direction.

4 The organization of the book

The book has sixteen chapters divided into four main parts, where the first three address the methods and topics that are most identified with soft computing, and then the last part adds advanced and promising methods and areas:

I. Neural network methods: Chapters 2 to 3
II. Evolutionary methods: Chapters 4 to 7

III. Fuzzy logic methods: Chapters 8 to 11
IV. Advanced soft computing methods and areas: Chapters 12 to 16 Including: Swarm intelligence (12), diffusion process (13), and agent technology (14); and the areas of: approximate frequent item-set mining (15), and finally the impact of over-fitting and over-generalization on the classification accuracy in Data Mining (16).

In the following, edited abstracts of the chapters in the book are presented, for the reader map and convenience:

4.1 Neural network methods

The first methodology addressed in the book is **Neural Networks**, which have become elaborated important tools for data mining. Chapter 2 provides an overview of neural network models and their applications to data mining tasks. It also provides historical development of the field of neural networks and present three important classes of neural models including feed forward multilayer networks, Hopfield networks, and Kohonen's self-organizing maps. Modeling issues and applications of these models for data mining are discussed as well.

Then Chapter 3 continues in this direction by specifically addressing **Self-Organizing Maps** (SOMs). SOMs have been useful in gaining insights about the information content of large volumes of data in various data mining applications. As a special form of neural networks, they have been attractive as a data mining tool because they are able to extract information from data even with very little user-intervention (though some is needed). This chapter proposes a methodical and semi-automatic SOM labeling procedure that does not require a set of labeled patterns, and shows an effective alternative. The effectiveness of the method is demonstrated on a data mining application involving customer-profiling based on an international market segmentation study.

4.2 Evolutionary methods

A new family of methods starts in Chapter 4 with a review of **Evolutionary Algorithms** (EAs) for Data Mining. Evolutionary Algorithms are stochastic search algorithms inspired by the process of neo-Darwinian evolution. The motivation for applying EAs to data mining is that they are robust, adaptive search techniques that perform a global search in the solution space. This chapter first presents a brief overview of EAs, focusing mainly on two kinds of EAs, viz. Genetic Algorithms (GAs) and Genetic Programming (GP). Then the chapter reviews the main concepts and principles used by EAs designed for solving several data mining tasks, namely: discovery of classification rules, clustering, attribute selection and attribute construction. Finally, it discusses

Multi-Objective EAs, based on the concept of Pareto dominance, and their use in several data mining tasks.

Then Chapter 5 continues this topic by specifically addressing **Genetic Clustering** for Data Mining. Genetic Algorithms (GAs) have been successfully applied to several complex data analysis problems in a wide range of domains, such as image processing, bioinformatics, and crude oil analysis. The need for organizing data into categories of similar objects has made the task of clustering increasingly important to those domains. This chapter presents a survey of the use of GAs for clustering applications. A variety of encoding (chromosome representation) approaches, fitness functions, and genetic operators are described, all of them customized to solve problems in such an application context.

Chapter 6 addresses the discovering of new rule by induction algorithms with **Grammar-Based Genetic Programming**. Rule induction is a data mining technique used to extract classification rules of the form IF (conditions) THEN (predicted class) from data. The majority of the rule induction algorithms found in the literature follow the sequential covering strategy, which essentially induces one rule at a time until (almost) all the training data is covered by the induced rule set. This strategy describes a basic algorithm composed by several key elements, which can be modified to generate new and better rule induction algorithms. With this in mind, this work proposes the use of a **Grammar-based Genetic Programming** (GGP) algorithm to automatically discover new sequential covering algorithms. The proposed system is evaluated using 20 data sets, and the automatically-discovered rule induction algorithms are compared with four well-known human-designed rule induction algorithms. Results showed that the GGP system is a promising approach to effectively discover new sequential covering algorithms

Another general aspect of data mining issues is introduced in Chapter 7 with **Evolutionary Design** of code-matrices for multi-class problems. Given a dataset containing data whose classes are known, Machine Learning algorithms can be employed for the induction of a classifier able to predict the class of new data from the same domain, performing the desired discrimination. Several machine learning techniques are originally conceived for the solution of problems with only two classes. In multi-class applications, an alternative frequently employed is to divide the original problem into binary subtasks, whose results are then combined. The decomposition can be generally represented by a code-matrix, where each row corresponds to a codeword assigned for one class and the columns represent the binary classifiers employed. This chapter presents a survey on techniques for multi-class problems code-matrix design. It also shows how evolutionary techniques can be employed to solve this problem.

