

**DATA MINING TECHNIQUES BASED ON
ROUGH SET THEORY**

SHEN LIXIANG

NATIONAL UNIVERSITY OF SINGAPORE

2001

**DATA MINING TECHNIQUES BASED ON
ROUGH SET THEORY**

**SHEN LIXIANG
(B. ENG., M. ENG.)**

**A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
DEPARTMENT OF MECHANICAL ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE**

2001

Acknowledgement

This work could not have been carried out without both direct and indirect help and many collaborators and friends deserve credit.

First and foremost, it is my great pleasure to express my heartfelt appreciation and gratitude to my supervisor, Dr. Tay Eng Hock, Francis, for his untiring guidance and constant encouragement throughout my candidature. I am indebted to him for his patience and his precious time in amending the drafts of this thesis. I thank Dr. Tay for everything I learned from him.

I would like to thank Mr. Yeong Wai Cheong for giving me the opportunity to learn and practice in Man-Drapeau Research, where I obtained much invaluable practical knowledge on financial activities. Man-Drapeau Research also provides me with all the stock price data used in this research, which facilitates this research to a great extent. I also would like to thank my co-supervisor, Dr. Lawrence Ma, and Mr. Ngan Ngiap Teng for their help and suggestion on this project from the practical point of view.

My colleague, Cao Lijuan discussed with me during the various stages of my research work and offered advice in assisting me to finish the whole project. My best friends Chen Tao and Wang Guohua encouraged me from the beginning of the study to the final accomplishment of my thesis, as well as pursuing a meaningful life. I wish to thank all of you!

Last but not the least, I sincerely thank my parents, whom I owe too much, for their unfailing support throughout my study, without which I cannot further my study in Singapore. I also would like to show my deep gratitude to my friend, Low Leong Kai, for his support throughout this study.

Table of Contents

Acknowledgement	i
Table of Contents	iii
Summary	vii
List of Figures	ix
List of Tables.....	xii
Chapter 1 Introduction	1
1.1 Introduction	1
1.2 Research Objectives	4
1.3 Overview of the Thesis	6
Chapter 2 Background.....	7
2.1 Introduction.....	7
2.2 Knowledge Discovery and Data Mining	7
2.2.1 Goals and themes of KDD	8
2.2.2 The KDD process	10
2.2.3 KDD and the Rough Set Theory.....	12
2.3 Data Mining using Rough Set Theory	14
2.3.1 Typical applications in KDD	14
2.3.2 Economic and financial forecasting using the Rough Set Theory	15
2.3.2.1 business failure prediction.....	16
2.3.2.2 database marketing.....	19
2.3.2.3 financial investment	23
2.4 Significance of Research	27

Chapter 3 Rough Set Basics and Its Application in Fault Diagnosis.....	29
3.1 Introduction.....	29
3.2 Rough Set Basics	29
3.2.1 Information system and decision table	29
3.2.2 Lower and upper approximation.....	30
3.2.3 Quality of approximation.....	32
3.2.4 The discernibility matrices and discernibility function	33
3.2.5 Reduct and core of attributes	34
3.2.6 Decision rules	36
3.3 Different Rough Set Models and Corresponding Software used in Economic and Financial Prediction	38
3.4 Fault Diagnosis using Rough Set Theory	43
3.4.1 The characteristics of vibration signals for a 4153 diesel engine	44
3.4.2 Specification of attributes field.....	49
3.4.3 Discretization of attributes.....	51
3.4.4 Implementation and discussion.....	55
3.5 Summary.....	58
 Chapter 4 Discretization Techniques for Rough Set Theory	60
4.1 Introduction.....	60
4.2 Formulation of Modified Chi2 Algorithm	62
4.3 Experimental Results and Discussion – Benchmark C4.5.....	68
4.4 Experimental Results and Discussion – Benchmark RoughSOM.....	73
4.5 Case Re-Study - The Fault Diagnosis on a 4135 Diesel Engine	75
4.6 Summary.....	78

Chapter 5 RoughSOM System	80
5.1 Introduction.....	80
5.2 The Self-Organizing Map (SOM).....	84
5.2.1 Basic algorithm.....	84
5.2.2 Properties of SOM.....	86
5.3 Case Study	88
5.4 Experiment and Result Analysis.....	91
5.5 Summary.....	97
 Chapter 6 Time Series Forecasting using Rough Set Theory.....	99
6.1 Introduction.....	99
6.2 Temporal Rule Discovery Problem	99
6.3 Temporal Information System (TIS)	100
6.4 Converting Time Series to Rough Set Objects.....	101
6.4.1 The mobile window	101
6.4.2 “columnizing”	103
6.5 Financial Market Prediction	104
6.6 Indicators Study	106
6.6.1 Market trend.....	106
6.6.2 WARS – Weighted Accumulated Reconstruction Series	107
6.6.2.1 formulation of WARS	108
6.6.2.2 comparison of WARS and daily profit curves	112
6.6.2.3 some issues in generating WARS	115
6.6.2.4 using WARS to generate trading system.....	117
6.6.2.5 remarks	119
6.6.3 MACD – Moving Average Convergence/Divergence	120

6.6.4 ROC – Price Rate of Change	121
6.6.5 Stochastic Oscillator	122
6.6.6 RSI - Relative Strength Index.....	123
6.6.7 DI – Directional Indicator.....	124
6.6.8 Linear regression lines.....	125
6.7 Summary.....	127
Chapter 7 Time Series Forecasting Experiments and Discussions	128
7.1 The Process	128
7.2 Data Preparation	129
7.3 Rules Extraction.....	131
7.4 Results Discussion	134
7.5 Summary.....	142
Chapter 8 Conclusions and Recommendations	143
8.1 Introduction.....	143
8.2 Conclusions and Contributions.....	143
8.3 Recommendations for Future Work	145
References	147
Appendix A Proof: The χ^2 value of the reconstructed decision table is greater than that of the original table for a 2-class decision table	168
Publications of the Author.....	171

Summary

Targeting the data mining techniques' study may improve the industry's competence. For our special case, temporal data mining problem such as financial and economic forecasting problem, the study on new approaches will enhance the competence of companies and banks.

The Rough Set Theory is a recently proposed data mining tool with many favourable advantages. Since its publication, this theory has been applied to various domains. The majority of these applications are used to solve the classification problems, which exclude the temporal factor in data sets. In this thesis, the research is focused on study temporal data mining problem. The ultimate objective is to build a trading system based on the financial time series, which is actually a time series forecasting problem. In order to achieve this, following tasks have been accomplished.

- A state-of-the-art review on the financial and economic forecasting based on Rough Set Theory is presented. Through this in-depth review, the readers can get the updated information on what has been done in this area and determine which direction they will push their research forward.
- A real case study – a 4135 diesel engine fault diagnosis using Rough Set Theory – is presented to show its powerful capability to extract useful information from the data.

- A modified Chi2 algorithm is presented as a completely automatic discretization method for Rough Set Theory since the traditional Rough Set Theory cannot handle continuous data. This new algorithm removes the inaccuracy existed in the ChiMerge algorithm and adds a new stopping criterion which makes it be a completely automatic method. Experiments on the machine learning data sets support these modifications.

- Aiming to remove the uncertainty from the system and increase the predictive accuracy on the unseen data sets, the cluster algorithm SOM (Self-Organizing Maps) is applied to discover the “inner relationship” of the data set. This new rough system named RoughSOM improves the capability to classify unseen objects of the Rough Set Theory.

- A temporal rule discovery system has been built. In the final experiments, 4 futures are selected to be studied. By building trading systems on them using above-mentioned research, the Rough Set Theory outperforms the Buy-hold strategy. These results show that the Rough Set Theory is applicable to solve the temporal rule discovery problem, which achieve our research objective.

- A new indicator – WARS (Weighted Accumulated Reconstruction Series) is developed to classify the market states. The performance of the trading system built by this new indicator shows itself to be efficient and easy to operate in the real market.

List of Figures

2.1	KDD process	10
3.1	Rough set approximation	31
3.2	Normal state (sample point 1).....	46
3.3	Normal state (sample point 2).....	46
3.4	Normal state (sample point 3).....	46
3.5	Intake valve clearance is too large (sample point 1)	46
3.6	Intake valve clearance is too large (sample point 2)	46
3.7	Intake valve clearance is too large (sample point 3)	46
3.8	Intake valve clearance is too small (sample point 3)	47
3.9	Intake valve clearance is too small (sample point 3)	47
3.10	Intake valve clearance is too small (sample point 3)	47
3.11	Exhaust valve clearance is too large (sample point 3).....	47
3.12	Exhaust valve clearance is too large (sample point 3).....	47
3.13	Exhaust valve clearance is too large (sample point 3).....	47
4.1	Probability density function of χ^2 distribution (degree of freedom $v=10$).....	65

4.2	Probability density function of χ^2 distribution (degree of freedom $v=7$)	65
4.3	Probability density function of χ^2 distribution (degree of freedom $v=10$).....	65
5.1	Updating the best matching unit (BMU) and its neighbors towards the input sample x . The black and grey circles correspond to situation before and after updating, respectively. The line shows neighborhood relations.....	86
5.2	The flow chart of RoughSOM algorithm.....	91
6.1	Rising trend	107
6.2	Falling trend.....	107
6.3	The illustration of accumulated reconstruction series – original series	109
6.4	The illustration of accumulated reconstruction series – transformed series ...	109
6.5	The comparison of <i>WARS</i> and <i>Daily Profit Curve</i> for S&P 500 Futures.....	114
6.6	The comparison of <i>WARS</i> and <i>Daily Profit Curve</i> for IA Futures.....	114
6.7	The comparison of <i>WARS</i> and <i>Daily Profit Curve</i> for Dax Futures.....	114
6.8	The comparison of <i>WARS</i> and <i>Daily Profit Curve</i> for AO Futures.....	114
6.9	Correlation coefficient of <i>WARS</i> and <i>Daily Profit Curve</i> in different <i>Win_length</i> and <i>Win_len2</i> for AO Futures	116
6.10	Correlation coefficient of <i>WARS</i> and <i>Daily Profit Curve</i> in different <i>Win_length</i> and <i>Win_len2</i> for DAX Futures	116
6.11	Correlation coefficient of <i>WARS</i> and <i>Daily Profit Curve</i> in different <i>Win_length</i> and <i>Win_len2</i> for IA Futures.....	117

6.12	Correlation coefficient of <i>WARS</i> and <i>Daily Profit Curve</i> in different <i>Win_length</i> and <i>Win_len2</i> for S&P 500 Futures.....	117
6.13	Trading system based on <i>WARS</i>	118
6.14	MACD Indicator.....	121
6.15	ROC Indicator.....	121
6.16	Stochastic Oscillator Indicator	123
6.17	RSI Indicator.....	123
6.18	Directional Indicator.....	125
6.19	Linear Regression Indicator	125
7.1	The process of stock market data prediction and analysis.....	128
7.2	S&P 500 index covering period Jan. through Jul. 1999. The <i>Dec_att</i> is shown below the S&P 500 index.....	130
7.3	S&P 500 index covering period Jan. through Jul. 1999. The <i>WARS</i> indicator is shown below the S&P 500 index	132
7.4	Trading system by using threshold=3.0 and by <i>Dec_att</i> covering period Jan.4 to Jul. 26, 1999.....	137

List of Tables

2.1	The main application areas and their corresponding rough set models	16
3.1	A decision table.....	30
3.2	Discernibility matrix of Table 3.1.....	34
3.3	Reducts of Table 3.1.....	35
3.4	The quality of classification for different μ	54
3.5	The quality of approximation of each attribute for different μ	55
3.6	The decision table composed of indices of vibration signals collected from the surface of a 4135 diesel engine.....	56
3.7	Final reducts for different μ	57
4.1	Data sets information	69
4.2	The predictive accuracy (%) using C4.5 with different discretization algorithm.....	71
4.3	The tree size (before / after pruning) comparison of 4 methods.....	72
4.4	Data sets information	74
4.5	The predictive accuracy (%) using RoughSOM with different discretization algorithm.....	74
4.6	The strength of every attribute appeared in the final reducts	76

4.7	The quality of approximation of every attribute.....	77
4.8	The classification accuracy of each part.....	78
5.1	Decision table composed of sorting examples	88
5.2	Decision rules generated from reduct $\{c_1, c_3\}$	89
5.3	Data sets information	93
5.4	The predictive accuracy (%) of original RST and RoughSOM.....	94
5.5	The Interval number after discretization of original decision table and reconstructed decision table	96
5.6	The distribution of data sets.....	97
6.1	Trading system performance based on the indicator <i>WARS</i> on S&P 500 futures	119
6.2	Definitions of the 6 indicators.....	126
7.1	Data sets information.....	131
7.2	Performance benchmark of Buy-hold strategy and original decision attribute for period Jan. 4, 1999 to Jul. 26, 1999 of S&P 500 Index.....	135
7.3	Performance of the trading system for period Jan.4 1999 to Jul.26 1999 of S&P 500 index.....	136
7.4	Performance benchmark of Buy-hold strategy and original decision attribute for period Jan. 4, 1999 to Jul. 6, 1999 of MATIF-CAC Index	138
7.5	Performance of the trading system for period Jan.4 to Jul. 6 1999 of MATIF- CAC Index.....	139

7.6	Performance benchmark of Buy-hold strategy and original decision attribute for period Jan. 4 to Aug. 12, 1999 of EUREX-BUND Index.....	140
7.7	Performance of the trading system for period Jan.4 to Aug.12, 1999 of EUREX-BUND Index	140
7.8	Performance benchmark of Buy-hold strategy and original decision attribute for period Jan. 4 to Jul. 6, 1999 of CBOT-US Index.....	141
7.9	Performance of the trading system for period Jan.4 to Jul. 6, 1999 of CBOT-US Index.....	141
A.1	The original decision table	168
A.2	The reconstructed decision table	165

Chapter 1

Introduction

1.1 Introduction

In the last decade, the amount of data collected and generated in all industries grew at a very fast rate (Brachman et al., 1996). From the financial to manufacturing sector, more and more companies are relying on the analysis of huge amount of data to compete. Although *ad hoc* mixtures of statistical techniques and file management tools once sufficed for analyzing mounds of data, the size of modern data warehouses, the mission-critical nature of the data and the speed with which analyses need to be made now call for a new approach.

A new generation of techniques and tools is emerging to intelligently assist humans in analyzing mountains of data, finding useful knowledge and in some cases performing analysis automatically. These techniques and tools are the subject of the growing field of Data Mining and Knowledge Discovery in Database (KDD). KDD is defined to be the non-trivial process of first identifying valid, novel and potentially useful patterns in data and then understanding these patterns. Data Mining is the process of analyzing data from different perspectives and summarizing them into useful information. It is a step in the KDD process consisting of applying data analysis and discovery algorithms that, under acceptable computational efficiency limitations, produce a particular

enumeration of patterns over the data (Fayyad et al., 1996). For details on KDD and Data Mining, the reader is referred to Derry, 1997; Fayyad et al., 1996; Piatetsky-Shapiro, 1996.

Different data mining methods have different goals. In general, they can be categorized into two types:

- Verification - the system is limited to verifying a user's hypothesis;
- Discovery - the system finds new patterns.

Discovery includes prediction, through which the system finds patterns to help the future behavior of some entities; and description, through which the system finds patterns in order to present the patterns to users in an understandable form. Examples of the key predictive methods include regression and classification, and key description methods include clustering, summarization, visualization and change and deviation detection.

Following these two goals, many techniques have been applied into a variety of domains in data mining, such as marketing, finance, banking, manufacturing and telecommunications. Some results obtained were translated directly into business plans, considerably improving the quality of companies' decisions.

Among these methods, the Rough Set Theory of Pawlak (1982) - as a relatively new knowledge discovery tool has inspired many scholars to carry out research in adapting this theory to various domains. So far they have achieved many promising results

(Polkowski and Skowron, 1998; Shen et al., 2000; Slowinski, 1992; Tsumoto et al., 1996; Weiss, 1996 and 1997; Ziarko, 1993b).

Based on the notion of the existence of indiscernibility relation between objects, the Rough Set Theory deals with the approximation of sets or concepts by means of binary relations. This new method has the following advantages (Dimitras et al., 1999; Greco et al., 1998):

- It is based on the original data only and does not need any external information, unlike probability in statistics or grade of membership in fuzzy set theory (Grzymala-Busse, 1988).
- It is a tool suitable for analyzing not only quantitative attributes but also qualitative ones.
- It discovers important facts hidden in data and expresses them in the natural language of decision rules.
- The set of decision rules derived by the Rough Set Theory gives a generalized description of the knowledge contained in the financial information tables, eliminating any redundancy typical of the original data.
- The decision rules obtained from the Rough Set Theory are based on facts, because each decision rule is supported by a set of real examples.
- The results of the Rough Set Theory are easy to understand, while the results from other methods (credit scoring, utility function and outranking relation) require an interpretation of the technical parameters, with which the user may not be familiar.

This project is a study of the tools used and applications of the Rough Set Theory in data mining. As such, the scope of the current work encompasses an intersection of several scientific areas, such as machine learning, artificial intelligence and financial informatics since the financial data sets are involved in the project. This work focuses on the development of tools and techniques which will be applied in financial data mining.

1.2 Research Objectives

The main objective in the study of data mining techniques is to improve the industrial competency. In this project, the study of new approaches for temporal data mining problems, such as financial and economic forecasting problems, is aimed at enhancing the competence of companies and banks. In order to achieve this, the following tasks will be accomplished:

- **Study discretization methods which can be used to preprocess data for the Rough Set Theory.** Since the traditional Rough Set Theory cannot be applied to knowledge discovery on continuous data sets, the data must be discretized first. There are a number of well-established discretization methods and the selection of a suitable method, especially for financial data discretization, will be studied.
- **Modify the Rough Set Theory to make it more accurate.** In classifying a new object by matching its description to decision rules, one situation is always being overlooked. This is the case whereby the *rule strengths* of both rules leading to different classes are the same and hence there is no clear indication of which class

this object belongs to. This is caused by uncertainty and imprecision contained in the data. Although Rough Set Theory is a powerful tool in dealing with granularity of information and has no requirement of exterior information, it ignores the inner relationships of the data. The determination of the inner relationship and the tool required will be studied.

- **Apply the Rough Set Theory in a temporal rule discovery problem.** In this project, the Rough Set Theory will be applied to discover rules from the time series. As the time series is temporal in nature (i.e. it cannot be solved by the Rough Set Theory), it must be transformed to rough set objects which are in the format of a decision table. The way to implement the transformation and the application of the Rough Set Theory to it will be studied.
- **Develop suitable indicators to compose the decision table.** The decision table is composed of attributes and objects, in which the attributes reflect the system's characteristics and objects represent different situations in this system. During the transformation from the time series to the rough set objects, to implement the technical analysis, indicators must be developed to build up the decision table. The selection and choice of indicators which can reflect the most information contained in data sets will be studied.

In tackling the above problems, the application areas of the Rough Set Theory will be broadened.

1.3 Overview of the Thesis

This thesis contains eight chapters organized as follows. An overview of knowledge discovery and data mining techniques are discussed in Chapter 2. The chapter also includes a review on financial and economic forecasting based on Rough Set Theory. Chapter 3 describes in detail the basics of the Rough Set Theory. A case study for the application of the Rough Set Theory in a 4135 diesel engine fault diagnosis is presented to show its capability to extract useful information from the data. As the conventional Rough Set Theory cannot be applied to a continuous decision table, a study on the discretization problem is undertaken. In Chapter 4, a modified Chi2 algorithm is presented as a completely automatic discretization method for the Rough Set Theory. Aiming to remove the uncertainty from the system and increases the predictive accuracy on the test data sets, the SOM (Self-Organizing Maps) is applied to determine inner relationships contained in data set. This new rough system named RoughSOM is introduced in Chapter 5. In Chapter 6 and 7, the Rough Set Theory is applied to solve the temporal rule discovery problem. The key point to solve this problem is to convert the Temporal Information System (TIS) to an Information System, which can then be processed by conventional Rough Set Theory. Corresponding to this kind of problem, a case study involving the building of trading systems in the financial market is presented, which is essentially a time series forecasting problem. The conclusions and suggestions for future work are presented in Chapter 8.

Throughout the thesis, the Rough Set Theory will be abbreviated as RST.

Chapter 2

Background

2.1 Introduction

This chapter gives a short introduction to the subject matter and the background to this project. First of all, the KDD field is introduced. Several techniques and applications are described and data mining will be identified as the core step within the KDD process. Then, state-of-the-art applications of the RST in data mining are reviewed, with focus on the economic and financial forecasting problem.

2.2 Knowledge Discovery and Data Mining

KDD is commonly defined as “the nontrivial process of identifying valid, novel, potentially useful and ultimately understandable patterns in data” (Fayyad et al., 1996). The term “data” is here a set of facts or atomic pieces of information (e.g. cases in a database) while “knowledge” is a higher-level concept that relates to the properties of the collection of data as a whole (e.g. dependencies among sets of attributes in a database and rules for predicting attribute values (Pawlak, 1991)).

Data mining is a step in the KDD process consisting of applying data analysis and discovery algorithms that, under acceptable computational efficiency limitations, produce a particular enumeration of patterns over the data (Fayyad et al., 1996).

2.2.1 Goals and themes of KDD

The goals are defined by the intentions of the users. The two main goals of KDD are:

- **Prediction:** Using available data to predict unknown or future values giving some variables.
- **Description:** Finding some interesting patterns and presenting them to the user in an easily understood way.

The main distinction between prediction and description is who interprets the discovered knowledge – the system (in case of prediction) or the user (in case of description). However, the boundary between these two goals is not distinct since some predictive models can be used for description and vice versa (Piatetsky-Shapiro, 1996).

KDD covers many fields, such as statistics, database theory and artificial intelligence techniques. Therefore, research themes in KDD are scattered across a range of topics including:

- *Data representation:* Beyond two-dimensional data mining problem, there is a growing interest towards mining free text, multimedia databases and the World Wide Web.

- *Larger Databases:* Databases with hundreds of fields and tables, millions of records, and multi-gigabyte size are quite common nowadays. These require the development of efficient algorithms, sampling and approximation methods to process them.
- *Model pruning and simplification:* A considerable amount of research focuses on developing methods for simplifying models so that they can be understandable. Approaches to this include pruning or filtering an existing model, and transforming one type of model into another more interpretable one.
- *Visualization:* Techniques for visualizing relevant portions of data and knowledge summaries are very useful, and may greatly enhance the steps in the KDD process. Ankerst (2000) provided an overview of this area in his thesis.
- *Quality assessment of data mining results:* After a model has been built, it is necessary to assess its qualities. Methods for knowledge evaluation, benchmarks and metrics for system evaluation and statistical tests in KDD applications are therefore needed.

Other areas in KDD include decomposition of the process, development of parallel steps in the KDD process, development of discretization methods and other pre-processing techniques and ensuring data quality among others.

2.2.2 The KDD process

The overall KDD process is presented in Figure 2.1. It consists of several steps and phases that are interactive and iterative with many decisions being made by the users (Fayyad et al., 1996). From a data source containing raw data, all or portions of the data are selected for further processing. The selected raw data – target data - is then typically pre-processed and transformed in some way, before being passed on to the data mining algorithm itself. The patterns output from the mining procedure are then post-processed, interpreted and evaluated, hopefully revealing new knowledge contained in the data. Along the way, backtracking on each of the steps will in practice inevitably occur in practice.

The selection phase more or less defines the KDD problem. A target data set is created by selecting a data set or by focusing on a subset of data attributes or data samples, on which knowledge discovery is to be performed.

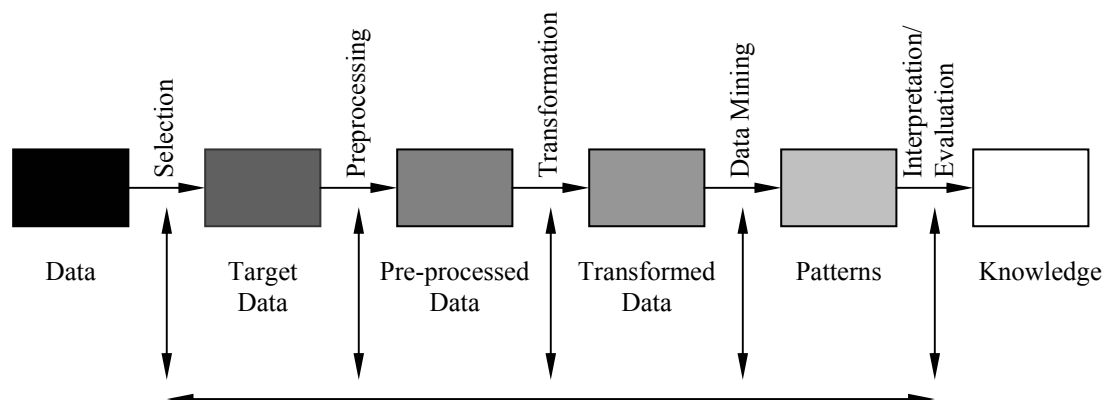


Fig. 2.1 KDD process

In the pre-processing and transformation steps, care should be taken that no unwanted biases are introduced. How these steps are executed depend largely on the method

selected for the subsequent data mining step. Usually, the basic operations in pre-processing step include removing the noise in data, handling missing data field and collecting the time sequence information and known changes.

In the data mining step, many different learning and modelling algorithms are potential candidates. Some of them are symbolically oriented and are trained through logical and algebraic methods while the others have numerical foundations and are trained by regression and curve-fitting methods. The choice of method depends on whether the target to learn is continuous or can be treated as a discrete classification task. In the following, some popular methods are presented.

- *Logistic regression*: It is the most popular analysis technique used in the field of health sciences whereby a set of numerical coefficients is adjusted to obtain a best-fit between a sigmoidal function and the data (Hosmer and Lemeshow, 1989).
- *Neural networks*: Simple nonlinear processing elements are interconnected in a large network in which an input signal is propagated towards one or more designated output nodes (Hertz et al., 1991). Neural networks are useful when a large amount of data has to be modelled and a physical model is not known well enough to use statistical methods. However, with this approach, it is difficult to make a physical interpretation of model parameters. In addition, predicted outputs of the model are limited to the scope of the training set used. As they are not able to discover new relationships in the data, neural nets are not true data mining (Kittler and Wang, 1999).

- *Bayesian networks*: A network of relationships built from a training data set where the weights on the links between nodes are constructed from conditional probability distributions. The networks can be built interactively or by searching a database. Though more physical than neural networks because the nodes in the network are measured variables, it is still difficult to extract elements of a physical model from the network or to effectively visualize the relationships embodied in it.
- *Decision trees*: The input space are recursively partitioned according to various data-driven splitting criteria (Quinlan, 1993). Subsequent predictions can be made by propagating a sample from the root towards the leaves and noting the class distribution of training instances in the destination leaf. Decision trees are useful when relationships are not known and broad categorical classifications or predictions are needed. They are less useful for precise predictions for a continuous variable.
- *Rule-based models*: The if-then rules are extracted which fully or partially describe the example classifications in the training data set. Rules can also be induced via decision trees.

2.2.3 KDD and the Rough Set Theory

The RST is suitable for problems that can be formulated as classification tasks, and has gained significant scientific interest as a framework for KDD (Polkowski and Skowron, 1998).

Adapting the elements of Figure 2.1 to the way models are typically constructed in the framework of RST, the following items can be noted:

- *Selection:* The basic formats for data representation in the rough set framework are two-dimensional data tables, with rows and columns representing the objects and attributes respectively. One suitably formed decision table is selected for subsequent analysis.
- *Pre-processing:* If the selected table contains “holes” in the form of missing values or empty cell entries, the table may be processed in various ways to yield a completed table in which all entries are present (Kryszkiewicz, 1998; Piasta and Lenarcik, 1998; Stefanowski and Tsoukias, 1999).
- *Transformation:* Numerical attributes and attributes that have an ordering on them may have to be discretized, that is, transformed in such a way that intervals or ranges are used instead of the exact observations themselves. This is called the discretization which makes quantitative data more qualitative.
- *Data mining:* In the rough set approach, “if-then” rules are produced. This is implemented in a two-stage process, in which minimal attribute subsets are first computed (which are called reducts) before patterns or rules are generated from these (rule extraction).

- *Interpretation and evaluation:* Individual patterns or rules can be interpreted and evaluated by experts. These rules can also be employed to classify new cases and present their classification performance.

It should be noted that not all of these steps are necessarily carried out, but vary with the application at hand. In some cases the pre-processing and transformation steps may not be needed or desirable to be executed. Furthermore, many of the above steps are not specific to the rough set approach, but are shared by many different learning approaches. For example, most methods operate on flat data tables, and discretization is a step that almost all symbolically based data mining methods require in order to perform well.

2.3 Data Mining using Rough Set Theory

2.3.1 Typical applications in data mining

Industries, which rely heavily on information processing, are the early adopters of data mining techniques. These include the banking, investment, insurance, retail, telecommunications and healthcare industries. To maintain a competitive advantage, these industries have been carrying out research in data mining for several years. The business applications of data mining are plentiful and in the following, a few typical application areas are introduced (Brachman et al., 1996).

Marketing: In marketing, the primary application is database marketing systems, which analyze customer databases to identify different customer groups and forecast

their behaviour. Another notable marketing application is a market-basket analysis system, which finds patterns such as “If a customer bought X, he/she is also likely to buy Y and Z.” Such patterns are valuable to retailers.

Financial Investment: Numerous companies use data mining for investment, but most of them do not publicize their details. One exception is LBS Capital Management which uses expert systems, neural nets and genetic algorithms to manage portfolios totalling \$600 million; since its start in 1993, the system has outperformed the overall stock market.

Fraud detection: There are several systems developed for fraud detection. For examples, FALCON assessment system from HNC applied neural networks to monitor credit card fraud. The FAIS system from the U.S. Treasury Financial Crimes Enforcement Network is used to identify financial transactions that might indicate money-laundering activity.

Other applications include the controlling and scheduling of technical production processes, network management, data quality evaluations and health care. Several successful applications have been developed in these areas (Brachman et al., 1996).

2.3.2 Economic and financial forecasting using the Rough Set Theory

The RST was proposed as a knowledge discovery tool. Inevitably, it have been applied into the above mentioned typical business areas. In the following, the applications of the RST in business are reviewed with the focus on economic and financial prediction.

The applications of the RST in economic and financial prediction can be divided into three main areas: business failure prediction, database marketing and financial investment. The corresponding references are given in Table 2.1, together with the rough set models used (the details of the different models formulated based on basic RST will be described in Chapter 3). The applications are described here to show the diversity of the problems that the RST can handle. In the following, the details of application of the RST to financial domains are presented.

Table 2.1 The main application areas and their corresponding rough set models

RS Models	Business Failure Prediction	Database Marketing	Financial Investment
RSES			Bazan et al. (1994) Baltzersen (1996)
LERS DataLogic	Szladow and Mills (1993)	Poel (1998) Mills (1993) Mrozek and Skabek (1998)	Ziarko et al. (1993) Golan (1995) Golan and Edwards (1993) Ruggiero (1994a, b, c) Skalkos (1996) Lin and Tremba (2000)
TRANCE		Eiben et al. (1998) Kowalczyk and Slisser (1997) Kowalczyk and Piasta (1998) Kowalczyk (1998a) Poel (1998) Poel and Piasta (1998)	
ProbRough			
Dominance Relation	Greco et al. (1998)		
RoughDas and ProFit	Slowinski and Zopoundis (1994 and 1995) Dimitras et al. (1999) Slowinski et al. (1997) Slowinski et al. (1999)		Susmaga et al. (1997)
Hybrid Model	Ahn et al. (2000) Hashemi et al. (1998)		

2.3.2.1 business failure prediction

Business failure prediction is a scientific field in which many academic and professional personnel are interested. Financial organizations, such as banks, credit

institutes and clients among others need these predictions for evaluating firms in which they have an interest. A large number of methods such as discriminant analysis, logit analysis, probit analysis and recursive partitioning algorithm have been applied to model this problem. Most of these methods have already been investigated in the course of comparative studies. Dimitras et al. (1996) gave a complete review of methods used for the prediction of business failure and of new trends in this area. Although some of these methods led to models with a satisfactory ability to discriminate between healthy and bankrupt firms, they suffered from some limitations, often due to the unrealistic assumption of statistical hypotheses or due to a confusing language of communication with the decision makers (experts in this domain). Compared with these methods, the RST appeared to be an effective tool for the analysis of financial decision tables describing a set of objects (firms) by a set of multi-valued attributes (financial ratios) (Dimitras et al., 1996).

The RST has also been used for the analysis and explanation of financing decisions in a Greek industrial development bank called ETEVA (Slowinski and Zopounidis, 1994 and 1995). The ETEVA bank was interested in investing its capital in the better firms so the risk involved in investing was the primary element in its assessment of a firm. A sample of 39 companies was chosen. With the help of a decision maker (financial manager of ETEVA), the decision table was built with 12 attributes (financial ratios) and 1 decision attribute with 3 categories, which indicated whether the company was “acceptable”, “unacceptable” or “uncertain”. The rules generated from the decision table, on one hand, can be used to reveal the financial policy applied in the selection of viable firms. On the other hand, they can be used to evaluate another sample of firms which seek financing from ETEVA bank for the first time, although there was no

validation test provided in the paper. Slowinski and Zopounidis (1995) illustrated how the RST was a useful tool for discovery of a preferential attitude of the decision maker in multi-attribute sorting problems, especially for the bankruptcy risk evaluation of firms.

The application of the RST in business failure prediction was investigated by Slowinski et al. (1999) and Dimitras et al. (1999). In their works, the RST was tested for its prediction ability and was compared with three other methods, namely, C4.5 inductive algorithm, discriminant analysis and logit analysis. In this case, 40 failed firms and 40 matching healthy firms from 13 industries were selected, meeting the criteria of having been in business for more than 5 years and data availability. All together 12 financial ratios were selected by a decision maker (the credit manager of a large Greek bank) to compose the decision table with 1 decision attribute (0 – fail; 1 – healthy). Slowinski et al. (1999) and Dimitras et al. (1999) used a distance measure based on a valued closeness relation (Slowinski, 1993), VCR, to determine which category a test object belongs to in the case of no rules matching this object. The decision rules generated from reducts selected by the decision maker were applied to the validation data set, which were comprised of the previous three years' data (year - 1, -2 and -3) of the same firms. By comparing the predictive accuracy, the RST was found to be more accurate than the classical discriminant analysis by an average of 6.1% per case using minimal set of reduced rules. It also outperformed the C4.5 inductive algorithm as far as the classification accuracy is concerned. Its superiority over logit analysis was not as distinct as that over discriminant analysis. Szladow and Mills (1993) presented a comparative study of rough set model against multivariable discriminant analysis (MDA) for prediction of corporate bankruptcy from five

financial ratios, namely, working capital, retained earnings, earnings before interest and taxes, market value of equities and sales to total assets volumes. By applying the RST, correct predictions for bankrupt firms were increased from 96.0 for MDA to 97.0 percent for the RST. The above comparison results showed that the prediction model based on the RST has more advantages over classical statistical models, such as discriminant analysis, logit analysis and probit analysis.

The study of the financial characteristics of the acquired firms aims at discriminating the acquired firms from the non-acquired ones is another application area of business failure prediction problems. In Slowinski et al. (1997), a study of 30 acquired firms and a sample of equivalent non-acquired firms was carried out. Patterns which would be able to distinguish between the two classes of firms were created using the RST. The RST used here was the same as that in Dimitras et al. (1999). The comparison of predictive accuracy with the discriminant analysis also showed that the RST was a strong alternative for the prediction of company acquisition.

2.3.2.2 database marketing

Database marketing is a capacious term related to the way of thinking and acting which contains the application of tools and methods in studies and formation of the companies surroundings, their structure and internal organization in order that they could achieve success on a fluctuating and difficult to predict consumer market. In simplicity, database marketing can be defined as a method of analyzing customer data to look for patterns among existing preferences and to use these patterns for more targeted selection of customers (Fayyad et al., 1996). Database marketing is characterized by enormous amounts of data at the level of the individual consumer.

However, these data have to be turned into useful information. To this end, several different problem specifications can be investigated. These include market segmentation, cross-sell prediction, response modeling, customer valuation and market basket analysis (Ananyan, 2000). Building successful solutions for these tasks requires the application of advanced data mining and machine learning techniques to obtain relationships and patterns in historical data and using this knowledge to predict each prospect's reaction to future situations. The RST has also been applied in this domain.

Poel (1998) gave a rough overview on database marketing with the focus on customers' response modeling. The goal was to use past transaction data of customers, personal characteristics and their response behavior to determine whether these clients were good mailing prospects during the next period. The data were real-world data collected from one of the largest European mail-order companies. Poel (1998) applied statistical techniques (discriminant analysis, logistic regression, CART and CHAID), machine learning methods (C4.5), mathematical programming (linear programming classification), RST (LERS and ProbRough) and neural networks to model this customers response problem. The performance of each method was evaluated on the basis of two criteria, that are, the percentage classified correctly in the validation samples and gains chart analysis. (The gains chart analysis is widely used in comparing alternative techniques as shown in Furness (1994). It is in fact an application of Lorenz curve of incremental expenditure to the database marketing setting (Thompson, 1994).) The result of one-fold cross-validation test showed that the rough set model scored second (with a classification accuracy of 74.35%), following the CHAID (with a classification accuracy of 74.62%), and was characterized by only a small drop-off between learning and estimation samples. In the ten-fold cross-

validation test, the RST offered similar level of performance to CART, with a higher mean accuracy and lower standard deviation, in terms of the number of rules generated and in terms of the variables in the rules. From the analysis of gains chart, two rough set model – LERS and ProbRough had distinctly different gains charts compared with other methods. The results generated by LERS, offered good predictive capability at the end of the chart, whereas ProbRough only showed top performance at lower side of the chart. The comparison among these methods revealed that the classical statistical parametric techniques, discriminant analysis and logistic regression, performed very well for the relevant range of the gains chart. The machine learning method, the RST and neural network, which were new to database marketing, can also be used successfully as techniques for response modeling.

Poel and Piasta (1998) further discussed the efficacy of the RST in response modeling for the mailing-order prediction problem. By the analysis of the classification ability of the RST, they drew the conclusion that the results obtained from the application of the RST rediscovered the variables, namely, Recency, Frequency and Monetary value (RFM) as the most significant predictors for mail-order buying behavior. In addition, a significant finding on the importance of non-RFM variables in predicting purchasing behavior as the ratio of misclassification costs became higher, was agreeable with prior beliefs of marketing managers. These results were not revealed by other data mining efforts before. From this point of view, the RST can excavate the inherent important factors contained in the system.

Using the RST to model customer retention is another application area in database marketing (Eiben et al., 1998; Kowalczyk, 1998a; Kowalczyk and Piasta, 1998;

Kowalczyk and Slisser 1997). The retention of its customers is very important to a commercial entity, in particular, a bank. When a client decides to move to another bank, it usually implies some financial losses for this bank. Therefore, banks are very interested in identifying some mechanisms behind such decisions and determining which clients are about to leave the bank. One way to find such potential customers is to analyze the historical data which describe customer behavior in the past. Kowalczyk and Slisser (1997) gave the simple description on how to model the customer retention problem using the RST. It included the following steps: the conceptual analysis of problem, initial analysis of available data, identification of the more important attributes, construction of models and extraction of rules from the data base. By applying these steps in modeling customer retention, it showed that the clients who were investors for a long time, invested money in funds with very small risk, and acquired small profits had terminated their relationship with the company. This result was similar to the experience of an expert. In addition, compared with genetic programming, logistic regression and CHAID, the model based on the RST was a more efficient and simpler approach and it identifies more factors influencing the customers' behavior.

Besides the above two main applications, the RST has been applied to evaluate the qualifications of credit cards applicants (Mrozek and Skabek, 1998; Piasta and Lenarcik, 1996), draft an advertising budget for a company (Mrozek and Skabek, 1998) and predict the likely buyers for a company (Mill, 1993). In their works, the RST was used to analyze data on current customers and patterns were created to describe typical customers. Then any new prospect who fits these patterns would be

identified as a potential customer. The RST is an easy tool to help a manager organize the data and look at it in a different way.

2.3.2.3 financial investment

Many financial analysis applications employ predictive modeling techniques, for example, statistical regression and neural network, to create and optimize portfolios and to build trading systems. These problems belong to the domain of financial investment. In the KDD domain, it is actually a temporal rule discovery problem. For the traditional RST, this problem cannot be solved unless the Temporal Information System (TIS) is converted to Information System. This part of work will be discussed in Chapter 6.

Building trading systems using the RST has been studied by several researchers. Ziarko et al. (1993), Golan and Edwards (1993) and Golan (1995) applied the RST to discover strong trading rules, which reflect highly repetitive patterns in data, from historical database of the Toronto Stock Exchange (TSE). By using 1980's stock and economic data, they extracted the trading rules of five major TSE companies' stocks. These five companies were the Bank of Montreal, the Bell Canada Enterprises, the Imperial Oil, the Loblaws and the Northern Telecom. As indicated by the expert, these rules described the stock behavior of these companies and their sensitivity to market or economic indices, although not all the rules discovered by the RST were of high quality. The "generalized rules" (with a low roughness parameter, this concept will be introduced in Section 4) extracted by the RST were all recognized rules or relationships in the investment industry, while the "exact rules" (with a high roughness parameter up to 1) made less sense to them. Their works proposed a methodology

which established the RST as a good candidate for knowledge discovery in stock market data.

Bazan et al. (1994) discussed the same trading system building problem using the RST. In his work, 15 market indicators were collected and the problem was focused on how to deduce the rules that map the financial indicators at the end of a given month onto the stock price changes a month later. Aimed at reducing the reduct set computation time, new solutions - conceptual clustering, extracting new attributes from decision table and joining groups of attributes - were discussed to meet hardware requirement. The preliminary results using the RST seemed to be only satisfactory with a classification accuracy of 44%. In addition, there were still some problems such as data filtration, incomplete data and evolution learning problem. Baltzersen (1996) also did some research on the Total Index of the Oslo Stock Exchange (OSE) using the RST. His studies included data collection and selection, conversion of time-series to rough set objects, reduct analysis, rules construction and using the RST to forecast the development of the index. Although the classification accuracy was not satisfactory (from 25% to 45% for different discretization methods), he extracted several clear indications for some of the factors whose change had more effect than their level, such as the slope of the interest rate, currency rate and consumption rate of gasoline for motor vehicles were more important than themselves.

Building trading systems on the S&P index using the RST is another major application in financial investment. Skalkos (1996), following the KDD procedures, extracted a set of rules from S&P 100 index, put/call data, NYSE (New York Stock Exchange) trading statistics, and US Treasury bond yield data from Oct. 1987 to Dec. 1994. These

extracted rules were applied to trade using data from Jan. 1995 to Dec. 1995. There were 9 trades initiated by the trading rules, among which 7 were profitable. By analysis of the results, Skalkos proposed that the interest rates and market sentiment played important roles in a short-term trading system. His study was noteworthy for its use of technical analysis as well as financial data.

Ruggiero (1994a) has also done a lot of research on the building trading systems on S&P 500. He developed a set of rules to predict 'short' and 'long' positions in the S&P 500 while recognizing different market price cycles (Ruggiero, 1994a). It was claimed that excellent performance was achieved by using strong rules while discarding weak rules in trading. Over the whole trading period, this system achieved correct calling of over 70 percent of the positive moves in the next 5 weeks and average transaction represented over \$25,000 profit per S&P 500 contract. In Ruggiero (1994b), rules to predict strong rallies in the S&P 500 of 2 percent or more were developed. The trading system extracted the strong generalization rules for predicting S&P 500 'buy' and 'sell' signals. Ruggiero (1994c) also developed a hybrid trading system incorporating neural networks, the RST and a spreadsheet. Neural network models were supervised by rule generated by the RST to correct for possible errors in predictions. The system allowed the user to address and exploit inefficiencies that exist in the market. This trading system obtained the reduction of drawdown by 25 to 50 percent and increased the average winner/loser ratio by 50 to 100 percent. The average trade length was reduced by 50 to 80 percent.

The RST was also applied into the domains of portfolio management. Greco et al. (1996) did some research on stock selection problem of the Italian Stock Market based

on the RST. 22 Italian “Blue Chips” which has been listed for a long period were chosen to compose an equally weighted portfolio. Seven factors being proxies of risk exposures were selected to compose the decision table. This portfolio was studied using the RST and two conventional approaches of MFM (Multi-Factor Models). The results show that the RST outperforms the MFM in the case of small samples and during turbulent periods. In addition, the RST provides a specific integration of classical market description for the investment process. It is a promising tool in portfolio risk management with focus on behavioral finance.

Portfolio tilting was another application. The notion of “tilting” in portfolio management was to define a systematic approach to construct a portfolio which had a higher (or lower) value of a particular attribute than the value which can be found in a benchmark portfolio (such as mean-variance framework). For example, an investor preferred a low dividend paying stock over high dividend paying stock (all else being equal). This implied the portfolio was tilted in such a way that either it would include a higher number of low dividend paying stocks or would have a higher investment in the few low dividend paying stocks. Susmaga et al. (1997) studied the portfolio tilting problem according to some well-established price-related stock attributes. The data set used included all the publicly traded companies listed on the Toronto Stock Exchange. By using the n-fold cross validation test, it was concluded that for potential investors the price-related attributes were more significant in designing successful investment strategies. The predictive quality on stock top performers group in the next year reached up to 70%, using the rules generated only from “Common Reduct”. This indicated that the rough set analysis was useful in determining the contributions of

attributes to identify top stock performers, allowing for a construction of the decision rules which might be applied to the evaluation of the new stocks.

2.4 Significance of Research

It can be seen that the RST is a promising alternative to the conventional methods through the above review.

In this project, the RST will be applied to the temporal rule discovery problem. Temporal rule discovery problems involve the applications of knowledge discovery techniques to identify temporal rules from time related series (Golan and Edwards, 1993). Here the temporal rules refer to the discovered relationships which reflect common repetitive patterns occurring in time series. As the temporal factor existing in this kind of data cannot be processed by the RST directly, they must be transformed to rough set objects first. In the three main application areas reviewed in the previous section, the majority of financial investment applications, such as building trading systems on S&P index, belongs to the temporal rule discovery problem. The different rough set models and transferring methods are applied to find trading rules. Some promising results have been generated. As the application of the RST to extract rules from time series is more challenging than from ordinary data sets, the current research is focused on this area. In this project, a special case in this area, that is, building a system for stock or futures trading, is studied. Several problems such as discretization, indicator selection and system uncertainty removal will be discussed.

The methods proposed in this project can be applied to address similar problems in other areas, where activities like the conversion of the Temporal Information System (TIS) to Information System, composition of the decision table and discretization are necessary. This research will shed some light on how to make the RST more powerful and practical for addressing various real-world problems.

Chapter 3

Rough Set Basics and Its Application in Fault Diagnosis

3.1 Introduction

The RST (Pawlak, 1982) was proposed as a new mathematical tool to handle vagueness and uncertainty inherent in making decisions. This theory finds applications primarily in some branches of artificial intelligence and cognitive sciences, such as machine learning, expert systems, inductive reasoning, automatic classification, pattern recognition and learning, computer security analysis. Some representative applications can be found in the following references: Lin et al. (1996), Lin and Cercone (2000), Nowicki et al. (1992), Slowinski (1992), Slowinski et al. (1995), Tsumoto and Tanaka (1996) and Ziarko et al. (1993). For completeness and clarification, the fundamentals of RST is described in the following sections.

3.2 Rough Set Basics

3.2.1 Information system and decision table

Information Systems are used to represent knowledge. The notion of an Information System presented here is described in Pawlak (1991).

An Information System $S=(U, \Omega, V_q, f_q)$ consists of:

U – a nonempty, finite set called the *universe*,

Ω – a nonempty, finite set of *attributes*; $\Omega = C \cup D$, in which C is a finite set of *condition attributes* and D is a finite set of *decision attributes*,

for each $q \in \Omega$, V_q is called the *domain of q* ,

f_q – an *information function* $f_q: U \rightarrow V_q$.

There are various possible interpretations of *objects* in practical applications, for example, cases, states, processes, patients and observations. *Attributes* can be interpreted as features, variables and characteristic conditions. A special case of Information Systems called *decision table* or *attribute-value table*. In a decision table, the columns are labeled by attributes, and rows are by objects (states, processes, events, etc.). Example of a decision table (attribute-value table) is given in Table 3.1.

Table 3.1 A decision table

U Φ	A	b	c	d	E
1	1	0	2	1	0
2	0	0	1	2	1
3	2	0	2	1	0
4	0	0	2	2	2
5	1	2	2	1	0

In Table 3.1, $U=\{1, 2, 3, 4, 5\}$ and $\Omega=\{a, b, c, d, e\}$.

3.2.2 Lower and upper approximation

In an Information System, to every subset of attributes $B \subseteq \Omega$, a binary relation, denoted by $IND_{\Omega}(B)$ or $IND(B)$, called the *B-indiscernibility relation*, is associated and defined as follows:

$$IND(B) = \left\{ (x, y) \in U^2 : \text{for every } a \in B, a(x) = a(y) \right\} \quad (3.1)$$

where $IND(B)$ is an equivalence relation and $IND(B) = \bigcap_{a \in B} IND(a)$

Objects x and y satisfying relation $IND(B)$ are indiscernible by attributes from B .