4.3 Fuzzy logic methods

The role of **Fuzzy Sets** in Data Mining is introduced in Chapter 8. This chapter discusses how fuzzy logic extends the envelop of the main data mining tasks: clustering, classification, regression and association rules. The chapter begins by presenting a formulation of the data mining using fuzzy logic attributes. Then, for each task, the chapter provides a survey of the main algorithms and a detailed description (i.e. pseudo-code) of the most popular algorithms.

Continuing with the same area Chapter 9 addresses **Support Vector Machines and Fuzzy Systems.** Fuzzy set theory and fuzzy logic provide tools for handling uncertainties in data mining tasks. To design a fuzzy rule-based classification system (fuzzy classifier) with good generalization ability in a high dimensional feature space has been an active research topic for a long time. As a powerful machine learning approach for data mining and pattern recognition problems, support vector machine (SVM) is known to have good generalization ability. More importantly, an SVM can work very well on a high (or even infinite) dimensional feature space. This chapter presents a survey of the connection between fuzzy classifiers and kernel machines.

KDD in Marketing with **Genetic Fuzzy Systems** is addressed in Chapter 10. This chapter presents a new methodology to marketing (causal) modeling. Specifically it is applied to a consumer behavior model used for the experimentation. The characteristics of the problem (with uncertain data and available knowledge from a marketing expert) and the multi objective optimization make genetic fuzzy systems a good tool for this problem type. By applying this methodology useful information patterns (fuzzy rules) are obtained, which help to better understand the relations among the elements of the marketing system being analyzed (consumer model in this case).

In Chapter 11 the fuzzy theme is continued with a **Framework for Modeling with Words.** The learning of transparent models is an important and neglected area of data mining. The data mining community has tended to focus on algorithm accuracy with little emphasis on the knowledge representation framework. However, the transparency of a model will help practitioners greatly in understanding the trends and idea hidden behind the system. In this chapter a random set based knowledge representation framework for learning linguistic models is introduced. This framework is referred to as label semantics and a number of data mining algorithms are proposed. In this framework, a vague concept is modeled by a probability distribution over a set of appropriate fuzzy labels, which is called as mass assignment. The idea of mass assignment provides a probabilistic approach for modeling uncertainty based on pre-defined fuzzy labels.

4.4 Advanced soft computing methods and areas

A new soft computing methodology is introduced in Chapter 12, which addresses **Swarm Intelligence** algorithms for data clustering. Data mining tasks require fast and accurate partitioning of huge datasets, which may come with a variety of attributes or features. This, in turn, imposes severe computational requirements on the relevant clustering techniques. A family of bio-inspired algorithms, well-known as Swarm Intelligence (SI) has recently emerged that meets these requirements and has successfully been applied to a number of real world clustering problems. This chapter explores the role of SI in clustering different kinds of datasets. It finally describes a new SI technique for partitioning any dataset into an optimal number of groups through one run of optimization. Computer simulations undertaken in this research have also been provided to demonstrate the effectiveness of the proposed algorithm.

In Chapter 13 another type of method for soft computing is revealed, namely **Diffusion method**. This chapter describes a natural framework based on diffusion processes for the multi-scale analysis of high-dimensional data-sets. Many fields of research deal with high-dimensional data sets. Hyper spectral images in remote sensing and in hyper-spectral microscopy, transactions in banking monitoring systems are just a few examples for this type of sets. Revealing the geometric structure of these data-sets is a preliminary step to facilitate their efficient processing. Often, only a small number of parameters govern the structure of the data-set. This number is the true dimension of the data-set and is the motivation to reduce the dimensionality of the set. Dimensionality reduction algorithms try to discover the true dimension of a data set. The diffusion process scheme enables the description of the geometric structures of such sets by utilizing the Newtonian paradigm according to which a global description of a system can be derived by the aggregation of local transitions. Specifically, a Markov process is used to describe a random walk on the data set. The spectral properties of the Markov matrix that is associated with this process are used to embed the data-set in a low-dimensional space. This scheme also facilitates the parameterization of a data-set when the high dimensional data-set is not accessible and only a pair-wise similarity matrix is at hand.

Agent Technology as applied to Data Mining is introduced in Chapter 14. Today's applications are required to extract knowledge from large, often distributed, repositories of text, multimedia or hybrid content. The nature of this quest makes it impossible to use traditional deterministic computing techniques. Instead, various soft computing techniques are employed to meet the challenge for more sophisticated solutions in knowledge discovery. Most notably, Data Mining (DM) is thought of as one of the state-of-the-art paradigms. DM produces useful patterns and associations from large data repositories that can later be used as *knowledge nuggets*, within the context of any application. Individual facets of knowledge discovery, introduced by DM techniques, often need to be orchestrated, integrated and presented to end users in a unified way. Moreover, knowledge has to be exploited and embodied in autonomous software for learning purposes and, hence, a more increased performance. Agent Technology (AT) proves to be a promising paradigm that is suitable for modeling and implementing the unification of DM tasks, as well as for providing autonomous entity models that dynamically incorporate and use existing knowledge. Indeed, a plethora of multi-agent systems (MAS) and other agent-related solutions for knowledge-based systems can be found in the literature, and more specifically in the area of agent-based DM, as it is explained in detail in this chapter.

The issue of error-tolerant item-set is presented in Chapter 15, which addresses **Approximate Frequent Item-set Mining** in the presence of random noise. Frequent item-set mining has been a focused theme in data mining research and an important first step in the analysis of data arising in a broad range of applications. The traditional exact model for frequent item-set requires that every item occur in each supporting transaction. However, real application data is usually subject to random noise or measurement error, which poses new challenges for the efficient discovery of frequent item-set from the noisy data. Mining approximate frequent item-set in the presence of noise involves two key issues: the definition of a noise-tolerant mining model and the design of an efficient mining algorithm. This chapter gives an overview of the approximate item-set mining algorithms in the presence of random noise and examines several noise-tolerant mining approaches.

The impact of over fitting and over generalization on the classification accuracy in Data Mining is addressed in, Chapter 16, the last chapter of the book. Many classification studies often times conclude with a summary table, which presents performance results of applying various data mining approaches on different datasets. No single method outperforms all methods all the time. Further-more, the performance of a classification method in terms of its false-positive and false-negative rates may be totally unpredictable. Attempts to minimize any of the previous two rates, may lead to an increase on the other rate. If the model allows for new data to be deemed as unclassifiable when there is not adequate information to classify them, then it is possible for the previous two error rates to be very low. However, at the same time, the rate of having unclassifiable new examples may be very high. The root to the above critical problem is the over fitting and overgeneralization behaviors of a given classification approach when it is processing a particular dataset.

Although the above situation is of fundamental importance to data mining, it has not been studied from a comprehensive point of view. Thus, this chapter analyzes the above issues in depth. It also proposes a new approach called the Homogeneity-Based Algorithm (or HBA) for optimally controlling the previous three error rates. This is done by first formulating an optimization problem. The key development in this chapter is based on a special way for analyzing the space of the training data and then partitioning it according to the data density of different regions of this space. Next, the classification task is pursued based on the previous partitioning of the training space. In this way, the previous three error rates can be controlled in a comprehensive manner. Some preliminary computational results seem to indicate that the proposed approach has a significant potential to fill in a critical gap in current data mining methodologies.

.

Part I

Neural Network Methods

Neural Networks For Data Mining

G. Peter Zhang

Georgia State University,
Department of Managerial Sciences,
gpzhang@gsu.edu

Summary. Neural networks have become standard and important tools for data mining. This chapter provides an overview of neural network models and their applications to data mining tasks. We provide historical development of the field of neural networks and present three important classes of neural models including feedforward multilayer networks, Hopfield networks, and Kohonen's self-organizing maps. Modeling issues and applications of these models for data mining are discussed.

Key words: neural networks, regression, classification, prediction, clustering

1 Introduction

Neural networks or artificial neural networks are an important class of tools for quantitative modeling. They have enjoyed considerable popularity among researchers and practitioners over the last 20 years and have been successfully applied to solve a variety of problems in almost all areas of business, industry, and science (Widrow, Rumelhart & Lehr, 1994). Today, neural networks are treated as a standard data mining tool and used for many data mining tasks such as pattern classification, time series analysis, prediction, and clustering. In fact, most commercial data mining software packages include neural networks as a core module.

Neural networks are computing models for information processing and are particularly useful for identifying the fundamental relationship among a set of variables or patterns in the data. They grew out of research in artificial intelligence; specifically, attempts to mimic the learning of the biological neural networks especially those in human brain which may contain more than 10^{11} highly interconnected neurons. Although the *artificial* neural networks discussed in this chapter are extremely simple abstractions of biological systems and are very limited in size, ability, and power comparing biological neural networks, they do share two very important characteristics: 1) parallel processing of information and 2) learning and generalizing from experience.