Consider the subset $B = \{a, b, c\}$ in Table 3.1, then

$$U/IND(\{a\}) = \{\{1, 5\}, \{2, 4\}, \{3\}\}; \quad U/IND(B) = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}\}$$

Sets that are unions of some classes of the indiscernibly relation $IND(B)$ are called definable by B .

If $S = (U, \Omega, V_q, f_q)$ is a decision table, $B \subseteq \Omega$ and $X \subseteq U$ then B -lower and B -upper approximation of X is defined respectively as follows:

$$\underline{B}X = \bigcup \{Y \in U / IND(B) : Y \subseteq X\} \quad (3.2)$$

$$\overline{B}X = \bigcup \{Y \in U / IND(B) : Y \cap X \neq \emptyset\} \quad (3.3)$$

where $U/IND(B)$ denotes the family of all equivalence classes of B (classification of U).

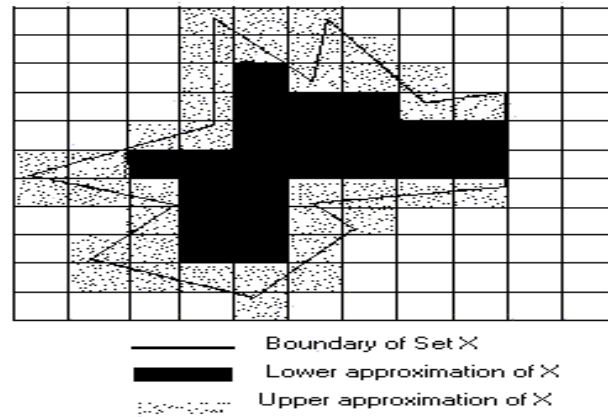


Fig. 3.1 Rough set approximation

The set $\underline{B}X$ is the set of all elements of U , which can be certainly classified as elements of X , with respect to the values of attributes from B ; and the set $\overline{B}X$ is those elements of U which can be possibly defined as elements of X with respect to the value of the attributes from B . Finally, $BN_B(X)$ is the set of elements which can be classified neither in X nor in \overline{X} on the basis of the values of attributes from B . The set $BN_B(X) = \overline{B}X - \underline{B}X$ is called the B -boundary of X . Figure 3.1 indicates these areas of (un)certainity.

For the decision table of Table 3.1, the subset of attribute $B = \{a, b\}$ and the subset of objects $X = \{2, 3\}$, the following approximations can be derived:

$$\underline{B}X = \{3\}; \quad \overline{B}X = \{2, 3, 4\}; \quad BN_B(X) = \{2, 4\};$$

3.2.3 Quality of approximation

Inexactness of a set (category) is due to the existence of a boundary region. The following *quality of lower approximation* of X by B on Ω were introduced in Grzymala-Busse (1991):

$$\underline{\gamma}_B(X) = \frac{|\underline{B}X|}{|U|} \quad \text{and} \quad \overline{\gamma}_B(X) = \frac{|\overline{B}X|}{|U|} \quad (3.4)$$

Thus, the *quality of lower approximation* of X by B in Ω is the ratio of the number of all certainly classified objects by attributes from B as being in X to the number of all objects in the system. $\overline{\gamma}_B(X)$ is intended to capture the degree of completeness of knowledge about the set X . It is a kind of relative frequency. The *quality of upper*

approximation of X by B in Ω is the ratio of the number of all possible classified objects by attributes from B as being in X to the number of all objects in the system. It is also a kind of relative frequency.

Two measures to describe inexactness of approximation classifications have been defined; the first measure is the *accuracy of approximation* of Ω by B . It expresses the possible correct decisions when classifying objects employing the attribute B .

$$\alpha_B(\Omega) = \frac{\sum \text{card}(\underline{BX}_i)}{\sum \text{card}(\overline{BX}_i)} \quad (3.5)$$

The second measure is called the *quality of approximation* of Ω by B . It expresses the percentage of objects, which can be correctly classified into class Ω employing the attribute B :

$$\gamma_B(\Omega) = \frac{\sum \text{card}(\underline{BX}_i)}{\text{card}(U)} \quad (3.6)$$

3.2.4 The discernibility matrices and discernibility function

The elements of a discernibility matrix (Skowron and Rauszer, 1992) of B is defined as follows:

$$\left(c_{ij} \right) = \left\{ a \in B : a(x_i) \neq a(x_j) \right\} \text{ for } i, j = 1, 2, \dots, n \quad (3.7)$$

and a discernibility function $f(B)$ is a Boolean function of n attributes defined as:

$$f(B) = \prod_{(x,y) \in U^2} \sum \delta(x,y) = \bigcap \{ \bigcup (c_{ij}) : 1 \leq j \leq i \leq n \cdot n; c_{ij} \neq 0 \} \quad (3.8)$$

where Π denotes products of partitions.

For the decision table as shown in Table 3.1, the discernibility matrix is listed in Table 3.2.

Table 3.2 Discernibility matrix of Table 3.1

U	1	2	3	4	5
1					
2	a, c, d, e				
3	a	a, c, d, e			
4	a, d, e	c, e	a, d, e		
5	b	a, b, c, d, e	a, b	a, b, d, e	

The discernibility function for this set is:

$$f(a,b,c,d,e) = (a \vee c \vee d \vee e)a(a \vee c \vee d \vee e)(a \vee d \vee e)(c \vee e)(a \vee d \vee e)b \\ (a \vee b \vee c \vee d \vee e)(a \vee b)(a \vee b \vee d \vee e) = ab(c \vee e)$$

The discernibility matrix and the discernibility function are used to extract the minimal reducts.

3.2.5 Reduct and core of attributes

There are two fundamental concepts in connection with the knowledge reduction. The B-reduct of A is the minimal subset of A, which provides the same classification of objects into elementary classes of B as the whole attributes A.

The B-core of A is the essential part of A, which cannot be eliminated without disturbing the ability to classify objects into elementary classes of B.

The attribute $a \in B$ is regarded as superflous in B if $IND(B) = IND(B - \{a\})$; otherwise the attribute a is indispensable in B . If all attributes $a \in B$ are indispensable in B , then B will be called orthogonal.

Subset $B' \subseteq B$ is a reduct of B , iff B' is orthogonal and $IND(B) = IND(B')$. The set of all indispensable attributes in B will be called the core of B and will be denoted $CORE(B)$.

$$CORE(B) = \bigcap_{R \in RED(B)} R \quad (3.9)$$

where $RED(B)$ is the family of all reduct of B .

The core is defined as the set of all single element entris of the discernibility matrix $M(B)$, i.e.

$$CORE = \{a \in B : c_{ij} = (a), \text{ for some } i, j\} \quad (3.10)$$

A set $B' \subseteq B$ is the reduct of B if B' is the minimal subset of B such that $B' \cap c \neq \emptyset$ for every nonempty entry $c (c \neq \emptyset)$ in $M(B)$. In other words reduct is the minimal subset of attributes that discerns all objects discernible by the whole set of attributes. For the decision table of Table 3.1, the core of the set $\{a, b, c, d, e\}$ is the set $\{a, b\}$ and two reducts are $\{a, b, c\}$ and $\{a, b, e\}$, shown in Table 3.3.

Table 3.3

U	a	b	c
1	1	0	2
2	0	0	1
3	2	0	2
4	0	0	2
5	1	2	2

(a)

Reducts of Table 3.1

U	a	b	e
1	1	0	0
2	0	0	1
3	2	0	0
4	0	0	2
5	1	2	0

(b)

3.2.6 Decision rules

Problems of inducing decision rules have been extensively investigated in many fields, particularly in the machine learning domain (Michalski, 1983; Shavlik and Dietterich, 1990; Weiss and Kulikowski, 1990). The RST can also be applied to different stages of rule induction and data processing. However, one aspect that distinguishes the RST from typical machine learning systems is that the RST does not correct or aggregate the inconsistency in the input data (Grzymala-Busse, 1988). The lower and upper approximation are applied to describe the inconsistency and consequently, certain and approximate rules are induced.

Procedures for derivation of decision rules from decision table were presented by Grzymala-Busse (1992), Skowron (1993), Slowinski and Stefanowski (1992), Stefanowski and Vanderpooten (1994) and Ziarko et al. (1993). More advanced rule induction methods has been studied in Bazan (1998) for comparing the dynamic and non-dynamic methods of induction rules from decision tables, Grzymala-Busse and Zou (1998) and Stefanowski (1998b) carried out work in this area with the focus on induction rules from inconsistent decision table. Lin (1996) and Lin and Yao (1996) studied the rule induction from very large databases combined with database technologies.

A decision rule can be expressed as a logical statement:

IF conjunction of elementary conditions

THEN disjunction of elementary decisions

Decision rules induced from a decision table can be applied to classify new objects. Specifically, the classification of a new object can be supported by matching its description to one of the decision rules. The matching may lead to one of the four situations:

- (i) The new object matches exactly one of the deterministic decision rules;
- (ii) The new object matches exactly one of the non-deterministic decision rules;
- (iii) The new object does not match any of the decision rules;
- (iv) The new object matches more than one rule.

In (i), the classification suggestion is obvious. In (ii), however, the suggestion is not direct since the matched rule is ambiguous. In this case, the Decision Maker (DM) is informed of the number of training objects that support each possible class. The number is called the *rule strength*.

Situation (iii) is more difficult to solve. In this case, a set of rules “nearest” to the description of the new object will be presented to the DM. The notion of “nearest” involves the use of the distance measure, such as a *valued closeness relation* R proposed by Slowinski and Stefanowski (1994).

Situation (iv) may also be ambiguous if the matched rules (deterministic or not) lead to different classes. Here, the suggestion can be based either on the *rule strength* of possible classes, or on an analysis of the training objects that support each possible class. In the latter case, the suggested class is that which is supported by a classification problem closest to the new object based on the relation R .

In the solutions to situation (ii) and (iv) mentioned above, one situation is missing, that is the situation whereby the *rule strengths* of both classes are the same in which case there is no clear indication on how to classify the new object. In order to tackle this problem, Tay and Shen (2001) applied the SOM (Self-Organizing Map) to extract the inner relationship inherent in data sets. This inner relationship is helpful in distinguishing the “strong” objects from the “weak” objects. By “strong object”, one means that the inner category for the object as determined by SOM is the same as the original value of decision attribute. Otherwise, the object is termed a “weak object”. The information inherent in the data sets helps to remove the uncertainty from the system and increase the classification accuracy on the new objects. This is especially efficient for inconsistent systems. The details will be given in Chapter 5.

3.3 Different Rough Set Models and Corresponding Software used in Economic and Financial Prediction

Since the RST was proposed in 1982, it has been studied and modified by many researchers. Consequently, different rough set models were advanced to widen the applications of the RST in economic and financial prediction. These models and their corresponding applications were given in Table 2.1.

LEERS (Learning from Examples using Rough Set) is a rule induction system developed by Grzymala-Busse (1992 and 1998). There are two different approaches for rule induction in this system, which are computing sufficient rule set using a machine learning approach and computing all rules by a knowledge acquisition approach. In both approaches the user has an additional choice between local and

global algorithms. Among these options, LEM2 (Learning from Examples Module, version 2), which is a local algorithm using the machine learning approach, is most widely used in practice. Poel (1998) applied LERS into database marketing with the focus on customers' response modelling and presented promising results on this new alternative approach. Stefanowski (1998a) proposed a modified LEM2 algorithm to handle directly continuous attributes and discretize them inside the learning algorithm while creating elementary conditions. This algorithm extracts better sets of decision rules than LEM2 so as to enhance the predictive accuracy of rough set based rule induction system.

Bazan et al. (1994) and Baltzersen (1996) engaged in market data analysis with the aid of the RSES (Rough Set Expert System) (Bazan and Szczuka, 2000). The rule induction system of this software is based on Boolean Reasoning and dynamic reducts (Bazan, 1998). Besides all the basic operations of the RST, this software also provides the discretization algorithms and template generation algorithms.

The Variable Precision Rough Set Model (VPRSM) was proposed by Ziarko (1993) as a derivative of the basic RST. This model broadened the deterministic data dependencies, which was the foundation of basic RST, to non-deterministic relationships. Given an acceptable rule probability β , the strong non-deterministic rules were extracted. These rules were likely to be correct or almost correct but their usefulness depends on β . In Ziarko et al. (1993), the VPRSM (Ziarko, 1993a), which has been developed into a commercial software – DataLogic – was applied to extract the trading rules with β set to 0.55. The same rough set model was applied to build the

trading system in Lin and Tremba (2000), Skalkos (1996), Golan (1995), Golan and Edwards (1993) and Ruggiero (1994a and 1994b).

Rough Classifier, developed by Lenarcik and Piasta (1994) and Rough Data Models introduced by Kowalczyk (1998b) were two approaches that avoided the use of reducts. Both approaches were focused on finding a relatively simple partition of the attribute space and then drawing some conclusions from the structure of this partition. Two systems that were representative of these approaches were ProbRough (Piasta and Lenarcik, 1996 and 1998; Lenarcik and Piasta, 1998) and TRANCE (Kowalczyk, 1996 and 1998b). These two systems were mainly used in database marketing, such as customer retention modeling (Eiben et al., 1998; Kowalczyk and Slisser, 1997; Kowalczyk and Piasta, 1998; Kowalczyk, 1998a), purchase prediction (Poel and Piasta, 1998), respond modeling (Poel, 1998) and bankruptcy prediction (Piasta and Lenarcik, 1996).

The RST was founded on the assumption that every object in the universe was associated with some information. Objects characterized by the same description were indiscernible in view of the available information about them, which has been defined in the second section of this chapter as indiscernibility relation. Greco et al. (1998), taking into account the ordinal properties of the considered evaluation criteria, proposed the dominance relation instead of the discernibility relation to reconstruct the rough set model. This new model not only maintained the best properties of the basic RST but also presented more understandable rules to the user. In addition, the rules based on dominance relation were also better adapted to sort new actions than the rules on indiscernibility. This new model was applied to bankruptcy risk evaluation

(Slowinski and Zopounidis, 1995). Compared with previous results, the new model presented a smaller number of reducts (only 4 for dominance relation against 26 for indiscernibility relation) and a larger core ($\{\text{Attribute 7, Attribute 9}\}$ against $\{\text{Attribute 7}\}$). These two features were generally recognized as desirable properties of a good approximation (Pawlak, 1991; Slowinski and Stefanowski, 1996). Moreover, the decision rules from dominance relation generally perform better when applied to new objects.

Lin (1988 and 1989) proposed a more general framework for the study of approximation. He adopted neighborhood systems from topological spaces to describe relationship between objects in RST. This neighborhood approximation is also applicable to the situation where a new object cannot match any rule.

The Valued Closeness Relation (VCR) (Slowinski, 1993; Slowinski and Stefanowski, 1994) was proposed to solve the problem of no rules matching the new object during the prediction phase. It involved indifference, strict difference and veto thresholds on particular attributes, used in concordance and discordance test. The goals of these tests were to:

- (i) characterize a group of attributes considered to be in accordance with the affirmation “object x is close to rule y ”, and assess the relative importance of this group,
- (ii) characterize among the attributes, which are not in concordance with the above affirmation, the ones whose opposition is strong enough to reduce the credibility of the closeness, which would result from taking into account just concordance, and to calculate the possible reduction that would thereby result.

Dimitras et al. (1999) compared the classification results before and after the applications of VCR. They indicated that on the average, 60% of objects not classified by exactly matching rules were classified correctly by the VCR-nearest rules, which was a better result than random classification of these objects. In addition, using VCR also supported that two objects were marginal, which had not been detected before the application of VCR. This distance measure has also been used in portfolio tilting (Susmaga et al, 1997). Softwares supporting this technique include RoughDas and ProFit (Mienko et al., 1996). Now a new extension of this software called ROSE (Predki et al. 1998; Slowinski and Stefanowski, 1998) which provides different techniques on rule induction is available. In addition, besides all the basic operations of the RST, ROSE provides several approximation techniques to avoid data discretization, such as similarity relation and dominance relation. These techniques can be easily controlled by the users.

Neighborhood systems is geometric version of binary relation (Lin, 1989). So it is a generalization of RS; in this paper if a new instance that are not matched by known rules are discussed; it uses binary relation to capture the concept of near-ness.

In recent research works (Yasdi, 1995; Hashemi et al. 1998; Ahn et al. 2000), the RST combined with neural network has been used in economic and financial prediction. In these hybrid models, the RST took the role of preprocessor for the neural network by reducing the decision table. This is very useful for neural network in that reduction of attributes prevents overfitting and saves training time. Furthermore, removing conflicting objects and training neural network with consistent cases can improve the performance as well as reduce the training time. Ahn et al. (2000) applied this hybrid

model to predict the business failure for over 1200 healthy firms and 1200 failed firms in Korea. The results showed this hybrid model outperformed discriminant analysis model and neural network model.

3.4 Fault Diagnosis using Rough Set Theory

The frequent occurrence of faults in manufacturing system and machinery drives up the cost of production, adversely affects part quality, reduces the production rate, and limits the flexibility of the system. A reduction in the frequency of fault occurrence can be achieved through investment in better quality machines, operators and controllers among others. Reducing cycle time can also help to reduce some types of failure rates (mainly those affected by machine wear). As the flexibility of a manufacturing system increases, so does the number of possible failures. Consequently, it is increasingly important to understand how faults occur in a given manufacturing system so as to determine, in a timely fashion, their location; and to manage their consequences with flexibility so as to maximize the quality of parts produced, stabilize the production rate, and minimize the ensuing costs (Tay, 1999).

A contingency management system always consists of four stages (Wong et al., 1992). They are:

- Stage 1. Conception of an overall strategy.
- Stage 2. Identification of error conditions or faults.
- Stage 3. Implementation of the contingency management system.
- Stage 4. Verification of the contingency management system.

In these four stages, identification of error conditions or faults (stage 2) is an important stage because it will determine the methods and techniques to be implemented in the following stages.

Fault diagnosis on machinery is one of the applications in stage 2. This problem has been well researched in Qu and Meng (1995). Many effective methods are used to diagnose accurately and quickly a certain category of faults (Qu and He, 1981). For instance, large-scale centrifugal compressors can be diagnosed by holospectrum technique (Qu and Shen, 1993). However, up to now, it is difficult to diagnose more than one category of faults. This is especially so in diagnosing the dynamic characteristics of reciprocating machinery, such as reciprocating compressors and diesel engines. This is due to the complex structure of the reciprocating machinery. Although many methods can be used to determine specific fault category, such as broken valve and cracked crankshaft (Huang, 1995), the results obtained from such fault specific method are not easy to interpret. There is a need to have a method that can diagnose more than one category of faults in a generic manner. In the following section, one example to apply the RST to fault diagnosis on a 4135 diesel engine is presented.

3.4.1 The characteristics of vibration signals for a 4153 diesel engine

There are many surveillance methods (Qu and He, 1981; Tan, 1995) proposed to monitor the condition of a diesel engine, based on the processing of the vibration signals. Due to the complex structure and multi-excite sources that exist in a diesel

engine, the vibration signals collated from the engine surface have the following characteristics:

- Presence of a number of self-exciting vibration and forced vibration in the diesel engine that is running. Therefore, the width of spectrum in frequency domain is very large.
- The vibration signals in the time domain are more complex compared to a large-scale rotational machinery, which is a pure sine curve.
- In a diesel engine, such as the 4135 engine, the stroke cycles are fixed. Therefore, the time series appear periodical. However, in every period, there exists many other periodical vibrations within the stroke cycle.

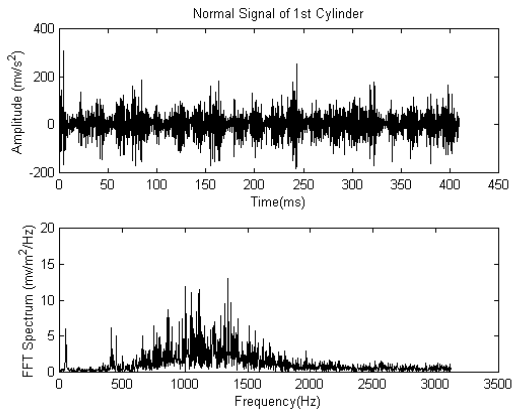


Fig. 3.2 Normal state (sample point 1)

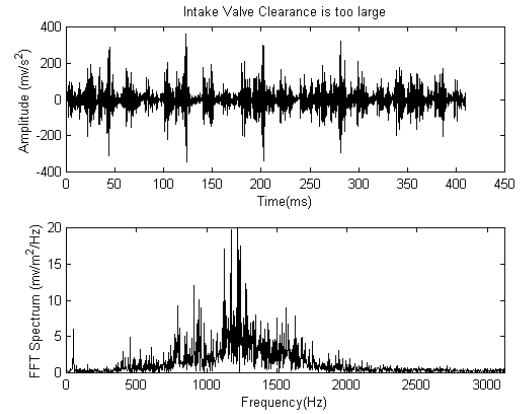


Fig. 3.5 Intake valve clearance is too large (sample point 1)

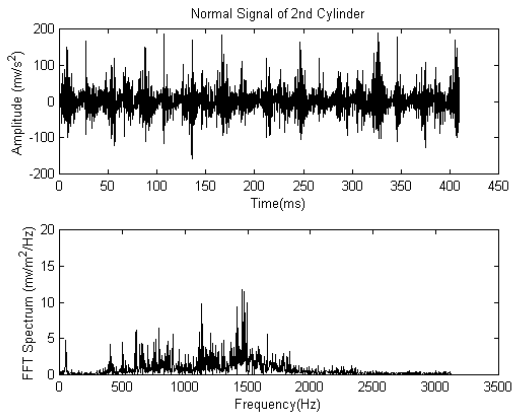


Fig. 3.3 Normal state (sample point 2)

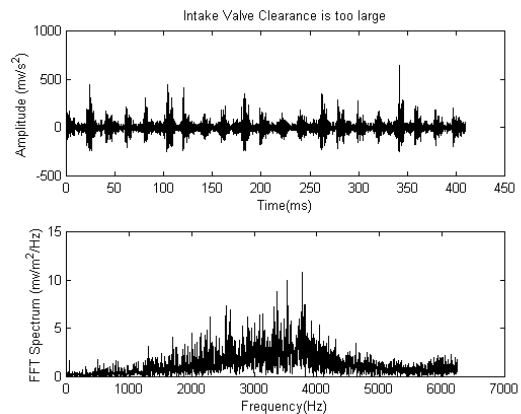


Fig. 3.6 Intake valve clearance is too large (sample point 2)

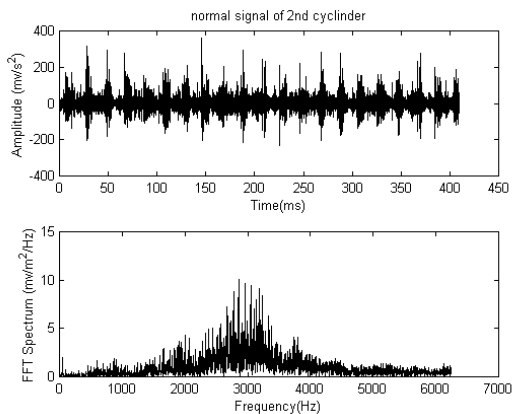


Fig. 3.4 Normal state (sample point 3)

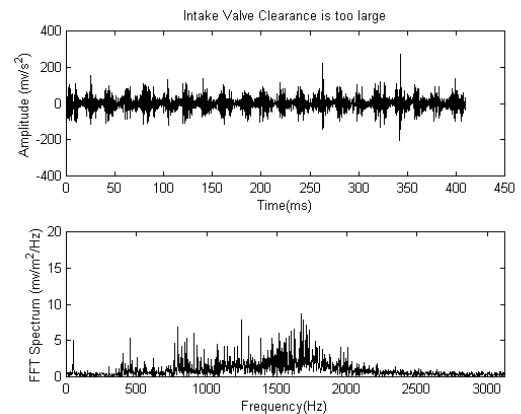


Fig. 3.7 Intake valve clearance is too large (sample point 3)

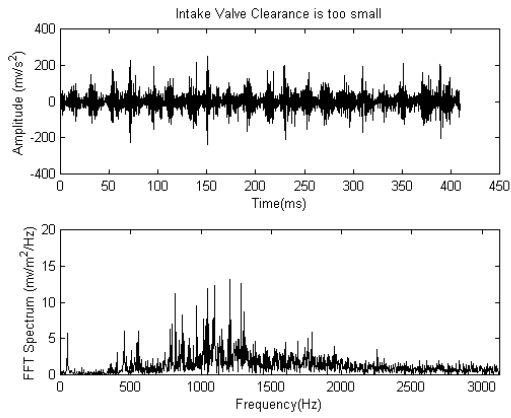


Fig. 3.8 Intake valve clearance is too small (sample point 1)