The popularity of neural networks is due to their powerful modeling capability for pattern recognition. Several important characteristics of neural networks make them suitable and valuable for data mining. First, as opposed to the traditional model-based methods, neural networks do not require several unrealistic *a priori* assumptions about the underlying data generating process and specific model structures. Rather, the modeling process is highly adaptive and the model is largely determined by the characteristics or patterns the network learned from data in the learning process. This data-driven approach is ideal for real world data mining problems where data are plentiful but the meaningful patterns or underlying data structure are yet to be discovered and impossible to be pre-specified.

Second, the mathematical property of the neural network in accurately approximating or representing various complex relationships has been well established and supported by theoretic work (Chen and Chen, 1995; Cybenko, 1989; Hornik, Stinchcombe, and White 1989). This universal approximation capability is powerful because it suggests that neural networks are more general and flexible in modeling the underlying data generating process than traditional fixed-form modeling approaches. As many data mining tasks such as pattern recognition, classification, and forecasting can be treated as function mapping or approximation problems, accurate identification of the underlying function is undoubtedly critical for uncovering the hidden relationships in the data.

Third, neural networks are nonlinear models. As real world data or relationships are inherently nonlinear, traditional linear tools may suffer from significant biases in data mining. Neural networks with their nonlinear and nonparametric nature are more cable for modeling complex data mining problems.

Finally, neural networks are able to solve problems that have imprecise patterns or data containing incomplete and noisy information with a large number of variables. This fault tolerance feature is appealing to data mining problems because real data are usually dirty and do not follow clear probability structures that typically required by statistical models.

This chapter aims to provide readers an overview of neural networks used for data mining tasks. First, we provide a short review of major historical developments in neural networks. Then several important neural network models are introduced and their applications to data mining problems are discussed.

2 A Brief History

Historically, the field of neural networks is benefited by many researchers in diverse areas such as biology, cognitive science, computer science, mathematics, neuroscience, physics, and psychology. The advancement of the filed, however, is not evolved steadily, but rather through periods of dramatic progress and enthusiasm and periods of skepticism and little progress.

The work of McCulloch and Pitts (1943) is the basis of modern view of neural networks and is often treated as the origin of neural network field. Their research is the first attempt to use mathematical model to describe how a neuron works. The main feature of their neuron model is that a weighted sum of input signals is compared to a threshold to determine the neuron output. They showed that simple neural networks can compute any arithmetic or logical function.

In 1949, Hebb (1949) published his book “The Organization of Behavior.” The main premise of this book is that behavior can be explained by the action of neurons. He proposed one of the first learning laws that postulated a mechanism for learning in biological neurons.

In the 1950s, Rosenblatt and other researchers developed a class of neural networks called the perceptrons which are models of a biological neuron. The perceptron and its associated learning rule (Rosenblatt, 1958) had generated a great deal of interest in neural network research. At about the same time, Widrow and Hoff (1960) developed a new learning algorithm and applied it to their ADALINE (Adaptive Linear Neuron) networks which is very similar to perceptrons but with linear transfer function, instead of hard-limiting function typically used in perceptrons. The Widrow-Hoff learning rule is the basis of today’s popular neural network learning methods. Although both perceptrons and ADALINE networks have achieved only limited success in pattern classification because they can only solve linearly-separable problems, they are still treated as important work in neural networks and an understanding of them provides the basis for understanding more complex networks.

The neural network research was hit by the book “Perceptrons” by Minsky and Papert (1969) who pointed out the limitation of the perceptrons and other related networks in solving a large class of nonlinearly separable problems. In addition, although Minsky and Papert proposed multilayer networks with hidden units to overcome the limitation, they were not able to find a way to train the network and stated that the problem of training may be unsolvable. This work causes much pessimism in neural network research and many researchers have left the filed. This is the reason that during the 1970s, the filed has been essentially dormant with very little research activity.

The renewed interest in neural network started in the 1980s when Hopfield (1982) used statistical mechanics to explain the operations of a certain class of recurrent network and demonstrated that neural networks could be trained as an associative memory. Hopfield networks have been used successfully in solving the Traveling Salesman Problem which is a constrained optimization problem (Hopfield and Tank, 1985). At about the same time, Kohonen (1982) developed a neural network based on self-organization whose key idea is to represent sensory signals as two-dimensional images or maps. Kohonen’s networks, often called Kohonen’s feature maps or self-organizing maps, organized neighborhoods of neurons such that similar inputs into the model are topologically close. Because of the usefulness of these two types of networks in solving real problems, more research was devoted to neural networks.

The most important development in the field was doubtlessly the invention of efficient training algorithms—called backpropagation—for multilayer perceptrons which have long been suspected to be capable of overcoming the linear separability limitation of the simple perceptron but have not been used due to lack of good training algorithms. The backpropagation algorithm, originated from Widrow and Hoff's learning rule, formalized by Werbos (1974), developed by Parker (1985), Rumelhart Hinton, and Williams (Rumelhart Hinton & Williams, 1986) and others, and popularized by Rumelhart, et al. (1986), is a systematic method for training multilayer neural networks. As a result of this algorithm, multilayer perceptrons are able to solve many important practical problems, which is the major reason that reinvigorated the filed of neural networks. It is by far the most popular learning paradigm in neural networks applications.

Since then and especially in the 1990s, there have been significant research activities devoted to neural networks. In the last 15 years or so, tens of thousands of papers have been published and numerous successful applications have been reported. It will not be surprising to see even greater advancement and success of neural networks in various data mining applications in the future.

3 Neural Network Models

As can be seen from the short historical review of development of the neural network field, many types of neural networks have been proposed. In fact, several dozens of different neural network models are regularly used for a variety of problems. In this section, we focus on three better known and most commonly used neural network models for data mining purposes: the multilayer feedforward network, the Hopfield network, and the Kohonen's map. It is important to point out that there are numerous variants of each of these networks and the discussions below are limited to the basic model formats.

3.1 Feedforward Neural Networks

The multilayer feedforward neural networks, also called multi-layer perceptrons (MLP), are the most widely studied and used neural network model in practice. According to Wong, Bodnovich, and Selvi (1997), about 95% of business applications of neural networks reported in the literature use this type of neural model. Feedforward neural networks are ideally suitable for modeling relationships between a set of predictor or input variables and one or more response or output variables. In other words, they are appropriate for any functional mapping problem where we want to know how a number of input variables affect the output variable(s). Since most prediction and classification tasks can be treated as function mapping problems, the MLP networks are very appealing to data mining. For this reason, we will focus more on feedforward networks and many issues discussed here can be extended to other types of neural networks.

Model Structure

An MLP is a network consisted of a number of highly interconnected simple computing units called neurons, nodes, or cells, which are organized in layers. Each neuron performs simple task of information processing by converting received inputs into processed outputs. Through the linking arcs among these neurons, knowledge can be generated and stored as arc weights regarding the strength of the relationship between different nodes. Although each neuron implements its function slowly and imperfectly, collectively a neural network is able to perform a variety of tasks efficiently and achieve remarkable results.

Figure 1 shows the architecture of a three-layer feedforward neural network that consists of neurons (circles) organized in three layers: input layer, hidden layer, and output layer. The neurons in the input nodes correspond to the independent or predictor variables that are believed to be useful for predicting the dependent variables which correspond to the output neurons. Neurons in the input layer are passive; they do not process information but are simply used to receive the data patterns and then pass them into the neurons into the next layer. Neurons in the hidden layer are connected to both input and output neurons and are key to learning the pattern in the data and mapping the relationship from input variables to the output variable. Although it is possible to have more than one hidden layer in a multilayer networks, most applications use only one layer. With nonlinear transfer functions, hidden neurons can process complex information received from input neurons and then send processed information to output layer for further processing to generate outputs. In feedforward neural networks, the information flow is one directional from the input to hidden then to output layer and there is no feedback from the output.

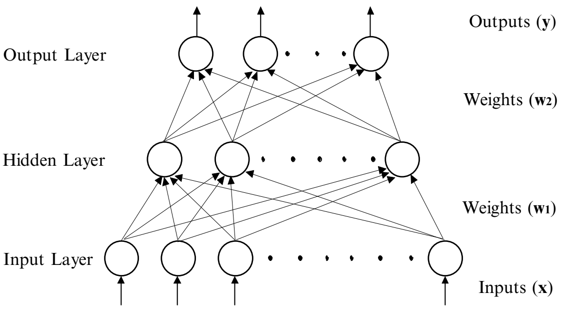

Fig. 1. Multi-layer feedforward neural network