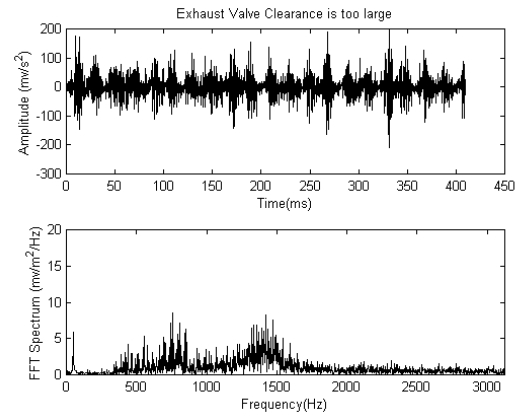


Fig. 3.11 Exhaust valve clearance is too large (sample point 1)

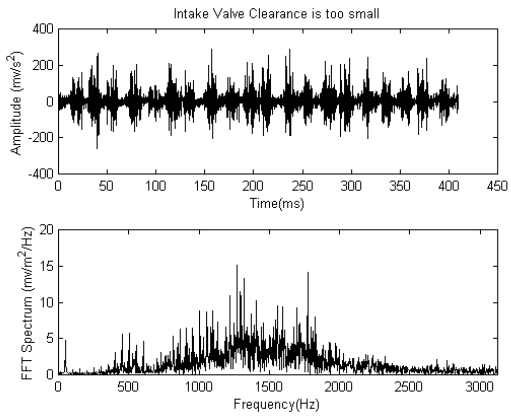


Fig. 3.9 Intake valve clearance is too small (sample point 2)

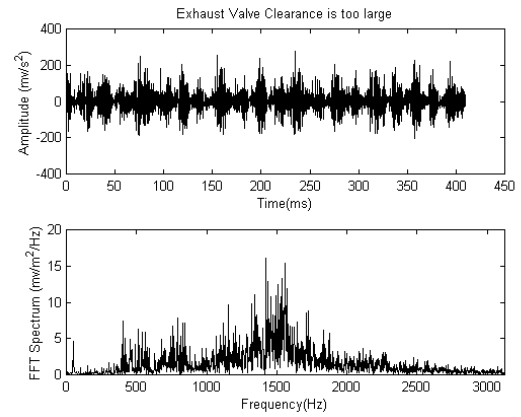


Fig. 3.12 Exhaust valve clearance is too large (sample point 2)

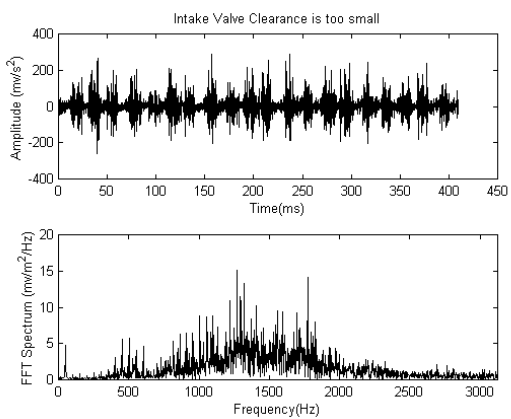


Fig. 3.10 Intake valve clearance is too small (sample point 3)

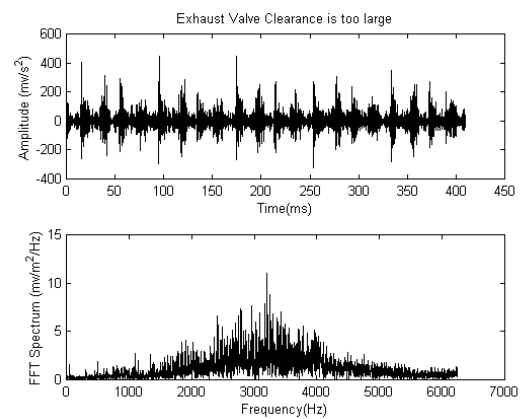


Fig. 3.13 Exhaust valve clearance is too large (sample point 3)

In Figures 3.2-3.13, some vibration signals, collected from a 4135 engine surface, are presented. The parameters of a 4135 diesel engine are:

Rated Engine Power: 80 horsepower

Rated Engine Speed: 1500rpm

Four states are studied in this experiment. They are:

- Normal state
- Intake valve clearance is too small
- Intake valve clearance is too large
- Exhaust valve clearance is too large.

Among these four states, three fault types were simulated in the intake valve and exhaust valve on the second cylinder head. In this process, three points are selected to collect vibration signals. These are the first cylinder head, the second cylinder head and at the center of the piston stroke on the surface of the *cylinder block*. In each of the above figures, the top portion represents the time series and the bottom portion represents the corresponding FFT (Fast Fourier Transformation) spectrum.

According to the diesel engine operating principle, when the clearance of valve increases, the energy will generally be distributed about a higher frequency (Tan, 1995). Thus, theoretically, from the state of small intake valve clearance to normal and to large intake clearance, the mean of the energy distribution will be at a higher frequency.

From the FFT spectrum in Figures 3.2 - 3.13, it is difficult to differentiate the energy change corresponding to the fault type change. The energy spectrum is distributed in a

wide range from 50Hz to 2000Hz for the vibration signals collected from the cylinder head. The energy spectrum covers the entire frequency range for the vibration signals collected from the point in the middle of piston stroke.

In the time domain, every waveform peak represents the *uncontrolled burning* or *flame propagation phase* (Dagel and Brady, 1998). From the time series waveform, no obvious difference exists among the different fault types. Therefore, due to the complexity of the vibration signals, the different valve faults cannot be diagnosed using conventional FFT spectrum.

3.4.2 Specification of attributes field

Before using RST, the attributes field must be specified to compose the decision table. In this experiment, six attributes are extracted from the vibration signal for each sampling point. These six attributes can be divided into two categories, namely, frequency domain and time domain.

(1) Frequency domain attributes:

a. *IF* – Waveform Complexity in frequency domain (Meng, 1996)

$$IF = - \sum_{i=1}^{N/2} X(i) \log X(i) \quad (3.9)$$

where $X(i)$ – the FFT spectrum

From the Eq. (3.9), it can be seen that IF is a frequency domain entropy, reflecting the complexity of FFT spectrum.

b. CG – the center frequency of spectrum (Liu, 1997)

$$CG(X) = \sum_{K=1}^{N/2} \frac{K}{N/2} \mu(X(K)) \quad (3.10)$$

where $\mu(X(K)) = X(K) / \sum_{j=1}^{N/2} X(j)$

$X(K)$ – the FFT spectrum

$k=1, 2, \dots, N$

(2) Time domain attributes:

a. IT – Waveform Complexity in time domain (Meng, 1995)

$$IT = -\sum_{i=1}^m \lambda_i \log \lambda_i \quad (3.11)$$

where λ_i – the singular value of a time series according to its period

m – the number of periods in a time series

The physical significance of IT is that it reflects the complexity of time series. It is time domain entropy.

b. σ - Nonperiod complexity (Liu, 1997)

$$\sigma = \frac{m}{m-1} \sum_{i=2}^m \lambda_i^2 / \sum_{i=1}^m \lambda_i^2 \quad (3.12)$$

where λ_i – the singular value of a time series according to its period.

c. D_x – the variance of time series

$$D_x = \frac{1}{n} \sum_{i=1}^n [x(t_i) - \bar{x}]^2 \quad (3.13)$$

where n – length of a time series

\bar{x} - mean value of the whole series

$x(t_i)$ - time series

d. α_4 – kurtosis of time series

$$\alpha_4 = \frac{1}{n} \sum_{i=1}^n [x(t_i)]^4 \quad (3.14)$$

The above six attributes present the information contained in vibration signals both from the frequency domain and time domain. For IT and σ , they reflect the time series periodical characteristic because the single fault type shows the periodicity in time domain and the energy will increase to a certain frequency in the spectrum, which is reflected by attributes, IF and CG . Variance D_x and Kurtosis α_4 are the measures of the data distribution.

3.4.3 Discretization of attributes

Having established the attributes field, the crucial problem now is how to discretize them because the traditional RST cannot be used to deal with the continuous attributes, which is the disadvantage of this method.

Usually, the value of the attribute changes monotonically with deterioration of the state. Boundary values to divide its domain into intervals corresponding to different

state are selected by experience. However, when experience to discretize the domain is not present, other discretization methods have to be used.

There are many discretization methods, such as *equal-width-intervals*, *equal-frequency-intervals* (Shan et al., 1996) and *Minimal Entropy Method* (Fayyad and Irani, 1992). Among these methods, there are always some disadvantages that make them impractical. In Nowicki et al. (1992), four methods defining attribute limit values were compared. They were called L-, W-, P- and C-method. For these four methods, expert experience is needed to calculate the limit value, so these four methods cannot be used in this case.

In this experiment, a new method is proposed to discretize the attributes extracted from vibration signals. This method extracts the salient features from ChiMerge algorithm (Kerber, 1992) and ϵ -indiscernibility relation algorithm (Krawiec et al., 1996). The algorithm is presented as follows:

Step 1: all the objects are sorted in ascending order of the value of the attributes.

Step 2: calculate the χ^2 value between each adjacent intervals. At the beginning, every object constitutes a separate interval. The calculation of χ^2 value is defined in the following equation:

$$\chi^2 = \sum_{i=1}^2 \sum_{j=1}^k \frac{(A_{ij} - E_{ij})^2}{E_{ij}} \quad (3.15)$$

where

k – class number of decision attribute within two adjacent intervals

A_{ij} – number of objects in the i th interval, j th class,

R_i – number of objects in the interval in the i th interval = $\sum_{j=1}^k A_{ij}$,

C_j – number of objects in the j th class = $\sum_{i=1}^2 A_{ij}$,

N – total number of objects,

E_{ij} – expected frequency of $A_{ij} = R_i * C_j / N$.

Step 3: The two intervals for which the computed χ^2 is minimal is found, if the *purity condition* described below is fulfilled, the selected intervals are merged. If for a selected pair of adjacent intervals, the *purity condition* is not fulfilled, the next pair according to the χ^2 is to be searched for. The process is repeated until there is no pair of intervals fulfilling the *purity condition* or, if the number of clusters decreases to 2.

The *purity condition* is defined as follows. During the entire process, the distribution of class frequencies for each interval (C_j/N) is stored and updated separately. When merging two adjacent intervals, their distributions are summed. Based on that distribution, a majority class, that is, the class with maximal share in the merged interval, can be easily computed. If the share of majority class in the merged interval does not drop below a fixed threshold μ , the two intervals considered as candidates for merging is deemed to fulfill the *purity condition*. The *quality of approximation* $\gamma_B(\Omega)$ is regarded as the criterion to evaluate the final discretization result. If the *quality of approximation* $\gamma_B(\Omega)$ remains stable, the corresponding μ is adopted.

Following the above algorithm, a case study – evaluation of vibroacoustic diagnostic symptoms by means of the RST - in Nowicki et al. (1992) is tested and compared with the L-method mentioned in their paper. The results are presented in Table 3.4 and

Table 3.5. There are 55 objects described by 12 symptoms (represented by $s_1 - s_{12}$) and 1 two-valued decision attribute (Class 0 – bearings in good condition; Class 1 – bearings in a bad state) in this case.

Table 3.4 The quality of classification for different μ

μ		Class 0	Class 1
0.1	Lower approximation	25	3
	Upper approximation	44	22
	Accuracy of approximation	0.5682	0.1364
	Accuracy of classification	0.4242	
	Quality of approximation	0.5091	
0.2	Lower approximation	25	4
	Upper approximation	43	22
	Accuracy of approximation	0.5814	0.1818
	Accuracy of classification	0.4462	
	Quality of approximation	0.5273	
0.3	Lower approximation	32	14
	Upper approximation	40	22
	Accuracy of approximation	0.8000	0.6364
	Accuracy of classification	0.7419	
	Quality of approximation	0.8364	
0.4	Lower approximation	34	21
	Upper approximation	34	21
0.9	Accuracy of approximation	1	1
	Accuracy of classification	1	
	Quality of approximation	1	

From Table 3.4, when value of μ is larger than 0.4, the *quality of approximation* reaches 1. For every attribute in Table 3.5, when the value of μ is larger than 0.4, the *quality of approximation* remain stable. So in this example, 0.4 is assigned to μ .

In the entire calculation, no *a priori* knowledge is needed. The *purity condition* is used to end the discretization. This advantage overcomes the shortcoming of the ChiMerge algorithm, which requires user-defined parameters to select the minimal and maximal intervals. However, this new algorithm only considers the relationship between a single continuous valued attribute and its decision attribute.

Table 3.5 The quality of approximation of each attribute for different μ

μ	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	s11	s12
0.1	0.35	0.29	0.35	0	0.40	0.35	0	0.38	0	0.38	0.36	0.36
0.2	0.35	0.29	0.35	0	0.40	0.35	0	0.38	0	0.38	0.36	0
0.3	0.35	0	0.35	0	0.40	0.35	0	0.38	0.20	0.55	0.36	0
0.4	0.35	0.29	0.35	0	0.65	0.20	0.15	0.60	0.20	0.55	0.36	0.31
0.5	0.45	0.29	0.44	0	0.65	0.20	0.15	0.69	0.47	0.55	0.16	0.31
0.6	0.53	0.36	0.56	0.04	0.65	0.20	0.15	0.69	0.47	0.55	0.16	0.31
0.7	0.53	0.36	0.56	0.04	0.65	0.20	0.15	0.69	0.47	0.56	0.16	0.31
0.8	0.53	0.15	0.56	0.04	0.65	0.20	0.15	0.69	0.47	0.56	0.16	0.31
0.9	0.67	0.56	0.56	0.04	0.65	0.24	0.33	0.69	0.49	0.64	0.20	0.31
L	0	0.04	0.35	0.24	0.02	0.13	0.35	0.13	0.16	0.09	0.09	0.09

3.4.4 Implementation and discussion

The decision table composed of 18 condition attributes (3 sampling points with 6 attributes for each point) and one decision attribute with four states is presented in Table 3.6. The decision attributes correspond to the following four states.

- Normal – 1
- Intake valve clearance is too small – 2
- Intake valve clearance is too large – 3
- Exhaust valve clearance is too large – 4

Having discretized the condition attributes, discernibility matrix is used to obtain the final reducts. Here the quick reducing method (Slowinski and Zopounidis, 1995) is used to choose the minimal reducts. The procedure of the quick reducing method is as follows.

Step 1: The core or attributes with highest *quality of approximation* are calculated.

Step 2: The above core or attributes are augmented by one of the remaining attributes.

The pair with the highest *quality of approximation* is chosen.

Table 3.6 The decision table composed of indices of vibration signals collected from the surface of a 4135 diesel engine

No	The first sampling point					The second sampling point					The third sampling point					D			
	IF	IT	σ	CG	Dx	α4	IF	IT	σ	CG	Dx	α4	IF	IT	σ		CG	Dx	α4
1	968.63	9.463795	0.000236	0.403311	2330.824	4.392614	920.8992	6.527313	9.52E-05	0.40796	1253.28	6.014534	1779.608	12.39776	0.000424	0.471584	2829.108	6.281026	1
2	966.0803	9.12031	0.000216	0.405479	2292.812	4.367723	854.1059	6.564167	9.76E-05	0.399058	1269.885	6.144371	1757.672	12.48108	0.00047	0.464052	2782.227	5.9828	1
3	928.178	9.212937	0.000203	0.408101	2275.398	3.857294	750.0185	7.272025	0.000129	0.386061	1747.084	4.22025	1631.194	11.93877	0.000417	0.481095	3184.857	5.841912	1
4	934.3243	9.652909	0.000266	0.402286	2273.328	3.832025	815.8439	8.049914	0.000175	0.3794	1698.614	4.294675	1689.364	12.26842	0.000451	0.472905	3098.786	5.509754	1
5	906.8166	8.18973	0.000163	0.403422	2393.748	3.969839	929.2076	6.078476	8.79E-05	0.364983	991.9959	3.886004	1657.002	11.52554	0.000387	0.479898	2819.799	6.387094	1
6	913.2175	8.270162	0.000168	0.40688	2417.229	3.97771	911.7933	5.750949	7.11E-05	0.367497	998.8395	3.935522	1632.826	12.2223	0.000442	0.468387	2811.04	6.353366	1
7	860.4138	10.25628	0.000291	0.402683	2364.624	3.934964	1206.267	15.42848	0.000812	0.499007	6187.435	6.20678	1842.024	12.03191	0.000434	0.481547	2962.745	6.938825	1
8	854.1437	10.19812	0.000289	0.400282	2298.57	3.966693	1179.122	16.36945	0.000956	0.48497	6201.076	6.220813	1907.579	11.12551	0.000346	0.494687	2997.293	7.18038	1
9	938.1746	10.29018	0.0003	0.410277	2404.253	4.769968	1003.692	5.862651	7.80E-05	0.420415	740.6773	4.53411	1718.902	11.46342	0.000372	0.497391	2788.227	4.824433	1
10	933.4472	10.92431	0.000347	0.406396	2462.082	4.70767	965.6075	5.571255	7.09E-05	0.421925	704.0458	4.51455	1700.047	12.2018	0.000442	0.493479	2827.68	5.027644	1
11	748.9689	11.53852	0.00042	0.40151	4115.304	7.406265	1083.173	7.379985	0.00013	0.456184	1332.931	4.590098	1856.117	13.308	0.000564	0.507847	3771.855	9.348355	2
12	759.4642	11.8019	0.000409	0.397719	3878.832	6.721322	1063.689	8.410843	0.000194	0.435923	1386.426	5.777319	1840.721	13.14226	0.00054	0.511589	3821.941	11.40681	2
13	828.2558	10.44427	0.000276	0.39677	2994.575	6.282765	1028.062	8.915699	0.000189	0.457331	1918.919	4.391126	2097.51	13.391	0.000507	0.509141	3497.027	7.669647	2
14	834.6539	10.05399	0.000272	0.397975	2905.803	5.931248	1036.01	8.372194	0.000176	0.461422	1897.649	4.367016	2072.981	13.49212	0.000573	0.50746	3508.434	7.633504	2
15	841.6564	11.49167	0.000393	0.407036	3763.61	7.153016	978.5342	9.329181	0.000239	0.452388	2186.351	9.243974	1943.545	13.8059	0.000604	0.505638	3363.842	8.870944	2
16	856.2049	10.90233	0.000335	0.413575	3721.703	7.245844	980.6038	8.691654	0.000196	0.451483	2234.845	10.66554	1989.373	13.8788	0.000649	0.498117	3307.159	9.017534	2
17	837.6024	12.09892	0.00037	0.38792	2842.585	4.466144	1053.039	16.13031	0.000835	0.465067	6109.673	7.04936	1924.621	14.95749	0.000689	0.519337	4255.616	13.00923	2
18	860.2143	11.44314	0.00039	0.396719	2843.152	4.233111	1095.057	14.70214	0.000702	0.482986	5923.511	6.747132	1972.851	14.48197	0.000701	0.523222	4326.552	12.61506	2
19	960.8651	10.30765	0.000296	0.425713	2322.519	5.68833	986.1711	12.45634	0.000469	0.456184	3991.37	5.155113	2052.661	11.15374	0.000352	0.507847	2259.82	5.365647	3
20	1006.686	10.06968	0.000272	0.438123	2399.944	5.655421	982.3972	11.67106	0.000395	0.435923	3784.813	5.057999	2097.9	11.92495	0.000414	0.511589	2361.131	5.99369	3
21	929.9577	10.6088	0.000322	0.416517	2511.911	6.263797	1010.356	17.34732	0.00111	0.457331	6879.374	4.825951	1983.917	11.47818	0.000379	0.509141	2521.838	5.814606	3
22	950.1628	9.867081	0.000258	0.42326	2532.227	6.000203	1020.085	17.95784	0.001219	0.461422	6777.754	4.746955	2013.993	11.72726	0.000398	0.50746	2546.619	6.290961	3
23	981.588	10.45415	0.00027	0.404788	2434.985	6.452266	1038.908	16.07058	0.000908	0.452388	6593.186	5.43869	2129.193	11.72058	0.000408	0.505638	2541.82	5.930342	3
24	952.3014	11.44342	0.000411	0.425976	2411.399	6.479662	1027.337	16.84694	0.001031	0.451483	6621.677	5.140761	2016.157	11.48792	0.00039	0.498117	2486.469	5.903653	3
25	1042.218	9.264848	0.00022	0.42183	2213.716	6.605749	994.8993	15.01201	0.000744	0.482986	5712.068	5.17196	1951.248	11.63425	0.000396	0.523222	2565.575	6.246081	3
26	978.3475	9.350612	0.000227	0.418596	2090.416	6.983087	1057.175	15.5163	0.000818	0.471835	6341.818	4.144594	2145.83	12.04028	0.000439	0.529551	2382.222	5.827804	3
27	976.9817	9.742837	0.000251	0.407637	2101.55	6.964501	1063.936	15.70303	0.000841	0.457609	6216.18	4.116981	2141.317	12.86789	0.000505	0.534045	2452.92	6.125845	3
28	1070.333	8.090354	0.000163	0.409697	1461.063	5.024917	1010.476	11.91118	0.000424	0.437947	3998.688	3.981336	1891.114	12.35004	0.000454	0.500906	3538.96	8.215332	4
29	1073.929	8.027985	0.000156	0.409937	1444.384	5.531318	1016.647	11.61503	0.000388	0.446539	4078.717	4.054873	1876.782	13.03733	0.000539	0.489654	3529.036	8.094243	4
30	978.7982	9.108615	0.000197	0.418583	1744.463	4.577565	998.7954	16.98574	0.000967	0.431412	7240.758	4.770887	2002.549	13.5434	0.000534	0.486128	3531.318	7.881434	4
31	905.979	7.856989	0.000151	0.422491	1794.124	4.729704	1015.769	16.92916	0.001014	0.432994	7213.852	5.011682	1999.3	13.12159	0.000533	0.495409	3591.101	7.407416	4
32	1030.805	9.255664	0.000225	0.419269	1846.876	5.051797	1082.497	16.11993	0.00084	0.418296	6545.535	5.099336	2043.259	14.48351	0.000687	0.484417	3774.407	7.736985	4
33	1039.858	10.42931	0.000309	0.409654	1798.844	5.115474	1094.823	15.05881	0.000763	0.424336	6527.427	5.136675	1958.081	11.9159	0.000421	0.500597	3757.967	7.628443	4
34	969.4824	7.667061	0.000139	0.380958	1485.435	5.246665	1024.016	18.28751	0.001295	0.424527	8084.484	4.826401	1997.751	12.8451	0.000499	0.500984	3557.557	7.417416	4
35	969.2537	7.65842	0.00014	0.381866	1444.417	5.524306	1051.173	16.56692	0.001003	0.428467	7641.543	5.047001	2030.748	12.80923	0.000514	0.498772	3634.432	7.901189	4
36	862.2621	7.294524	0.001235	0.357076	1812.592	4.667809	972.4308	15.9116	0.000802	0.40641	7219.99	5.356308	2025.661	13.37121	0.00052	0.498935	3330.902	6.197613	4
37	867.5084	8.224223	0.000173	0.353314	1826.878	4.685793	1028.222	15.23855	0.000806	0.412785	6332.31	5.779836	1869.5	13.09545	0.000525	0.49861	3215.49	7.485392	4

Step 3: The process is repeated until the *quality of approximation* is equal to one or equals to the *quality of approximation* for all the condition attributes.

According to this quick reducing method, the final reducts are presented in Table 3.7.

Table 3.7 Final reducts for different μ

μ	Quality of Approximation	Final Reducts
0.2	1	$\{a_4, a_9, a_{10}, a_{13}\}; \{a_5, a_7\}; \{a_5, a_{10}\}; \{a_5, a_{13}\}$
0.3	1	$\{a_5, a_6\}; \{a_5, a_{10}\}; \{a_{10}, a_{12}\}$
0.4	1	$\{a_4, a_7\}; \{a_6, a_{12}\}; \{a_5, a_{11}\}; \{a_{10}, a_{12}\}$
0.5	1	$\{a_6, a_{11}\}; \{a_{10}, a_{17}\}$

where $a_1, a_7, a_{13} - IF$ (Waveform Complexity in frequency domain);
 $a_2, a_8, a_{14} - IT$ (Waveform Complexity in time domain);
 $a_3, a_9, a_{15} - \sigma$ (Nonperiod complexity);
 $a_4, a_{10}, a_{16} - CG$ (Center frequency of spectrum);
 $a_5, a_{11}, a_{17} - D_s$ (Variance of time series);
 $a_6, a_{12}, a_{18} - \alpha_4$ (Kurtosis of time series)

In Table 3.7, according to the final reducts, the distribution of attributes belonging to different sampling points can be calculated. With μ value between 0.2 and 0.5, the attribute numbers appearing in the final reducts of sampling point 1 (from a_1 to a_6) – the first cylinder head – is 11/28; the attribute numbers of sampling point 2 (from a_7 to a_{12}) – the second cylinder head – is 14/28, and the attributes appearing in the final reducts of sampling point 3 (from a_{13} to a_{18}) – the middle point in piston stroke of the second cylinder is 3/28. That is to say that the second sampling point, which corresponds to the second cylinder, is more important than the other points in the final decision. This agrees well in practice. The second cylinder was observed to be the most sensitive to the changes in the level of vibration. Thus, the results obtained from the RST agrees well with the observation.

From the final reducts, the most important attributes can be chosen according to how many times they appear in the final reducts. Attributes a_5 and a_{10} have the highest

value (i.e. 6/28). Attribute a_5 is the variance value of the first sampling point and attribute a_{10} is the spectrum center value of the second sampling point.

3.5 Summary

In this chapter, the basic RST is introduced. A review of the development of this theory has been presented, covering different rough set models applied in economic and financial forecasting. From this review, it can be seen this theory has been broadly used in various domains.

Following the introduction of the RST, an example is presented to show how the RST can be applied into contingency management, especially in stage 2, that is, the identification of error conditions or faults. In the case of monitoring a diesel engine, the RST uses extracted rules to flag valve failures as they occur and to localize their causes. From this case study, the procedure of fault diagnosis using RST is summarized as follows. First of all, the decision table is established. In this process, the attributes field has to be specified according to collected signals. Next, using a discretization method, either with expert's experience or not, the continuous valued attributes are transformed to discrete ones. Finally, the RST is used to obtain the final reducts and to extract the rules. These rules are used to distinguish the fault types or to inspect the dynamic characteristic of the machinery.

Through the implementation and results analysis, the following observations and conclusions are made:

- The RST is useful in fault diagnosis problem.

- The RST used to diagnose the valve fault of a 4135 diesel engine has been found to be effective.
- The lack of *a priori* knowledge in practice for fault diagnosis is very prevalent and therefore the new discretization method proposed in this project will be very useful. As for the discretization method, the detailed discussion will be further expanded in the next chapter.

Chapter 4

Discretization Techniques for Rough Set Theory

4.1 Introduction

Before the data can be input a traditional rough set model, they must first be discretized. As a result of discretization, the precision of the original data will be decreased but its generality will be increased. When the subintervals for the discretization are specified by a domain expert following his judgement or using norms established in the subject domain, they are called expert discretization. On the other hand, when they are defined automatically, they are called automatic discretizations (Sugama et al., 1997). In the automatic discretization domain, there are three different axes by which discretization methods can be classified, namely, *global* vs. *local*, *supervised* vs. *unsupervised*, and *static* vs. *dynamic* (Dougherty et al., 1995). A *local* method discretizes in a localized region of the instance space (i.e. a subset of instances) while a *global* discretization method uses the entire instance space to discretize (Chmielewski and Grzymala-Busse, 1996). Many discretization methods, such as *equal-width-intervals* and *equal-frequency-intervals* methods, do not use the class information during the discretization. These methods are called *unsupervised* methods. Likewise, those methods that make use of the class information are known as *supervised* methods. Many discretization methods require a parameter, k , indicating the maximal number of partition intervals in discretizing a feature. *Static* methods,

such as Ent-MDLPC (Fayyad and Irani, 1993), perform the discretization on each feature and determine the value of k for each feature independent of the other features. In contrast, the *dynamic* methods search through the space of possible k values for all features simultaneously, thereby capturing interdependencies in feature discretization.

There are a lot of supervised discretization and unsupervised automatic discretization methods (Dougherty et al., 1995; Fayyad and Irani, 1993; Quinlan, 1986) studied by the machine learning community. However, these methods are seldom applied into pre-processing applications for the RST. The majority of researcher determine the coded decision table (after discretization) based on the knowledge of domain experts. The experts can definitely give more reasonable cut points than the automatic discretization method. However, at times when there is a lack of experts' supervision, automatic discretization methods have to be used as in the case study on the 4135 diesel engine fault diagnosis problem presented in Chapter 3.

The rough set community has also been committed to constructing effective algorithms for new feature extraction, in particular discretization and symbolic attribute value grouping (Chmielewski and Grzymala-Busse, 1996; Lenarcik and Piasta, 1992, 1993 and 1997; Nguyen, 1997; Nguyen and Skowron, 1995 and 1997). The most successful among these methods are:

- discretization techniques (Nguyen, 1997, 1998a and 1998b; Nguyen and Skowron, 1995 and 1997);
- methods of partitioning (grouping) of nominal attribute value sets (Nguyen, 1997; Nguyen and Nguyen, 1998; Nguyen and Skowron, 1997);
- combinations of the above methods (Nguyen and Nguyen, 1998).

In this project, more new indicators will be applied to carry out the technical analysis. Since expert experience in this area is not available, studies on appropriate discretization methods are carried out. In this chapter, the discretization method used as the pre-processing step for the RST is studied. A modified Chi2 algorithm is proposed in the following sections.

4.2 Formulation of Modified Chi2 Algorithm

The χ^2 test is a statistical measure used to test the hypothesis that two discrete attributes are statistically independent. Applied to the discretization problem, it tests the hypothesis that the decision attribute is independent of two adjacent intervals to which a condition attribute belongs. If the conclusion of the χ^2 test is that the decision attribute is independent of the intervals, then the intervals should be merged. On the other hand, if the χ^2 test concludes that they are not independent, it indicates that the difference in relative class frequencies is statistically significant and therefore the intervals should remain separate. The formula for computing the χ^2 value is given by Eq. (3.15). For the convenience of clarity and completeness, it is presented here again:

$$\chi^2 = \sum_{i=1}^2 \sum_{j=1}^k \frac{(A_{ij} - E_{ij})^2}{E_{ij}}$$

where

k – class number of decision attribute within two adjacent intervals

A_{ij} – number of objects in the i th interval, j th class,

R_i – number of objects in the interval in the i th interval = $\sum_{j=1}^k A_{ij}$,

C_j – number of objects in the j th class = $\sum_{i=1}^2 A_{ij}$,

N – total number of objects,

E_{ij} – expected frequency of $A_{ij} = R_i * C_j / N$.

The value for the χ^2 threshold is calculated by selecting a desired significance level α and specifying the number of degree of freedom, which is 1 less than the number of classes.

Based on the χ^2 test, the modified Chi2 algorithm is developed as follows. The modified Chi2 algorithm can be sectioned into two different phases:

Phase1:

```
Set  $\alpha=0.5$ ;
do while (ConCheck (data) <  $\delta$ )
{ for each numeric attribute
{      Sort (attribute, data);
      Chi-sq-init (att, data);
      do
      { Chi-sq-calculation (att, data);
      }while (Merge (data))
}
 $\alpha_0=\alpha$ ;
 $\alpha=\text{decreSigLevel}(\alpha)$ ;
}
```

Phase 2:

```
Set all sigLvl[i]= $\alpha_0$  for attribute i;
do until no attribute can be merged
{ for each mergeable attribute i
{      Sort (attribute, data);
      Chi-sq-init (att, data);
      do
      { Chi-sq-calculation (att, data);
      }while (Merge (data))
      if (ConCheck (data) <  $\delta$ )
          sigLvl[i]=decreSigLevel (sigLvl[i]);
      else attribute i is not mergeable;
}
}
```

where

ConCheck() – consistency check, this subroutine will be described in a later section;

decreSigLevel() – decreasing the significance level by one level, i.e. starting from 0.5, decreasing it by 0.1 each step;

Merge() – returning true or false depending on whether the concerned attribute is merged or not;

Chi-sq-init() – calculation of the A_{ij} , R_i , C_j , N , E_{ij} and k for the calculation of the χ^2 value;

Chi-sq-calculation() – calculation of χ^2 value according to Eq. (3.15).

The first part of this algorithm can be regarded as a generalization of the ChiMerge algorithm (Kerber, 1992). Instead of a predefined significance level α , the modified Chi2 algorithm provides a wrapping that automatically increments the threshold (by decreasing the significance level α). Consistency checking is utilised as a stopping criterion to replace the user specified maximal intervals and minimal intervals in ChiMerge. These enhancements ensure that the modified Chi2 algorithm automatically determines a proper threshold while keeping the fidelity of the original data.

The second phase is to refine the merging process. If any of the attributes consisting of the intervals can be further merged without increasing the inconsistency beyond the given limit, then the merging phase is carried out. The first phase of the modified Chi2 algorithm works on a global significance level α while the second phase uses separate local significance levels for each attribute (Risvik, 1997).

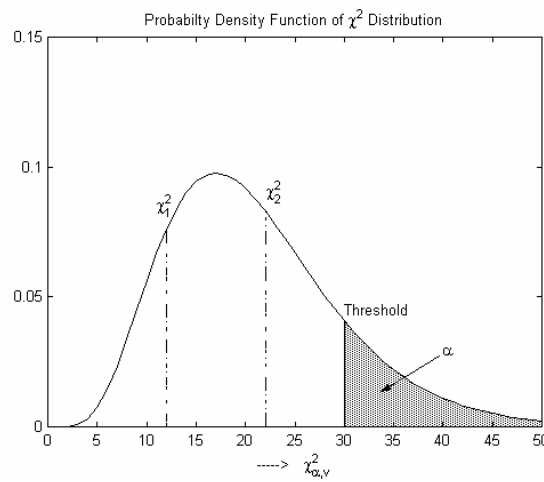


Fig. 4.1 Probability density function of χ^2 distribution (degree of freedom $v=10$)

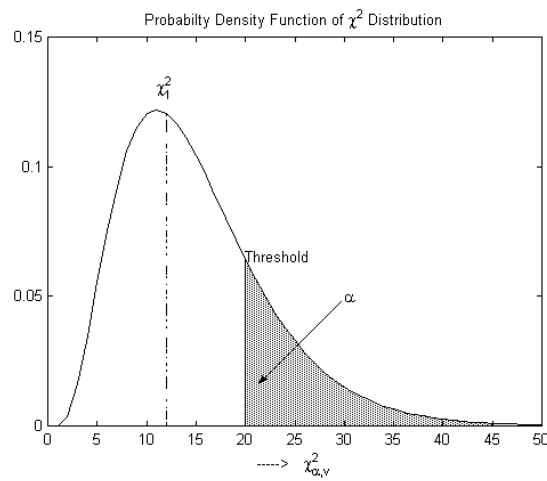


Fig. 4.2 Probability density function of χ^2 distribution (degree of freedom $v=7$)

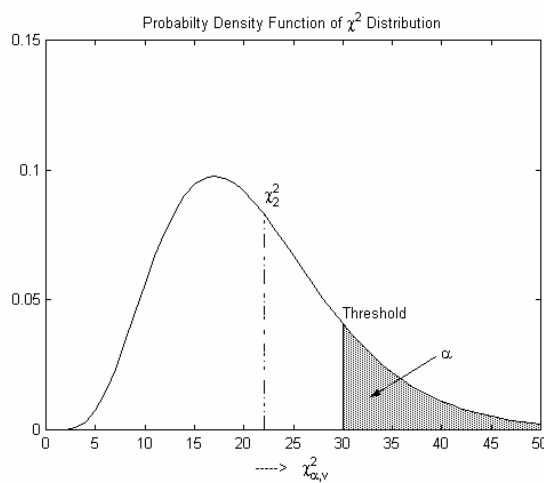


Fig. 4.3 Probability density function of χ^2 distribution (degree of freedom $v=10$)

From the above description, it can be seen this new algorithm is very similar to the Chi2 algorithm proposed by Liu and Setiono (Liu and Setiono, 1997). However, this new method removes the inaccuracies existing in current Chi2 algorithms. By replacing the stopping criterion of the Chi2 algorithm, it has become a more accurate and completely automatic discretization method. In the following sections, the inaccuracy that exists in the Chi2 algorithm and the proposed modifications are presented.

The merging criterion (subroutine Merge ()) of the Chi2 algorithm selects the pair with the lowest χ^2 value for merging. This merging criterion does not consider the degree of freedom, which must first be specified to calculate the χ^2 value. As illustrated in Figure 4.1, only the fixed degree of freedom (the class number-1) and a specified significance value α are used to obtain the threshold. After all the χ^2 values of adjacent interval are computed, the two intervals with the minimal value are merged despite the fact that the degrees of freedom are not the same for different adjacent intervals. From a statistical point of view, this is inaccurate (Montgomery and Runger, 1999). The interpretation is depicted in Figures 4.2 and 4.3. For the convenience of clarity, the minimal difference between the calculated χ^2 values are enlarged. Figures 4.2 and 4.3 show the probability density function of a χ^2 distribution with different degrees of freedom. The two vertical lines represent the χ^2 values calculated from the adjacent intervals and threshold. The shaded areas represent the significance value α . In Figure 4.2, the χ^2 value is 13 while the corresponding threshold is 20. In Figure 4.3, the χ^2 value is 22 while the threshold is 30. If the merging criterion of the Chi2 algorithm is applied, adjacent intervals with χ^2 value equals to 13 are merged compared with threshold 30. However, if the difference in degrees of freedom is considered, from

Figures 4.2 and 4.3, the difference in the χ^2 value and the threshold for the second case (i.e. 8) is larger than that for the first case (i.e. 7). This means that the independence of the two adjacent intervals from the second case is greater than that of the first case. Therefore, these two intervals should be merged first. From the above analysis, it is shown that it is more reasonable and more accurate to take into consideration the degree of freedom.

Since the Chi2 algorithm only considers the maximal degree of freedom, the merging procedure will continue until all the χ^2 values exceed the threshold. This will result in some attributes being over-merged while others being discretized partially. Consequently, it will bring about more inconsistency after discretization.

Another problem in the Chi2 algorithm is its stopping criterion, which depends on the characteristics of the data sets and the targets of the user. The stopping criterion in the Chi2 algorithm is defined as the point which the inconsistency rate exceeds the predefined rate δ ($\text{InConCheck}() > \delta$). Liu and Setiono (1997) assigned different δ values to different data sets for feature selection and some features were removed according to a larger δ value. It can be seen that these results were obtained by decreasing the fidelity of the original data set. In addition, the δ value was obtained only after some tests were performed on the training data set to remove some features. This is unreasonable for an unknown data set. It has been mentioned that discretization is needed to first preprocess the data for the RST. It is ideal to keep the fidelity from the RST point of view. For this purpose, a *quality of approximation* (Pawlak, 1982), coined from the RST, is introduced to replace the inconsistency rate. This concept has always been used to remove the redundant objects and features in the RST.

The *quality of approximation*, denoted γ_B , is defined by Eq. (3.6). For the convenience of clarity and completeness, it is presented here again:

$$\gamma_B = \frac{\sum card(\underline{B}X_i)}{card(U)}$$

where X_i ($X_i \in U$) is a subset of U , $X_i \cap X_j = \emptyset$ and $\cup X_i = U$ ($i=1, 2, \dots, n$)

$\underline{B}X$ is the *lower approximation* of X , $\underline{B}X = \cup \{x_i \in U \mid [x_i]_{IND(B)} \subseteq X\}$.

γ_B represents the percentage of instances which can be correctly classified into class X_i with respect to B . For a consistent data set, $\gamma_B=1$. By using *quality of approximation* as the stopping criterion, it guarantees that the fidelity of the training data is maintained during discretization. In addition, it makes the discretization process completely automatic.

Considering the above 2 modifications, the algorithm with the modified merging criterion and new stopping criterion is termed the “modified Chi2 algorithm”.

4.3 Experimental Results and Discussion – Benchmark C4.5

For the convenience of comparing the modified Chi2 algorithm with the original Chi2 algorithm, 11 data sets were chosen to be discretized. The data contained real-life information from the medical and scientific fields which were used previously in testing pattern recognition and machine learning methods. Table 4.1 gives a summary of data sets used in this experiment.

Table 4.1 Data sets information

Data Set	Name	Examples	Continuous attributes	Discrete attributes	class
Medical data from the University of Wisconsin Hospitals	Breast Cancer	699	9	0	2
A liver disorder data set gathered by BUPA Medical Research Ltd, England	Bupa	345	6	0	2
The glass types data created by Home Office Forensic Science Service, Canada	Glass	214	9	0	6
Medical data from the Cleveland Clinic Foundation	Heart disease	297	5	8	5
Raw data on treatment of duodenal ulcer by HSV[10]	Hsv - r	122	9	2	4
The famous iris classification data by R. A. Fisher	Iris	150	4	0	3
Wine recognition data from Institute of Pharmaceutical and Food Analysis and Technologies, Italy	Wine	178	13	0	3
Blocks Classification from University of Bari, Italy	Page-blocks	5473	10	0	5
Optical Recognition of Handwritten Digits data from Bogazici University, Turkey	Optdigit	5620	56	8	10
The training part of Pen-Based Recognition of Handwritten Digits data from Bogazici University, Turkey	Pendigit	7494	12	0	10
Data set generated from Landsat Multi-Spectral Scanner image data	Satellite	6435	36	0	6

The data set *hsv-r* (Slowinski, 1992) represented raw data on treatment of duodenal ulcer by HSV. The remaining 10 data sets were taken from the University of California at Irvine repository of machine learning databases (Merz and Murphy), including 4 large data sets (examples > 1000). The reason for choosing these 4 large data sets is to test if the modified Chi2 algorithm produce large discrete groups because χ^2 values grow proportionally with the number of instances. All of the above 11 data sets have level of consistency equal to 1 except for the *page-blocks* data set which is 0.9916.

In the following experiments, C4.5 (Release 8) (Quinlan, 1993) was chosen to be the benchmark for evaluating and comparing the performance of the modified and original

Chi2 algorithm. The reasons for this choice were that C4.5 worked well for many making-decision problems and it was a well-known method, thus requiring no further descriptions (Liu and Setiono, 1997). C4.5 was selected as the benchmark to evaluate the original Chi2 algorithm in Chmielewski and Grzymala-Busse (1996) and has shown that the Chi2 algorithm was an effective discretization method. To compare the efficacy of these two methods, the predictive accuracy of C4.5 on the undiscretized data sets was presented, which was denoted by *Continuous* in Table 4.2 and Table 4.3. C4.5 was run using its default setting and the predictive accuracy was chosen as the evaluation benchmark.

The ten-fold cross-validation test method (Weiss and Kulikowski, 1990) was applied to all the data sets. The data set was divided into ten parts of which nine parts were used as training sets and the remaining one part as the testing set. The experiments were repeated ten times. The final predictive accuracy was taken as the average of the ten predictive accuracy values. As the two modifications on the original Chi2 algorithm were made to maintain the fidelity of the original data set, the modified Chi2 algorithm was compared with the original Chi2 algorithm with threshold δ value of 0 in the experiments, except for the *page-blocks* data set with an inconsistency rate of 0.0022. In addition, since the modified Chi2 algorithm was a parameter free discretization method, its efficacy in discretization was compared to another parameter-setting-free method – the Ent-MDLPC algorithm (Fayyad and Irani, 1993), which has been accepted as one of the best supervised discretization methods (Dougherty et al., 1995; Kohavi and Sahami, 1996). All 11 data sets were first discretized using the original Chi2 algorithm, the modified Chi2 algorithm and the Ent-MDLPC algorithm respectively, after which the discretized data sets were input

into C4.5. The predictive accuracy and the standard deviation of these four methods were presented in Table 4.2. The tree size using C4.5 with different discretization methods was presented in Table 4.3.

Table 4.2 The predictive accuracy (%) using C4.5 with different discretization algorithm

Data Set	C4.5				
	Continuous	Original algorithm ($\delta=0$)	Chi2 Modified algorithm	Chi2	Ent – MDLPC
Breast Cancer	95.00± 3.00	96.77± 3.81	97.21± 5.54		95.88± 4.43
Bupa	68.41± 4.81	50.00± 9.10	50.29±11.13		63.53± 8.23
Glass	68.69±13.95	36.19±23.14	32.86±18.21		71.43± 8.69
Heart disease	53.87± 7.67	55.17±10.41	60.69± 9.06		54.48±11.92
Hsv – r	59.02±10.47	54.17±22.29	58.33±11.91		60.67±13.59
Iris	95.33± 6.32	94.00± 7.98	94.67± 7.57		94.00± 8.58
Wine	92.70± 7.42	88.82±14.79	93.21±12.02		87.65±11.25
Page-blocks	96.84± 1.13	93.97± 2.39 ($\delta=0.0022$)	94.61± 3.25		95.14± 2.51
Optdigit	81.23± 0.97	72.72± 2.98	76.05± 3.49		79.63± 2.20
Pendigit	90.01± 0.53	78.40± 1.63	85.41± 1.11		88.77± 1.63
Satellite	86.22± 1.26	80.86± 4.38	81.26± 3.99		81.62± 3.95
Average	80.67	72.82	74.96		79.35

To analyze the results obtained in Table 4.2, the Wilcoxon matched-pairs sign-rank test (Montgomery and Runger, 1999) was applied. The purpose of this nonparametric test was to determine if significant differences existed between two populations. Paired observations from the two populations were the basis of the test, and magnitudes of differences were taken into consideration. This was a straightforward procedure to either accept or reject the null hypothesis, which was commonly taken to be identical population distributions.

The modified Chi2 algorithm outperforms the original Chi2 algorithm at 1% significance level for a one-tailed test. However, it shows no significant performance difference from the C4.5 and Ent-MDLPC algorithm, that is, the null hypothesis could not be rejected even at the 5% significance level for a one-tailed test. The C4.5 outperforms the original Chi2 algorithm at 0.5% significance level for a one-tailed test.

However, it shows no significant difference in performance from the Ent-MDLPC algorithm. The Ent-MDLPC algorithm outperforms the original Chi2 algorithm at 5% significance level for a one-tailed test.

Table 4.3 The tree size (before / after pruning) comparison of 4 methods

Data Set	C4.5				
	Continuous	Original algorithm ($\delta=0$)	Chi2	Modified algorithm	Chi2 Ent – MDLPC
Breast-cancer	128.0±13.37/ 37.0±17.63	52.3±7.94/ 19.5±2.99		53.7±7.92/ 22.4±6.20	48.2±7.77/ 20.8±3.39
Bupa	57.0±12.33/ 43.8±12.51	292.0±80.44/ 43.5±31.49		298.7±45.93/ 48.0±34.15	52.4±11.39/ 45.8±10.16
Glass	48.6±5.72/ 44.6±5.72	148.8±42.76/ 65.4±17.13		161.7±33.09/ 73.3±32.62	40.3±8.58/ 33.2±7.56
Heart disease	124.2±5.79/ 83.1±15.25	185.5±22.06/ 71.2±14.83		162.4±24.78/ 74.3±11.10	124.7±11.55/ 81.3±16.24
Hsv-r	41.8±3.58/ 30.8±5.75	62.4±14.52/ 15.9±9.04		58.0±11.73/ 3.1±4.48	38.6±6.74/ 31.0±8.18
Iris	8.8±1.14/ 8.8±1.14	21.7±3.86/ 5.2±2.70		24.2±6.48/ 6.6±3.24	5.9±3.07/ 4.0±0.00
Wine	9.6±1.35/ 9.2±0.63	26.2±5.18/ 19.7±4.06		29.1±9.36/ 19.2±3.85	19.7±3.43/ 16.5±2.64
Page-blocks	119.4±12.50/ 80.4±12.00	3056.9±363.32/ 417.1±115.85 ($\delta=0.0022$)		3876.9±912.32/ 558.9±185.69	407.2±45.75/ 132.0±15.30
Optdigit	438.8±18.12/ 410.4±16.39	4878.7±91.79/ 3028.7±86.49		3123.9±89.85/ 2020.0±100.19	1942.3±62.51/ 1323.7±48.34
Pendigit	311.2±10.64/ 286.2±11.63	9716.5±520.62/ 4605.2±261.33		4643.5±137.29/ 2448.7±132.73	2595.9±79.62/ 1480.5±56.64
Satellite	656.4±18.21/ 48.8±17.87	5322.3±225.16/ 1763.7±111.71		4501.3±240.11/ 1546.5±85.30	3534.7±132.57/ 1416.9±81.65

From Table 4.3, it can be seen that the tree size after application of the three discretization algorithms is mostly reduced compared with the C4.5. This indicates that the three methods effectively discretize the numerical attributes and remove the irrelevant and redundant attributes for the subsequent C4.5 processing. However, for the 4 large data sets, although there is no significant difference in the predictive accuracy, the tree size is significantly bigger than those of the C4.5 for the remaining three algorithms. This means these three methods generate too many discrete intervals for a larger data set and they are more suitable for medium size data sets than the C4.5.

4.4 Experimental Results and Discussion – Benchmark RoughSOM

Since the modified Chi2 algorithm is studied for the RST, it should be tested based on the rough set model. Therefore, an experiment using the RST as the benchmark is carried out.

As discussed in the previous section, the modified Chi2 algorithm is more suitable for medium size data sets. In the following experiment, the examples of the data sets are limited within 1000 objects. There are 11 data sets chosen for the experiment, including 5 data sets from previous experiment. Another 6 data sets are added in this experiment. Table 4.4 gives a summary of the data sets used in this experiment. All of the 11 data sets have *quality of approximation* equal to 1.

The ten-fold cross-validation test method (Weiss and Kulikowski, 1990) was applied to all the data sets. The modified Chi2 algorithm was compared with the original Chi2 algorithm with threshold δ value equaled to 0 in the experiments. As was the case for the experiment in the pervious section, its efficacy in discretization was compared to another parameter-setting-free method – the Ent-MDLPC algorithm (Fayyad and Irani, 1993). All 11 data sets were first discretized using the original Chi2 algorithm, the modified Chi2 algorithm and the Ent-MDLPC algorithm respectively, after which the discretized data sets were processed using the RST. Here the RoughSOM system, which will be described in detail in Chapter 5, is then applied to build the sorting system. The predictive accuracy and standard deviations of these 3 methods are presented in Table 4.5.

Table 4.4 Data sets information

Data Set	Name	Examples	Continuous attributes	Discrete attributes	class
Australian credit card applications data set provided by quinlan@cs.su.oz.au	Australian	690	6	8	2
A liver disorder data set gathered by BUPA Medical Research Ltd, England	Bupa	345	6	0	2
Pima Indians Diabetes Database from National Institute of Diabetes and Digestive and Kidney Diseases	Diabetes	768	8	0	2
Medical data from the Cleveland Clinic Foundation	Cleveland	297	5	8	5
Heart disease dataset with 270 cases	Heart disease	270	5	8	2
The glass types data created by Home Office Forensic Science Service, Canada	Glass	214	9	0	6
Raw data on treatment of duodenal ulcer by HSV[10]	Hsv – r	122	9	2	4
The famous iris classification data by R. A. Fisher	Iris	150	4	0	3
Thyroid gland data donated by Stefan Aeberhard, Dept. of Comp. Science, James Cook University, Australia	New-thyroid	215	5	0	3
Vehicle silhouette data set from the Turing Institute, Glasgow, Scotland	Vehicle	846	18	0	4
Wine recognition data from Institute of Pharmaceutical and Food Analysis and Technologies, Italy	Wine	178	13	0	3

Table 4.5 The predictive accuracy (%) using RoughSOM with different discretization algorithm

Data Set	Original Chi2 algorithm	Modified Chi2 algorithm	Ent – MDLPC
Australian	78.26± 4.43	78.84± 6.42	80.93± 5.34
Bupa	53.53± 9.16	58.82±10.83	52.35±17.31
Diabetes	70.26± 6.74	73.95± 7.07	72.87± 6.39
Cleveland	56.21± 8.30	55.86± 7.23	57.59± 6.31
Heart disease	79.26± 6.34	80.37± 4.95	78.52± 7.96
Glass	41.43±28.04	41.90±24.67	48.10±19.50
Hsv – r	64.17±16.69	64.17±15.74	56.67±19.56
Iris	93.33± 7.70	94.00± 7.34	91.33± 9.45
New-thyroid	95.24±10.29	94.76± 9.38	91.90±10.05
Vehicle	60.71± 3.59	63.57± 4.53	63.93± 4.20
Wine	85.88±18.23	88.82±10.54	88.24±14.14
Average	70.75	72.28	70.58

According to the analysis results using the Wilcoxon matched-pairs sign-rank test (Montgomery and Runger, 1999), the modified Chi2 algorithm outperforms the original Chi2 algorithm at the 1% significance level for a one-tailed test. This result supports the proposed modifications. However, the results show no significant difference in performance between the modified Chi2 algorithm and Ent-MDLPC algorithm, that is, the null hypothesis can not be rejected even at the 5% significance level for a one-tailed test. The original Chi2 algorithm and the Ent-MDLPC algorithm have no significant difference in performance.

From the point of view of computational complexity, the modified Chi2 algorithm does not increase the computational complexity as compared to the original Chi2 algorithm although the former involves an additional step (i.e. to select merging intervals). It is an efficient discretization method for the RST.

4.5 Case Re-Study - The Fault Diagnosis on a 4135 Diesel Engine

In Chapter 3, an example on the fault diagnosis of a 4135 diesel engine using RST was presented. In that case study, a discretization method using the *purity condition* as the merging criterion is applied to discretize the attributes. As there is no obvious stopping criterion is provided, the discussion on the final results appears not so deterministic. Hence, the case study – fault diagnosis on a 4135 diesel engine - on the same subject matter but using the modified Chi2 algorithm is carried out to determine whether the conclusion drawn in Chapter 3 is still valid.

The decision table is given by Table 3.6. This decision table is discretized using the modified Chi2 algorithm. After the discrete decision table is processed using the RST, there are 149 final reducts generated with empty core. The core of attributes is empty, which means that no single attribute is absolutely necessary for perfect approximation of the decision classes. One reason for such an outcome is that the condition attributes are not very representative. They cannot reflect the inner information very well. The other reason is that cases collected from vibration signals are less than the expected. There are all together 37 cases corresponding to four decision states. The extracted rules are very dispersive as these cases are not repetitive. From another viewpoint, it shows that the fault diagnosis of a diesel engine is a very difficult and complex problem.

In Table 4.6, according to the minimal reducts, the distribution of attributes belonging to different sampling points – *support*, can be calculated.

Table 4.6 The strength of every attribute appeared in the final reducts

Attribute	1	2	3	4	5	6
Strength	28	12	0	29	20	40
Attribute	7	8	9	10	11	12
Strength	26	31	31	51	31	12
Attribute	13	14	15	16	17	18
Strength	24	16	22	53	23	65

In Table 4.6, the attribute numbers appearing in the final reducts of sampling point 1 (from a_1 to a_6) – the first cylinder head – is 129/514; the attribute numbers of sampling point 2 (from a_7 to a_{12}) – the second cylinder head – is 182/514, and the attributes appearing in the final reducts of sampling point 3 (from a_{13} to a_{18}) – the middle point in piston stroke of the second cylinder – is 203/514. This means that the second and third sampling point, which corresponds to the second cylinder, is more important than the

other points in the final decision. This agrees well with practice. As all the three fault states are simulated on the second cylinder, the second cylinder was observed to be the most sensitive to changes in the level of vibration. Thus the results obtained from the RST agrees well with the observation. It also supports the analysis in Chapter 3.

Table 4.7 The quality of approximation of every attribute

Attribute	1	2	3	4	5	6
QA	0.08	0	0.14	0.24	<u>0.49</u>	0
Attribute	7	8	9	10	11	12
QA	0.24	0.19	0.19	0.05	0.16	0.11
Attribute	13	14	15	16	17	18
QA	0.22	0	0.16	0.19	<u>0.70</u>	0.16

From Table 4.7, the important attributes can be chosen according to their *quality of approximation*. Attributes a_{17} , a_5 , a_4 , a_7 and a_{13} have the higher value compared to other attributes. Attributes a_{17} and a_5 are the “variance” of the third and first sampling points, attribute a_7 and a_{13} are the “waveform complexity in frequency domain” of the second and third sampling points and a_4 is the “the centre frequency of spectrum” of the first sampling point. From Table 4.6, it can be seen that the information arising from the second and third sampling point is more important for the valve fault diagnosis than that of the first sampling point since the *quality of approximation* of the second and third sampling points are much greater than that of the first sampling point. These results are different from those in Chapter 3, with the attributes a_5 and a_{10} being most important attributes. It can be seen that different discretization methods would yield different results. Expert’s experience will be valuable in evaluating such results.

To show the ability of the RST in fault diagnosis, the cross-validation test is applied to an information table. The whole data set is divided into two parts equally. Each part

contains half of the objects in each class. These two parts are trained and tested alternatively. The classification accuracy is listed in Table 4.8.

Table 4.8 The classification accuracy of each part

Data set	1 st part – training data	2 nd part - training data
	2 nd part – testing data	1 st part - testing data
Classification accuracy	0.78947	0.73684

It can be seen that the RST can easily be applied to distinguish the multiple fault categories in a system. It is a promising method in solving the complex fault diagnosis problem.

4.6 Summary

In the data mining community, many algorithms can only acquire knowledge on the nominal features. However, a lot of real world classification tasks consist of continuous features. Even though the RST is an appropriate knowledge-mining tool for such tasks, it cannot be applied to generate rules from the continuous features unless they are first discretized. This requires a discretization method to pre-process the data. In this chapter, a modified Chi2 algorithm is proposed as a completely automated discretization method for the RST. It replaces the inconsistency check in the original Chi2 algorithm using a Quality of Approximation, coined from the RST, which maintains the fidelity of the training set after discretization. In contrast to the original Chi2 algorithm which ignores the effect of degree of freedom, this modified algorithm takes into consideration the effect of degree of freedom which consequently results in greater accuracy. With these modifications, the ChiMerge has become a completely automated discretization method and its predictive accuracy is better than the original

Chi2 algorithm. This is the same with either C4.5 or the RST as the benchmark. However, compared with the C4.5 and Ent-MDLPC algorithm, the modified Chi2 algorithm has no significant performance difference in predictive accuracy. In addition, for a large data set, it generates a larger tree compared with the C4.5. This is not favourable to its application and this problem will be studied in the following research.

With the introduction on this new discretization method, the 4135 diesel engine fault diagnosis problem is re-studied. The analysis results support part of the conclusions drawn in Chapter 3. However, due to different discretization methods, the important attributes presented in final results are different. Expert experience will be the final judge on this problem. The cross-validation test on the data shows that fault diagnosis based on the RST is applicable and promising.

Chapter 5

RoughSOM System

5.1 Introduction

In Chapter 4, one RoughSOM system has been used to be the benchmark for testing the modified Chi2 algorithm against the original Chi2 algorithm. In this chapter, this system will be described.

Decision making is one of the most natural actions of human beings. A lot of researchers have undertaken studies to provide various mathematical tools to deal with it. Generally speaking, a decision making problem involves a set of *objects* (actions, states and competitors among others) described or evaluated by a set of *attributes* (criteria, features and issues among others) in the form of a *table* whose rows and columns correspond to *objects* and *attributes* respectively. From the decision analysis, important facts and dependencies in the *table* to describe a decision situation are determined.

When making real decisions, the decision maker (DM) usually takes into account multiple points of view (criteria) for evaluation of decision alternatives. However, the multi-criteria decision problem has no solution without additional information about the DM's preferences (Slowinski 1993). Having the preferential information, a global

preference model based on the multiple criteria can be built to provide the best solution for a given decision problem.

There are two major ways of constructing a global preference model upon preferential information obtained from the DM involved in the decision process (Pawlak and Slowinski 1994). The first method involves a mathematical decision analysis and consists of building a functional or relational model (Roubens and Vincke 1985). The second way is based on artificial intelligence and builds up the global preference model via inductive knowledge acquisition. The resulting model is a set of ‘if ... then ...’ rules or a decision tree (Michalski 1983; Quinlan 1986). New objects are sorted by matching its description to one of the rules. The second method of constructing the global preference model is studied in this project.

The information concerning a decision situation is usually vague (inconsistent) because of uncertainty and imprecision arising from many sources (Roy 1989). Vagueness may be caused by granularity of representation of the information. For example, if a global preference model is assessed in the form of generated rules, the ambiguity makes some rules non-deterministic, that is, they are not univocally described by means of ‘granules’ of the representation of preferential information.

The RST is one of frameworks proposed for dealing with granularity of information. This theory assumes a representation of the information in a decision table known as an Information System. This table is just an appropriate form for the description of the decision situation. Through many years of development, the RST has proved to be a useful tool for analyzing multi-attribute decision problems (Slowinski 1992b). Based

on the indiscernibility among objects, RST uses a *lower* and an *upper approximation* of a set to determine how exactly a new object can be described using available information.

Using RST to analyze a preferential information system can generate the following results:

- Evaluation of importance of particular attributes;
- Construction of minimal subsets of independent attributes ensuring the same quality of sorting as the whole set, that is, reducts of system;
- Non-empty intersection of those reducts gives a core of attributes which is indispensable attribute in system;
- Elimination of redundant attributes from system;
- Generation of decision rules from the reduced system. These rules involve the relevant attributes only and explain a decision policy of the DM.

The classification of a new object can be supported by matching its description to one of the decision rules. The matching may lead to one of four situations:

- (i) The new object matches exactly one of deterministic decision rules;
- (ii) The new object matches exactly one of non-deterministic decision rules;
- (iii) The new object does not match any of the decision rules;
- (iv) The new object matches more than one rule.

In (i), the classification suggestion is obvious. In (ii), however, the suggestion is not direct since the matched rule is ambiguous. In this case, the DM is informed of the number of training objects that support each possible class. The number is called the

rule strength. If the *rule strength* of one class is greater than the *rule strength* of other class occurring in the non-deterministic rule, one can conclude that according to this rule, the considered object most likely belongs to the strongest class.

Situation (iii) is more difficult to solve. In this case, a set of rules ‘nearest’ to the description of the new object is presented to the DM to make decision. The notion of ‘nearest’ involves the use of the distance measure such as a *valued closeness relation* R proposed by Slowinski (1993).

Situation (iv) may also be ambiguous if the matched rules (deterministic or not) lead to different classes. Here, the suggestion can be based either on the *rule strength* of possible classes, or on an analysis of the training objects which support each possible class. In the latter case, the suggested class is that which is supported by a classification problem closest to the new object based on the relation R .

In the solutions to situation (ii) and (iv) mentioned above, one situation is missing, that is the situation whereby the *rule strengths* of both classes are the same in which case there is no clear indication on how to classify the new object. For situation (iii), from the view of *valued closeness relation* R , there are also some parameters such as the *relative importance* k_l and *veto threshold of criterion* c_l (Slowinski 1993) which have to be pre-defined. This is not favorable to the users without sufficient experience. In order to enhance the classification capability of the RST and remove some ambiguities of the system, the current research is undertaken. In the RST, although no exterior information is needed in generating the rules, the “inner relationships” among objects are ignored. Here “inner relationship” refers to the class the object belongs to as

determined by a cluster analysis. This inner relationship is helpful in distinguishing the “strong” objects from the “weak” objects. By “strong object”, one means that the inner category for this object as determined by cluster analysis is the same as the original decision class. Otherwise, the object is termed as a “weak object”. The information inherent in the objects helps to remove the uncertainty from the system and increase the sorting accuracy on the new objects. This is especially efficient for inconsistent systems. The SOM (Self-Organizing Map) is applied in this project as a cluster method.

5.2 The Self-Organizing Map (SOM)

The Self-Organizing Map (SOM), developed by Teuvo Kohonen (Kohonen 1995), is one of the most popular neural network models. The SOM algorithm is based on unsupervised competitive learning; the output neurons of the network compete among themselves to be activated or fired, with the result that only one output neuron, or one neuron per group, is on at one time. The result of this process is both a clustering of similar elements and an ordering of the clusters within the observation space. In the following, the basic algorithm of SOM is presented.

5.2.1 Basic algorithm

The basic idea of SOM is simple yet effective. The basic SOM consists of m neurons located on a regular low-dimensional grid, usually 1 or 2-dimensional. The lattice of the grid is either hexagonal or rectangular.

The basic SOM algorithm is iterative. Each neuron i has a d -dimensional prototype vector $m_i = [m_{i1}, m_{i2}, \dots, m_{id}]$. At each training step, a sample data vector x is randomly chosen from the training set. Distance between x and all the prototype vectors are computed. The best-matching unit (BMU), denoted by b , is the map unit with prototype closest to x :

$$\|x - m_b\| = \min_i \{\|x - m_i\|\} \quad (5.1)$$

where $\|\cdot\|$ is the distance measure

Next, the prototype vectors are updated. The BMU and its topological neighbors are moved closer to the input vector in the input space, as shown in Figure 5.1. The update rule for the prototype vector of unit i is:

$$m_i(t+1) = m_i(t) + \alpha(t)h_{bi}(t)[x - m_i(t)] \quad (5.2)$$

where t - time

$\alpha(t)$ - learning rate

$h_{bi}(t)$ - a neighborhood kernel centered on the winner unit. The kernel function

can be for example Gaussian function $h_{bi}(t) = e^{-\frac{\|r_b - r_i\|^2}{2\sigma^2(t)}}$, where r_b and r_i are positions of neuron b and i on the SOM grid and $\sigma(t)$ is the neighborhood radius.

Both learning $\alpha(t)$ and neighborhood radius $\sigma(t)$ decrease monotonically with time. During the training step, the SOM behaves like a flexible net that folds onto the “cloud” formed by the training data. The prototype vectors are iteratively adjusted to correspond to the training data, because of the neighborhood relations, neighboring

prototypes are pulled to the same direction, and thus prototype vectors of neighboring units resemble each other.

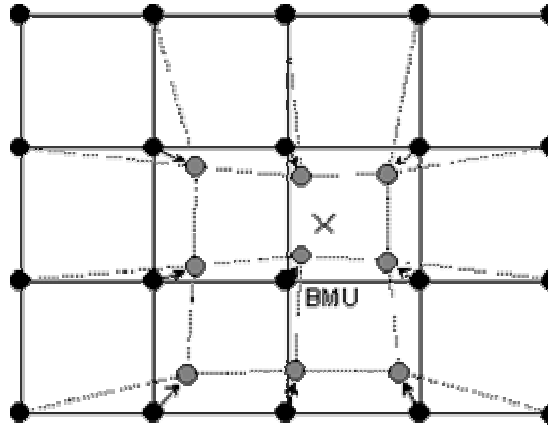


Fig. 5.1 Updating the best matching unit (BMU) and its neighbors towards the input sample x . The black and grey circles correspond to situation before and after updating, respectively. The line shows neighborhood relations.

5.2.2 Properties of SOM

Supervised algorithms, such as the Multi-Layered Perceptron (MLP), require the target values for each data vector to be known. In contrast, the SOM does not have this limitation. Since its introduction in 1981, the SOM has been applied in a large variety of tasks ranging from machine vision to full-text analysis and from process control to neuro-physiological research (Demartines and Herault 1997).

The main properties of SOM are as follows (Kaski and Kohonen 1995):

- The mapping represents the full set of data in an ordered form. Mutual similarities in the data samples are represented as geometric relationships on the map.

- The structures in the data set can be visualized on the map automatically whereby the degree of clustering is represented by shades of grey.
- The natural distribution of the data samples enables the map to be used as a natural framework, on which the individual statistical indicators can be visualized as different shades of grey thus making complex relationships clearly visible.
- As the process does not require all the information, missing data is not a problem.
- Even if no explicit clusters exist in the data set, the self-organizing mapping method reveals “ridges” and “ravines”. The former is the open zone with irregular shapes and high clustering tendencies whereas the latter separates data sets that have a different statistical nature (Kaski and Kohonen 1996).

These SOM properties address in practice the main objectives of data analysis but more broadly the objectives in KDD, which are concerned with the creation and extraction of the underlying knowledge of the domain (Fayyad et al. 1996). These proprieties are congruent with the author’s overall aim of constructing and exploring a new conceptual space.

The SOM is chosen as one of representative clustering methods in this project. It is used to find the “inner relationship” among training data sets. In the following section, a simple case study is presented to show how SOM works to reduce the uncertainty in the RST sorting system.

5.3 Case Study

In order to explain how cluster analysis helps to remove uncertainty in a system, a case study taken from “Selection of Candidates to a School in Slowinski (1993)” is presented in the following. The decision table is given in Table 5.1.

Table 5.1 Decision table composed of sorting examples

Criterion Candidate	c ₁	c ₂	c ₃	c ₄	c ₅	c ₆	c ₇	Decision D	SOM d	Support
x ₁	4	4	4	4	2	2	1	A	/	/
x ₂	3	3	4	3	1	1	1	R	R	1
x ₃	3	4	3	3	2	2	2	R	R	1
x ₄	5	3	5	4	1	1	2	A	A	1
x ₅	4	4	5	4	2	2	1	A	A	1
x ₆	3	4	3	3	1	1	3	R	R	1
x ₇	4	4	5	4	2	2	2	A	A	1
x ₈	4	4	4	4	2	2	2	A	A	1
x ₉	4	4	4	4	2	2	2	R	A	0.5
x ₁₀	5	3	5	4	1	1	2	A	A	1
x ₁₁	5	4	4	4	1	1	2	A	A	1
x ₁₂	5	3	4	4	2	2	2	A	A	1
x ₁₃	4	3	3	3	2	2	2	R	R	1
x ₁₄	3	3	4	3	3	3	3	R	R	1
x ₁₅	4	5	5	4	1	1	1	A	A	1

where: c₁ – score in mathematics, {5, 4, 3};

c₂ – score in physics, {5, 4, 3};

c₃ – score in English, {5, 4, 3};

c₄ – mean score in other subjects, {5, 4, 3};

c₅ – type of secondary school, {1, 2, 3};

c₆ – motivation, {1, 2, 3};

c₇ – opinion from previous school, {1, 2, 3}.

d = A – admission; d = R – rejection;

In this case study, there are 15 candidates having different application packages. In these 15 candidates, the two candidates, x₈ and x₉, are indiscernible by 7 condition attributes, which means both of them have the same condition attribute values but

different decision attribute values. It also means that the decision is inconsistent based on evaluation of the candidates by their condition attributes. For such a problem, Slowinski (1993) proposed a third category of decision, that is, those candidates who meet these criteria should be invited to an interview, which means that exterior information such as expert experience is needed to decide which category these objects fall into. In the case whereby no exterior information is available, the object cannot be categorized by the present system. For the sake of reducing the uncertainty and excavating as much information contained in the data set as possible, the SOM is applied to study the inherent relationships that exist between these objects. The same example is studied. The first case is chosen as a test object. The remaining 14 cases are chosen as training data sets and they are processed by the RST. There are 4 reducts generated by RST: $\{c_1, c_3\}$, $\{c_2, c_3, c_4, c_5\}$, $\{c_2, c_3, c_5, c_7\}$, $\{c_2, c_3, c_6\}$. The rules generated from reduct $\{c_1, c_3\}$ are presented in Table 5.2.

Table 5.2 Decision rules generated from reduct $\{c_1, c_3\}$

Rule #					Strength	
					Original	SOM
1	if $c_1=5$			then $d=A$	4	4.0
2	if		$c_3=5$	then $d=A$	5	5.0
3	if $c_1=4$	and	$c_3=4$	then $d=A$	1	1.0
4	if $c_1=4$	and	$c_3=4$	then $d=R$	1	0.5
5	if $c_1=3$			then $d=R$	3	3.0
6	if		$c_3=3$	then $d=R$	3	3.0

The *strength* for each rule is also included in Table 5.2. “Original” refers to the case which SOM is not applied to modify the support value of each object. The “SOM” re-determines the *strength* by cluster analysis. Now the first object x_1 is tested by these rules. It falls into situation (ii), in which it matches one non-deterministic rule (i.e. Rule #3 and #4). If the “Original” *strength* is used to determine whether this candidate is accepted or not, then no deterministic result can be obtained because there is a tie between them (both *strengths* for Rule #3 and #4 are equal to 1). At this point, the

SOM is used to analyze the relationship among these 14 objects from x_2 to x_{15} . They are categorized into 2 classes. After the application of SOM, the new class label are obtained and presented in the 2nd last column of Table 5.1 labeled as “SOM d”. It can be seen the object x_9 is categorized into class “*Accept*”. Because it is different from the original decision value “*Reject*”, this object is regarded as a “weak” object, whose support value is given a value of 0.5 (the value here can be any value less than 1). The new support value of each object is calculated according to following rules:

- If the original decision attribute is the same as the one generated by SOM, then this object is a “strong” object and its support is 1.
- If the original decision attribute is different from the one generated by SOM, then this object is a “weak” object and its support is 0.5.

From Table 5.1, it can be seen that except for x_9 , all the other objects are “strong”. Each of them contributes 1 support to the decision rules.

The re-calculated *strengths* for decision rules are presented in the last column in Table 5.2 labeled as “SOM”. Now, the object x_1 is tested by these rules with a new *strength*. It can be seen that x_1 can be classified into class “*Accept*” definitely (new *strength* for Rule #3 is 1 and Rule #4 is 0.5). This is the same as its original decision value. So far, the uncertainty has been reduced by applying SOM. From this case study, it can be seen although the RST can extract any inner information from the data set, it is still effected by the uncertainty contained in the data set. Using SOM allows one to make use of the “inner relationship” among the data sets to reduce the uncertainty. This is also applicable to situation (iv) in which case the test object matches more than one rule.

5.4 Experiment and Result Analysis

Following the previous analysis, a new system termed RoughSOM which combines the SOM and RST is generated. The flow chart of this new system is illustrated in Figure 5.2.

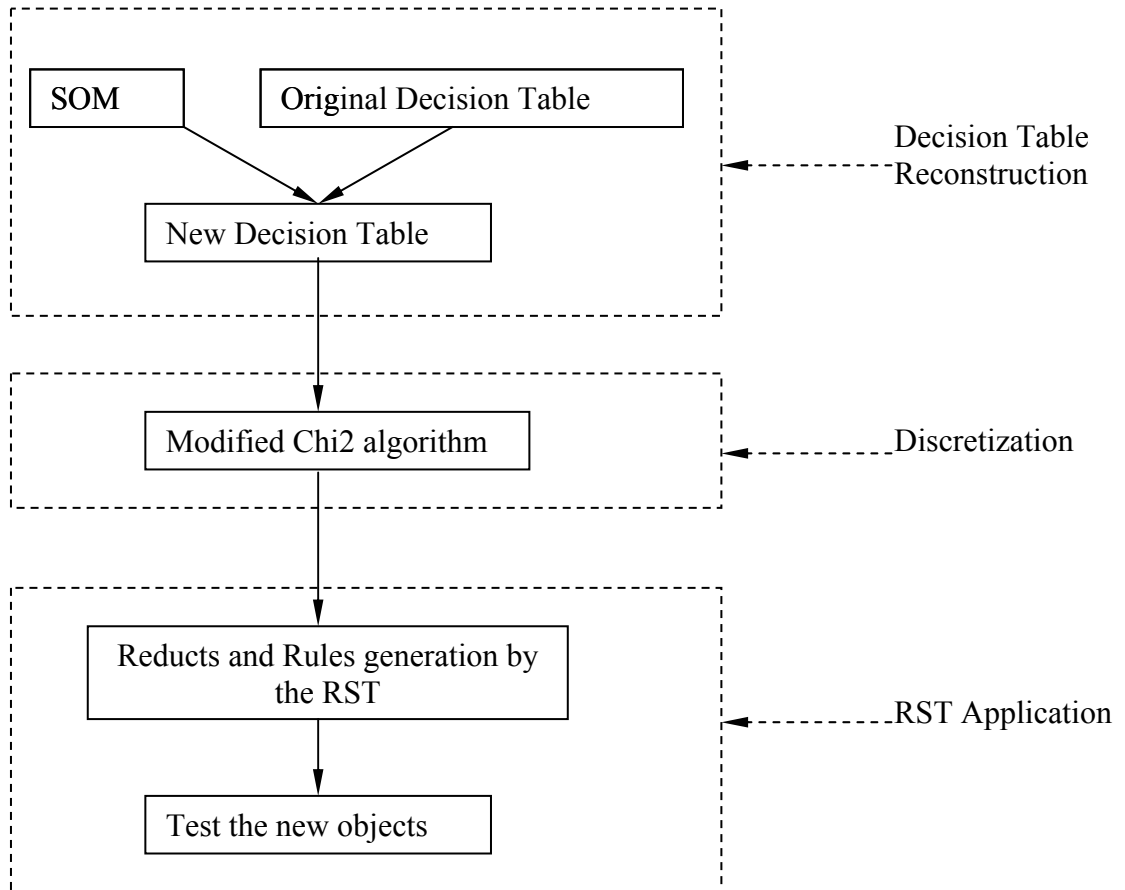


Fig 5.2 The flow chart of RoughSOM algorithm

This new algorithm is composed of three steps, namely, Decision Table Reconstruction, Discretization and RST application. In the Decision Table Reconstruction step, the original training data set is categorized using the SOM. The number of the output groups is constrained to the number of decision values in this project. Through the process on the training data set by SOM, a new decision value is obtained for every object. Now each object corresponds to 2 decision values. So far,

there are fewer references on solving this problem. In this project, the original decision table together with the new decision attribute obtained by SOM is then reconstructed according to the following rules:

- If the object's original decision value is the same as the one generated by SOM, then its decision value does not change.
- If the object's original decision value is different from the one generated by SOM, then this object is given a new decision value. For example, there are two original decision values, say 1 and 2, if original Decision=1, SOM Decision=2, then new Decision=3; if original Decision=2, SOM Decision=1, then new Decision=4.

The reason for this reconstruction is to maintain the *quality of approximation* of the new decision table as that of the original decision table.

The second step of this new algorithm is to discretize the continuous attributes in the reconstructed decision table for the subsequent RST processing. In this step, there are a number of well-established discretization methods, which have been discussed and compared in Chapter 4 (Dougherty et al. 1997; Chmielewski and Grzymala-Busse 1996; Fayyad and Irani 1993). The modified Chi2 algorithm is chosen as the discretization method in the RoughSOM system.

The last step is to apply the RST to generate rules and check the predictive accuracy on the test data sets. In this step, basic RST is used. In addition, the SOM, which is used to modify the support value of each object, is applied as well. In cases that several reducts are generated by the RST, the reduct with the minimal attribute number is selected. In addition, the reduct including the attribute with the highest *quality of*

approximation is selected in the case that there are several equal-length reducts. The reconstructed decision values from the final rules will be transferred back to original decision values. However, at this time, the “strong” rules have been distinguished from the “weak” rule with their *strengths*.

Table 5.3 Data sets information

Data Set	Name	Examples	Continuous attributes	Discrete attributes	classes
Australian credit card applications data set provided by quinlan@cs.su.oz.au	Australian	690	6	8	2
A liver disorder data set gathered by BUPA Medical Research Ltd, England	Bupa	345	6	0	2
Pima Indians Diabetes Database from National Institute of Diabetes and Digestive and Kidney Diseases	Diabetes	768	8	0	2
Medical data from the Cleveland Clinic Foundation	Cleveland	297	5	8	5
Heart disease dataset with 270 cases	Heart disease	270	5	8	2
Raw data on treatment of duodenal ulcer by HSV[10]	Hsv – r	122	9	2	4
The famous iris classification data by R. A. Fisher	Iris	150	4	0	3
Thyroid gland data donated by Stefan Aeberhard, Dept. of Comp. Science, James Cook University, Australia	New-thyroid	215	5	0	3
Vehicle silhouette data set from the Turing Institute, Glasgow, Scotland	Vehicle	846	18	0	4
Wine recognition data from Institute of Pharmaceutical and Food Analysis and Technologies, Italy	Wine	178	13	0	3

For convenience of comparing the RoughSOM with the original RST algorithm, the same 10 data sets are tested. The data contained real-life information from the medical and scientific fields which were used previously in testing pattern recognition and machine learning methods. Table 5.3 gives a summary of the data sets used in this experiment.

The data set *hsv-r* represents raw data on the treatment of duodenal ulcer by HSV. The remaining 9 data sets are taken from the University of California at Irvine repository of machine learning databases (Merz and Murphy). The ten-fold cross-validation test method (Weiss and Kulikowski 1990) is applied to all the data sets. The predictive accuracy obtained using both methods is presented in Table 5.4.

Table 5.4 The predictive accuracy (%) of original RST and RoughSOM

Data Set	Rough Set Theory	
	Original	SOM
Australian	78.84± 6.42	84.20± 4.40
Bupa	53.53± 9.16	53.82±14.74
Diabetes	70.26± 6.74	71.05± 6.54
Cleveland	55.86± 7.23	55.43± 6.98
Heart disease	79.26± 6.34	80.15± 4.43
Hsv – r	64.17±15.74	64.17±15.74
Iris	94.00± 7.34	96.00±10.52
New-thyroid	94.76± 9.38	95.90± 5.04
Vehicle	63.57± 4.53	63.13± 5.56
Wine	87.65±18.23	88.82±12.84
Average	74.19	75.27

To analyze the results obtained in Table 5.4, the Wilcoxon matched-pairs sign-rank test (Montgomery and Runger 1999) is applied. The result of applying the Wilcoxon matched-pairs sign-rank test is that the RoughSOM outperforms the original RST algorithm at 2.5% significance level for a one-tailed test. It can be seen that the new method is more accurate than the original RST algorithm.

For this new method, the decision table is discretized using the modified Chi2 algorithm in the second step. As a result, what are the differences in the discretization results between the original decision table and the reconstructed decision table since the class number has been changed? To answer the question, the fundamental idea of the χ^2 test is first introduced. The χ^2 test is used to test the hypothesis that the decision attribute is independent of the two adjacent intervals a condition attribute belongs. The

formula for computing the χ^2 value is expressed in Eq. (3.15). For convenience of clarity, it is presented here again:

$$\chi^2 = \sum_{i=1}^2 \sum_{j=1}^k \frac{(A_{ij} - E_{ij})^2}{E_{ij}}$$

where

k – class number of decision attribute within two adjacent intervals

A_{ij} – number of objects in the i th interval, j th class,

R_i – number of objects in the interval in the i th interval = $\sum_{j=1}^k A_{ij}$,

C_j – number of objects in the j th class = $\sum_{i=1}^2 A_{ij}$,

N – total number of objects,

E_{ij} – expected frequency of $A_{ij} = R_i * C_j / N$.

Since the class number of the reconstructed decision attribute is always greater than that of the original decision table, more candidate cuts should be checked by the threshold. It has been proven that the χ^2 value of the reconstructed decision table is greater than that of the original table (See Appendix A for a 2-class decision table) and the new value is larger than the threshold. Therefore, there are more cuts generated after discretization. This conclusion is supported by Table 5.5. For the 10 data sets, it can be seen that most of the interval numbers after discretization of the reconstructed decision table is greater than that of the original decision table. It can be interpreted that the objects with new class, also known as uncertain objects, are distinguished from the certain objects and they are granulized into a new group. In the subsequent RST processing, the rules generated from certain objects have stronger generalization and

consequently, the predictive accuracy is increased. The uncertain objects provide the supplementary help in classifying the test data set.

Table 5.5 The interval number after discretization of original decision table and reconstructed decision table

Data Set	Original	Reconstructed	Data Set	Original	Reconstructed
Australian	43.1± 26.97	18.5± 9.57	Bupa	3.4± 2.88	3.9± 2.96
	5.1± 1.19	11.3± 5.37		17.6± 7.92	13.5± 8.44
	3.4± 0.69	6.1± 3.95		11.0± 9.11	8.7± 5.49
	2.1± 0.31	2.9± 0.31		11.8± 4.60	7.9± 3.17
	5.5± 0.97	5.3± 1.56		28.2± 8.61	20.2± 9.93
	6.7± 1.94	9.8± 2.20		6.8± 1.30	4.9± 1.37
Cleveland	3.0± 0.94	6.5± 11.59	Heart disease	3.3± 1.49	2.1± 0.31
	4.1± 2.13	4.3± 10.43		2.0± 1.69	1.8± 0.78
	8.1± 2.84	55.7± 29.74		5.8± 2.78	4.6± 3.74
	3.6± 2.01	33.2± 27.72		4.1± 5.32	5.6± 2.71
	3.3± 0.67	2.7± 5.30		3.0± 0	3.8± 1.13
Vehicle	6.0± 0.66	10.3± 1.25	Hsv – r	1.0± 0	5.0± 3.68
	5.7± 0.67	9.1± 0.87		2.5± 1.08	6.8± 2.34
	11.8± 2.74	12.7± 1.25		2.2± 2.14	5.6± 5.91
	10.3± 1.76	13.3± 1.63		8.8± 4.58	4.5± 4.83
	6.0± 0.47	15.0± 0.94		4.1± 1.66	3.7± 5.69
	4.6± 1.07	7.1± 0.56		4.4± 3.74	5.5± 3.83
	8.9± 1.10	11.0± 1.24		5.7± 4.66	4.7± 3.59
	6.0± 0.94	9.5± 0.84		3.6± 2.87	4.7± 5.96
	4.8± 0.63	7.6± 0.69		11.6± 5.14	4.8± 3.93
	6.0± 0.94	11.4± 1.07	Iris	4.3± 1.33	7.7± 5.05
	9.2± 1.03	12.4± 0.69		6.4± 3.47	4.3± 1.41
	18.1± 8.69	14.7± 2.49		3.9± 0.31	5.0± 1.24
	6.6± 1.50	17.1± 1.52	New-thyroid	3.9± 0.31	5.1± 0.87
	6.2± 1.47	12.4± 0.84		5.3± 1.05	8.0± 3.49
	2.9± 0.56	14.0± 0.81		5.2± 0.63	9.7± 5.55
	3.2± 0.78	23.2± 0.91		4.9± 0.99	7.9± 3.75
	5.4± 0.69	13.6± 0.96		3.4± 1.26	7.0± 2.10
	6.1± 0.73	14.1± 1.44		6.0± 1.24	12.3± 5.61
			Diabetes	3.1± 0.99	4.8± 0.91
Wine	5.4± 1.07	3.8± 1.13		5.5± 1.64	8.5± 3.62
	5.1± 1.52	5.2± 1.54		3.7± 0.48	3.6± 2.63
	2.8± 1.03	2.3± 0.67		4.8± 2.52	3.4± 0.69
	3.0± 0.47	3.7± 1.33		6.4± 2.27	8.7± 5.18
	3.0± 0.47	2.9± 0.31		17.3± 9.93	19.0± 7.98
	3.5± 1.26	5.4± 1.34		77.1± 33.25	41.6± 25.66
	7.8± 1.47	8.0± 2.21		5.0± 2.53	5.9± 1.59
	2.0± 0	3.0± 0.94			
	4.0± 1.69	4.9± 1.52			
	4.8± 1.31	5.7± 1.56			
	4.8± 0.42	3.5± 0.70			
	5.4± 1.57	4.7± 1.33			
	5.1± 1.19	4.9± 0.87			

However, for a data set with four or more classes, such as the Cleveland Heart Disease, Hsv-r and Vehicle data sets, the predictive accuracy of RoughSOM is not as good as

the RST results. The reason is partly because of the uneven distribution of the data sets. The distribution of 10 data sets is given in Table 5.6. For example, for the Cleveland Heart Disease data set, the objects belonging to the first class are much greater than the other 4 classes. This may cause skewness in the reconstructed decision table. This problem will be studied in the future.

Table 5.6 The distribution of data sets

Data Set	Distribution		Class			
			Data number			
Australian	1	0				
	307	383				
Bupa	1	2				
	145	200				
Diabetes	1	2				
	500	268				
Cleveland	0	1	2	3	4	
	164	55	36	35	13	
Heart disease	1	2				
	150	120				
Hsv – r	1	2	3	4		
	81	19	8	14		
Iris	1	2	3			
	50	50	50			
New-thyroid	1	2	3			
	150	35	30			
Vehicle	1	2	3	4		
	212	217	218	199		
Wine	1	2	3			
	59	71	48			

5.5 Summary

In classification a new object by matching its description to decision rules, there is a situation that has been overlooked. This is the case whereby the *rule strengths* of both classes are the same and hence there is no clear indication of which class this object belongs to. This is caused by uncertainty and imprecision contained in the data. Although the RST is a powerful tool in dealing with granularity of information and has

no requirement of exterior information, it ignores the inner relationships of the data. Therefore, a cluster analysis method – SOM (Self-Organizing Map) – is applied to determine these inner relationships so as to increase the classification accuracy of the RST. Following this idea, a new system termed RoughSOM which combines the RST and SOM is proposed. Through experiments carried out on 10 data sets, it has been shown that this new method removes the uncertainty from the system and increases the predictive accuracy on the test data sets. However, for a data set with four or more classes, including the Cleveland Heart Disease, Hsv-r and Vehicle data sets, the predictive accuracy is not as good as the RST results. The reason is partly because of the uneven distribution of the data sets. This problem will be studied in the future.

In the next chapter, the RoughSOM system will be applied to solve the temporal rule discovery problem. As it can be seen until now, the data sets applied in the experiments are time independent, the adjacent two objects can be exchanged without affecting the final results. However, for a time series, every object is related to a temporal factor. How can one apply the RoughSOM to discover knowledge from time series? This will be addressed in the next chapter.

Chapter 6

Time Series Forecasting using Rough Set Theory

6.1 Introduction

In previous chapters, it can be seen that the data sets applied to the RST are temporal independent, which means the data set can be shuffled without affecting the final results. However, for the time series, such as the medical history of the patients or historical data of a stock, how can the RST be applied to extract information from them? In this chapter, problems concerning time series forecasting using the RST are discussed.

6.2 Temporal Rule Discovery Problem

The temporal rule discovery problem involves the application of knowledge discovery techniques to identify temporal rules from time related series (Golan and Edwards, 1993). Here the temporal rules refer to the discovered relationships which reflect common repetitive patterns occurring in the data. As stated in Golan and Edwards (1993), these temporal rules are strong rules having a higher supporting *strength*, which means a lot of cases support them. They are not necessarily precise or deterministic, but can be used to predict the outcome with a higher probability. Discovering temporal rule relationships will also support the relationships discovered

with the data in current time. Since the temporal rule discovery problem adds one more factor to the Information System, and this factor is very important to determine the sequence of the objects, there is a need for modifications to the Information System. In the following section, the Temporal Information System is presented.

6.3 Temporal Information System (TIS)

Bjorvand (1996) presented the definition of the Temporal Information Table as follows:

Temporal Information System (TIS) $A_t = (U, A \cup \{d, t\}, \prec)$

U – objects (cases, states, patients, observations, ...)

A – features, variables, characteristic conditions, ...

d – the decision attribute : $d \notin A$

t – the sequence attribute: $t \notin A$

\prec is an ordering relation on the sequence attribute, t .

$\prec = \{(x, y) : x, y \in N \text{ and } x < y\}$

It can be seen that the TIS defined here is slightly different from the definition of Information System introduced in Chapter 3 in that TIS includes the temporal factor to describe the sequence of the objects.

The traditional RST cannot be applied to a Temporal Information System because the RST does not take the ordering relationship into consideration when it extracts information from the data. Therefore, the TIS should be converted to the Information System before the application of the RST. Defining a limit as to how far back

dependencies/sequences should be traced, the TIS can be transformed into an Information System.

6.4 Converting Time Series to Rough Set Objects

There is a difference between the time series with and without real-time constraints. Without real-time constraints, the only thing that matters is the chronology of events – this is also the case when the time between each event is a constant. However, with real-time constraints, the interval between each event should also be taken into consideration. Obviously, these intervals vary. When applying the RST, these time series cannot be input directly. Objects suitable for composing the decision table for RST should be constructed from these time series. Baltzersen (1996) presented two methods concerning this problem in his thesis. They are introduced in the following sections.

6.4.1 The mobile window

Baltzersen (1996) presented a practical method in his thesis. His algorithm for converting predicted objects to time series is listed as follows:

Step 1: Choose a starting point in the time series.

Step 2: Generate a rough set object from the attributes appearing at the ‘markings’.

Here ‘markings’ refers to the methods to convert time series to condition attributes and decision attribute.

Step 3: Move 1 step forward in time.

Step 4: If the window has started moving outside the time series range, then stop. If not, go to step 2.

Consider the following example of time-series pictured as rows:

1	2	3	5	8	13	21	34
34	21	13	8	5	3	2	1

A moving window 2row×5columns is chosen. It starts at the beginning of the time series, covering the matrix:

1	2	3	5	8
34	21	13	8	5

These values covered by the matrix are denoted:

$\bar{t}_{a,0}$	$\bar{t}_{a,1}$	$\bar{t}_{a,2}$	$\bar{t}_{a,3}$	$\bar{t}_{a,4}$
$\bar{t}_{b,0}$	$\bar{t}_{b,1}$	$\bar{t}_{b,2}$	$\bar{t}_{b,3}$	$\bar{t}_{b,4}$

The “markings” are defined as:

$$\begin{aligned} \text{Condition attribute 1 (CA1): } & \frac{\bar{t}_{b,1} - \bar{t}_{b,0}}{\bar{t}_{b,0}} \\ \text{Condition attribute 2 (CA2): } & \frac{\bar{t}_{b,2} - \bar{t}_{b,1}}{\bar{t}_{b,1}} \\ \text{Condition attribute 3 (CA3): } & \bar{t}_{a,0} \\ \text{Decision attribute (DA): } & \frac{\bar{t}_{a,4} - \bar{t}_{a,3}}{\bar{t}_{a,3}} \end{aligned}$$

The first object after transformation has following attributes:

CA1	CA2	CA3	DA
-0.38	-0.38	1	0.6

The window moves on in the next step, and flowing matrix appears in the window

2	3	5	8	13
21	13	8	5	3

A second object is generated, where three condition attributes and decision attribute are approximately -0.38, -0.38, 2 and 0.62, respectively. This process is repeated until the window reaches the end of time series.

This algorithm is a practical method of creating rough set objects from time series. It presents snapshots of the objects when the window moves. In his method, Baltzersen (1996) recommended that length of the window should be chosen based on the generalization of the Markov property, which means the current value is only dependent on previous values by a limited number. The process of discretizing and methods dealing with null values and other concepts could be integrated in this window algorithm as well.

6.4.2 “columnizing”

This method was first introduced in Synak (1995). He proposed that the time series be organized in columns, such that each row is an object with many attributes where each attribute is an indicator, and each row represents a different point in time. This is the way in which the vibration signals are converted to decision table using both frequency and time domain indicator described in Chapter 3. The vibration signals are collected based on the sampling frequency. One vibration signal records the working condition within a certain period. Based on Synak’s idea the concept of “columnizing” is presented in the following.

A “columnizing” of time-series consists of the following:

Step 1: Creating initial columns: The different indicators generated from time series are converted to adjacent columns.

Step 2: Synchronizing: The columns are shifted so that in each row, the same point of time is represented. For a stock’s price, moving average indicators may be

calculated using different time intervals. They can be synchronized by shifting one column against the other according to the time they are benchmarked.

Step 3: Dealing with null values: Appropriate methods for dealing with null values/mismatching frequencies are applied.

Step 4: Adding time attributes: Attributes identifying the time at which the events occur may be added. This attribute can be the decision attribute. For a stock price, trading signals can be added as time attribute.

Step 5: Creating derived series: Composing the decision table for the RST.

It can be seen that this algorithm is more straightforward compared to the “mobile window” method since the determination of the “markings” is difficult for us without expert’s experience. In addition, the choice of the window length with Markov property is a time-consuming problem. So in the following experiment, the “columnizing” method will be adopted.

Besides the above two methods, Bjorvand (1996) in his thesis formalized the algorithm proposed in Golan and Edwards (1993) to transfer a TIS to an Information System. It is a simple algorithm and the interested reader can refer to his thesis for the details.

6.5 Financial Market Prediction

As the financial market involves tremendous financial stake, a lot of effort has been made to predict economical development.

The problem addressed in this section is with regards to the stock prediction for use by investors. More specifically the stock market's movements will be analyzed and predicted. Knowledge which would guide investors on when to buy and sell is retrieved.

The most widely used method in economic forecasting is fundamental and technical analysis. Fundamental analysis involves the prediction of a economic time series from other factors. For instance, predicting the stock price for an individual company may involve the utilization of figures from the company's financial status. In contrast, technical analysis does not involve the analysis of income statements, balance sheets, company policies, or anything fundamental about the company. It only considers the actual history of trading and price in a security or index. The theory of technical analysis is based on the assumption that the market price reflects all known information about the individual security.

Technical analysis is mainly concerned with market indicators. These technical indicators look at the trend of price indexes and individual securities. They evaluate the position of the security or index. The theory underlying these indicators is that once a trend is in motion it will continue in that direction (Achelis, 1995). Technical analysis attempts to determine the strength of the trend and the direction of the trend. The technical analyst will use his analysis to stay away from a market or a security unless there is a good amount of protection in place.

Considering the data available in this project, technical analysis will be adopted to implement the stock market forecasting. Some case studies using the RST to forecast

the stock market have been presented in Chapter 2. As for the conventional techniques and other knowledge based approaches, such as Neural Networks, Genetic Algorithm and Fuzzy Logic, the interested reader can refer to Holden et al. (1990), Freisleben (1992) and Hiemstra (1994). In the following, the utilization of the RST to forecast the financial market is focused.

6.6 Indicators Study

Before the application of the RST, the Temporal Information System composed of the historical price chart should be converted to rough set objects. Following the “columnizing” method introduced in the previous section, the first step is to create the initial columns. As the tools of technical analysis are indicators and systems which are based on price charts, each column will correspond to an indicator which should be able to reflect market changes. In the following, the work carried out on indicators’ studies is presented.

6.6.1 Market trend

One of the basic tenets put forth by Charles Dow in the Dow Theory (Bishop, 1960) is that security prices do follow trends. Trends are often measured and identified by "trendlines" and they represent the consistent change in prices (i.e. a change in investor expectations). In Figures 6.1 and 6.2, rising trend and falling trend are illustrated.

As shown in Figure 6.1, a rising trend is defined by successively higher low-prices. A rising trend can be thought of as a rising support level - the “bulls” are in control and

are pushing prices higher. Figure 6.2 shows a falling trend. A falling trend is defined by successively lower high-prices. A falling trend can be thought of as a falling resistance level - the “bears” are in control and are pushing prices lower.

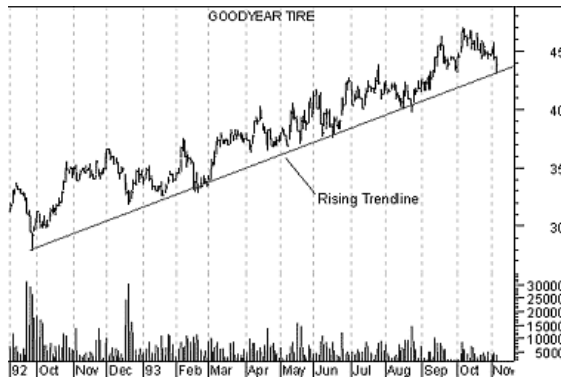


Fig. 6.1 Rising trend



Fig. 6.2 Falling trend

A principle of technical analysis is that once a trend has been formed, it will remain intact until broken (Achelis, 1995). The goal of technical analysis is to analyze the current trend using trendlines and then either invest with the current trend until the trendline is broken, or wait for the trendline to be broken and then invest with the new (opposite) trend. Hence, for practical trading, to know the current market state, either in the rising trend or in the falling trend, is very important. In the following sections, work is focused on searching for indicators that can reflect the fluctuation of price in the financial market.

6.6.2 WARS – Weighted Accumulated Reconstruction Series

In this project, an indicator, called *Weighted Accumulated Reconstruction Series* (*WARS*), has been constructed and found to have interesting characteristics. This indicator can reflect the trend in changes of the price. In addition, it makes use of more

information contained in the data than the *moving average* and therefore may be able to better reflect the state of the price or index.

6.6.2.1 formulation of WARS

The idea of generating *Weighted Accumulated Reconstruction Series (WARS)* comes from the computation of Entropy. The concept of Entropy was first proposed by Shannon (1948) in the Information Theory as a measure of the complexity of a system. Up to now, this concept has been applied in the economic domain to measure production flexibility (Frezelle and Woodcock, 1995), customer requirements (Johnston, 1996), and processing cost of administrating the production facility (Ronen and Karp, 1994). By building models based on the Entropy measurement, the weakness of system is found and the system is optimized. In the capital market, a derivative of information entropy - Kolmogorov Entropy - is applied to measure how chaotic a system is based on the analysis of real-time price or index (Barkoulas and Travlos, 1998; Mayfield and Mizrach, 1992; Frank and Stengos, 1989). By calculating the Kolmogorov Entropy, the predictability of the price changes or returns is studied. Kapur and Kesavan (1992) even use Kullback's Minimum Cross-Entropy Principle to minimize the risk in portfolio analysis.

The Shannon Entropy, represented by $Ent(S)$, of a system is defined as:

$$Ent(S) = - \sum_{i=1}^k P(C_i, S) \log(P(C_i, S)) \quad (6.1)$$

where C_i presents the i th event in system S , $i = 1, 2, \dots, k$;

$P(C_i, S)$ is the *a priori* probability of event C_i 's occurrence in S .

This concept can be used as a measure of uncertainty. The higher the uncertainty in the system, the higher the entropy and more information is required to understand what is happening in it. In the capital market, risk is inextricably associated with the concept of uncertainty. By maximizing expected return and simultaneously minimizing risk, one can obtain efficient portfolio.

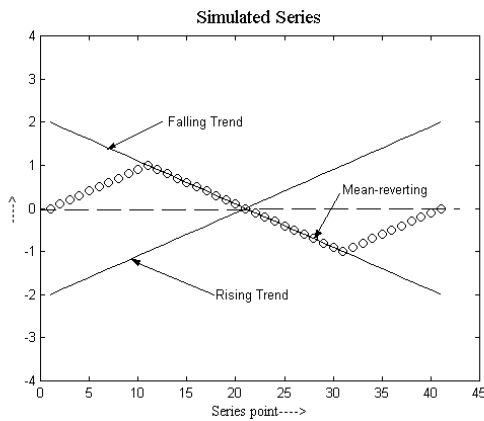


Fig 6.3 The illustration of accumulated reconstruction series – original series

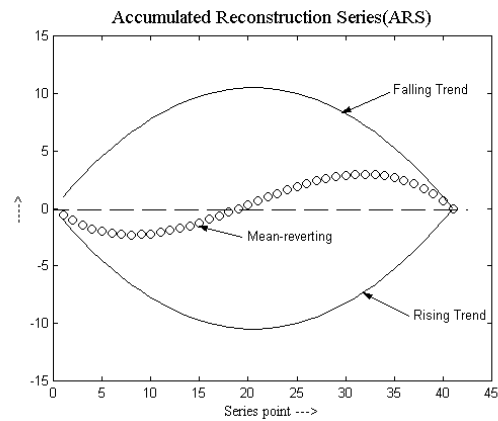


Fig. 6.4 The illustration of accumulated reconstruction series – transformed series

From the definition in Eq. 6.1, the distribution of the system must be known before the system entropy can be calculated. However, in practice, the distribution of the system is usually not known in advance. The easiest way to solve this problem is to accumulate this series and it will follow the exponential function for a positive series. Following this idea, the algorithm to construct the new indicator is formulated. Consider a series with a trend as shown in Figure 6.3. First, it is normalized, that is, the first point and mean value are subtracted. Next, consecutive data are aggregated to obtain another series. The final result will either be all greater than or less than zero, corresponding to down-trending series or up-trending series respectively. In the case of a mean-reverting series, after the above procedure is performed, a curve that fluctuates

around the zero line, will be obtained. A summary of the results is shown in Figures 6.3 and 6.4.

In Figure 6.3, three curves representing up-trending, down-trending and mean-reverting series are shown. After accumulated reconstruction, the corresponding three curves are drawn in Figure 6.4.

Now the algorithm to generate *WARS* is as follows:

Step 1: Normalize every value of this series between -1 to 1 (to remove amplitude effect from the series)

$$x_i = x_i / \max(|x_i|); \quad (i = 1, 2, \dots, \text{Win_length}) \quad (6.2)$$

Step 2: Subtract the first value of a series (to ensure that all the intervals have the same origin).

$$x_i = x_i - x_1; \quad (i = 1, 2, \dots, \text{Win_length}) \quad (6.3)$$

Step 3: Calculate the mean value of this series.

$$\text{Mean_}x = \frac{1}{n} \sum_i x_i; \quad (6.4)$$

Step 4: Subtract the mean value from the whole series.

$$x_i = x_i - \text{Mean_}x; \quad (i = 1, 2, \dots, \text{Win_length}) \quad (6.5)$$

Step 5: Reconstruct a new series (*WARS*) by means of weighted accumulation of the original one.

$$Weight_j = \frac{1 + 2 + \dots + j}{1 + 2 + \dots + Win_length} \quad j = 1, 2, \dots, Win_length \quad (6.6)$$

$$\begin{aligned} y_1 &= \frac{1}{1 + 2 + \dots + n} x_1; \\ y_2 &= \frac{1}{1 + 2 + \dots + n} x_1 + \frac{1 + 2}{1 + 2 + \dots + n} x_2; \\ &\vdots \\ y_n &= \frac{1}{1 + 2 + \dots + n} x_1 + \frac{1 + 2}{1 + 2 + \dots + n} x_2 + \dots + \frac{1 + 2 + \dots + n}{1 + 2 + \dots + n} x_n; \end{aligned} \quad (6.7)$$

In this reconstruction process, the more recent points have more contribution to *WARS*.

Step 6: Calculate the area of this interval and obtain its absolute value.

$$Area = |y_1 + y_2 + \dots + y_n| \quad (6.8)$$

In this step, the absolute value is calculated. The trending and mean-reverting states are distinguished according to the area value. If the area value is greater than 0, then the market is in a trending state, else the market is in a mean-reverting state.

The above algorithm is used to calculate the *area* of every interval. During the process of generating *WARS*, the original series is rolled and the area value of every interval is calculated iteratively. For instance, if there are 10 points in a series and *Win_length* is chosen as 4, then from the first to the fourth point, the first area value is calculated. From the second to the fifth point, the second area value is obtained. This process is

repeated and eventually six points are obtained to construct *WARS*. The length of *WARS* equals to the length of original series less *Win_length*. The selection of *Win_length* will be discussed in the later section.

6.6.2.2 comparison of *WARS* and daily profit curves

For a large company, a certain strategy will be usually adopted to direct its operation in the financial market. The *equity curve* of a period will normally be used to evaluate the pros and cons of the adopted strategy (Hampton, 1998). If the *equity curve* goes up, the company is making a profit and vice versa. The generation of the *equity curve* will not be described here. It is taken to be a given information. After constructing *WARS* based on historical data, *WARS* and some of the *equity curves* of futures are compared.

For the testing of the new indicator, 15 historical futures data supplied by Man-Drapeau Research Pte Ltd were selected. The *WARS* was generated from the *daily close price*. For the convenience of comparison of the above two curves, the *equity curve* was changed to *daily profit curve* by using the following method:

$$y_i = x_i - x_{i-1}; \quad (i = 1, 2, \dots, n) \quad (6.10)$$

Then the *moving average curve* (Webster, 1995) was calculated for both *WARS* and *Daily Profit Curve*. The procedure of calculating the *moving average* was introduced as follows:

Step 1: Select a value for the parameter, *Moving Average Interval* (*Win_len2*);

Step 2: From the origin, the mean value within Interval-*Win_len2* is calculated iteratively for the whole series.

$$y_i = \frac{x_i + x_{i+1} + \dots + x_{i+Win_len2-1}}{Win_len2}; \quad i = 1, 2, \dots, Win_len2 \quad (6.11)$$

In the above procedure, parameter - *Win_len2* - must be determined and its selection will be discussed in the later part of this section.

After the above transformation, the correlation coefficient of *WARS* and the *Daily Profit Curve* was calculated. The correlation coefficient was used to evaluate the degree of their similarity. The definition of the correlation coefficient of vector ξ_1 and ξ_2 is:

$$corrcoef = \frac{cov(\xi_1, \xi_2)}{\sqrt{D\xi_1} \cdot \sqrt{D\xi_2}}; \quad (6.12)$$

where $cov(\xi_1, \xi_2)$ is the covariance matrix of vector ξ_1 and ξ_2 .

D represents the variance of vector ξ .

In Figures 6.5 – 6.8, two curves are rescaled between -1 and 1 so that they can be compared directly. The solid line curves in Figures 6.5 – 6.8 represent *WARS* while the dashed curves represent the *Daily Profit Curve*. In these figures, the two curves, *WARS* and *Daily Profit Curve* are found to have similar shapes. It is clear that *WARS* reflected the fluctuation of *Daily Profit Curve*. It goes up when the *Daily Profit Curve* goes up and it goes down when the *Daily Profit Curve* is decreasing. From the correlation coefficient, the value is always larger than 0.6 (sometimes almost equal to 0.95). Thus *WARS* reflects the changing of *Daily Profit Curve* quite well.

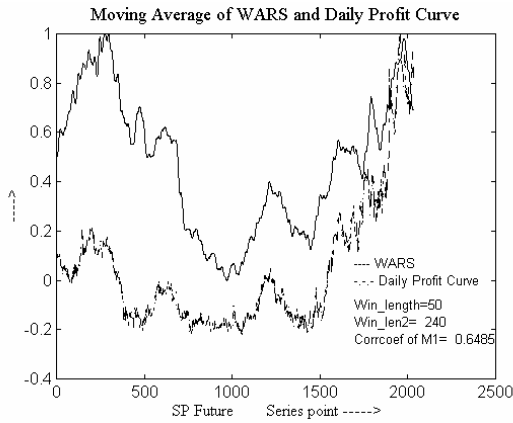


Fig. 6.5 The comparison of *WARS* and *Daily Profit Curve* for S&P 500 Futures

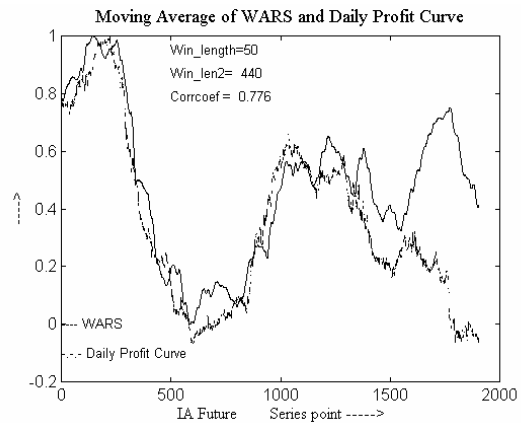


Fig. 6.6 The comparison of *WARS* and *Daily Profit Curve* for IA Futures

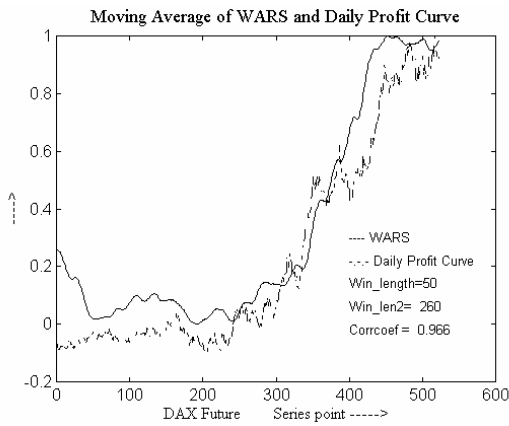


Fig. 6.7 The comparison of *WARS* and *Daily Profit Curve* for DAX Futures

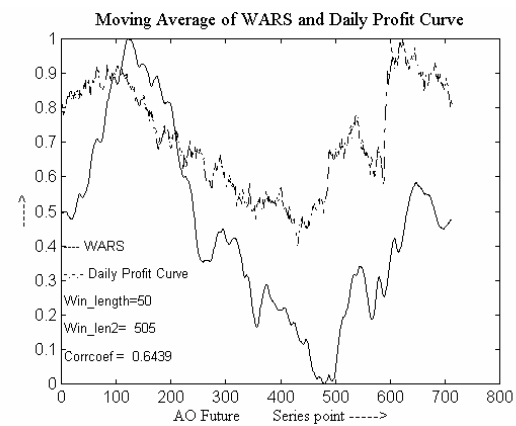


Fig. 6.8 The comparison of *WARS* and *Daily Profit Curve* for AO Futures

From the generation of *WARS*, its meaning in the financial market can be interpreted as follows: *WARS* is generated by using the closing price in a period. If within this period, the price changes in a trending way, either up-trending or down-trending, *WARS* will maintain its large value or it may increase. When the price fluctuates in a mean-reverting way, *WARS* will decrease or remain as a small value. As a whole, *WARS* reflects the state of a market. Indirectly, it continuously reflects the changing of prices. From the view of entropy, it can be simply interpreted as follows. The market states can be represented as 1 for up-trending, 0 for mean-reverting and -1 for down-trending. When the market falls in the trending state (either 1 or -1), the entropy of the

market is 0, corresponding to large value of *WARS*, close to 1. The mean-reverting state (0) is composed of up-trending and down-trending states, when the entropy of market equals to $\log 2$ ($Ent = -(\frac{1}{2} \log \frac{1}{2} + \frac{1}{2} \log \frac{1}{2})$), corresponding to a small value of *WARS*, close to 0. It is easy to understand the above results. When the market moves in a trending way, the system is more certain than in a mean-reverting state, in which the direction of price cannot be determined, that is, more uncertainties are contained in this system.

Although *WARS* is very similar in behavior to the *Daily Profit Curve*, there are still many factors affecting the final results, such as *Win_length* and *Moving Average Interval* (*Win_len2*). These issues will be discussed in next section.

6.6.2.3 some issues in generating *WARS*

From the point of view of the investors, they want the indicator to lead the *equity curve* so that they can determine how to modify their strategy. In Figures 6.5 – 6.8, some parts of *WARS* lead the *Daily Profit Curve* while other parts lag the *Daily Profit Curve*. What are the factors causing this behavior? How to make *WARS* consistently lead the *Daily Profit Curve*? In this section, some parameters used in construction of *WARS* are discussed.

There are two parameters used to construct indicator *WARS*. *Win_length* affects *WARS* generation and *Win_len2* is a parameter for calculating *moving average*. Both parameters affect the correlation coefficient between *WARS* and *Daily Profit Curve*. The accumulating procedure of *moving average* has the effect of smoothening the original curve. Inevitably it will remove some useful information from the original

system. Worse yet, moving average makes the transformed curve always lagging the original series. This is not desirable for the traders. Hence, *Win_len2* is the most important factor to be determined in this process. The larger it is, the more information it loses. Therefore the challenge is to determine the parameter, *Moving Average Interval*, to determine the largest correlation coefficient with the least information loss. This is an optimization problem. Many artificial intelligence methods such as Genetic Algorithms or Neural Network can be used to solve it (Goonatilake and Treleaven, 1995). In practice, the values used usually recommended by experts. As for the other parameter, *Win_length*, different values are selected to test its effect on the correlation coefficient. In Figures 6.9 – 6.12, four futures are studied in this project. Every curve in each figure represents the correlation coefficient curve calculated with a certain *Win_length* value and different *Moving Average Interval*. The symbol 'o' represents the maximum correlation coefficient value corresponding to a certain *Win_length* value. The upper curve represents a larger *Win_length*. From these four figures, for a certain futures data, whatever *Win_length* value is selected, the maximum correlation coefficient value will fall into a small interval on the *Win_len2* axis.

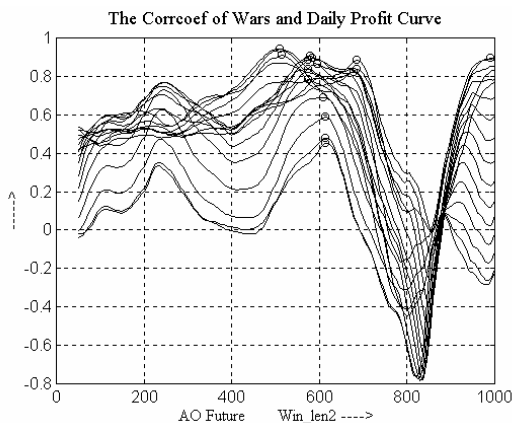


Fig. 6.9 Correlation coefficient of *WARS* and *Daily Profit Curve* in different *Win_length* and *Win_len2* for AO Futures

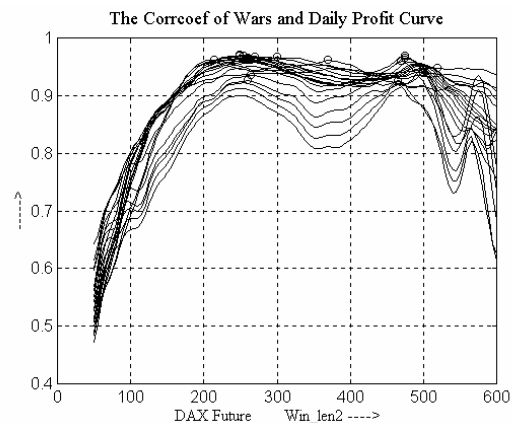


Fig. 6.10 Correlation coefficient of *WARS* and *Daily Profit Curve* in different *Win_length* and *Win_len2* for DAX Futures

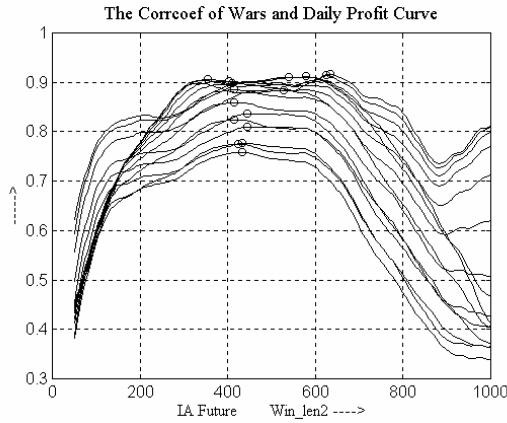


Fig. 6.11 Correlation coefficient of *WARS* and *Daily Profit Curve* in different *Win_length* and *Win_len2* for IA Futures

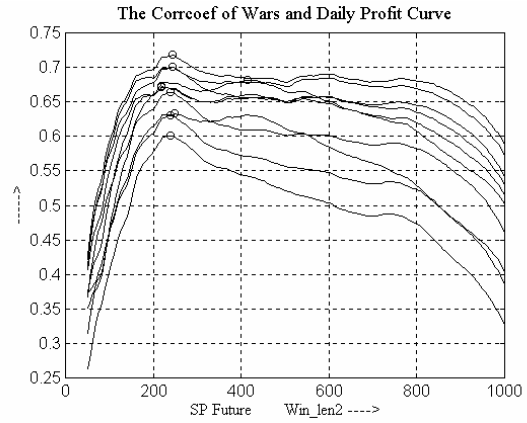


Fig. 6.12 Correlation coefficient of *WARS* and *Daily Profit Curve* in different *Win_length* and *Win_len2* for S&P 500 Futures

For instance, for DAX futures, from *Win_len2* values of 200 to 500, the correlation coefficient remains high with respect to any *Win_length* value. Therefore, *WARS* is very robust as far as parameters selection goes. Other futures also exhibit the same robust behaviour.

However, it is worth noticing that corresponding to the maximum correlation coefficient, *Win_len2* cannot be chosen randomly. This is because to obtain the maximum correlation, *Win_len2* is determined when the two curves change simultaneously. For practical purposes, it will be more useful if *WARS* leads the Daily Profit Curve. A smaller value or larger value must be chosen so as to ensure that *WARS* leads the Daily Profit Curve.

6.6.2.4 using *WARS* to generate trading system

Based on the previous analysis, in this section, *WARS* is used to generate a trading system. The trading system (Pardo, 1992) is built with the following steps:

- Calculate *WARS* using historical data.

- Determine the threshold value for buying and selling action according to the value of $WARS$ calculated using training data set.
- Generate the trading signals as follows:

If value of $WARS >$ threshold of buying, then buy at the next day's opening price

If value of $WARS <$ threshold of selling, then sell at the next day's opening price

If value of $WARS$ is in between the thresholds, then no action is taken, that is, hold.

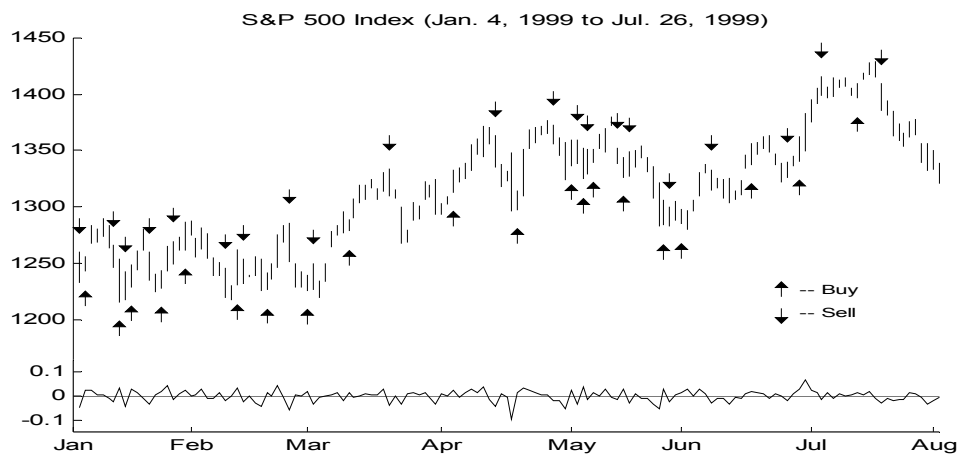


Fig 6.13 Trading system based on $WARS$

Figure 6.13 depicts such a trading system using S&P500 index. The historical data S&P 500 here is provided by Man-Drapeau Pte. Ltd (Singapore). The time interval of data is 15-minute-bar. The $WARS$ is calculated based on 1-day-Win_length.

From Figure 6.13, it can be seen that $WARS$ gives less trading signals when the market changes gradually, from March to May 1999. But when the market changes dramatically, from January to March, more signals are generated. This indicates that $WARS$ is a robust indicator, which can adapt itself to market changes. This is favorable to investors. The trading performance is illustrated in Table 6.1. Compared to buy-hold

strategy (will be described in Chapter 7), which generate a net profit of \$163.00 from January to August 1999, it performs better.

Table 6.1 Trading system performance based on the indicator *WARS* on S&P 500 futures

Training Data Set	Testing Data Set
Training period 01/04/1988 - 12/31/1998	Testing period 01/04/1999 - 08/12/1999
max_ <i>WARS</i> _area = 0.1606	Net_profit = 324.599854
min_ <i>WARS</i> _area = -0.1848	max_win = 70.500000
threshold_buy = 0.0149	max_loss = -47.199951
threshold_sell = -0.0129	Trading_number = 40
Mean_ <i>WARS</i> _area = 0.0007	Winning_Trade = 29
Std_ <i>WARS</i> _area = 0.0191	Sharpe_ratio = 0.318864

There are altogether 40 trades in this index, among which 29 of them are profitable. From these results, it can be seen that this new indicator is effective in differentiating the market states and so it can be used to trace the changing market and provide the trading signals.

6.6.2.5 remarks

In comparison with the *Daily Profit Curve*, *WARS* can indicate the *Daily Profit Curve* accurately and easily. Through the comparison of these two curves, the following results are presented:

- *WARS* can indicate the behavior of *Daily Profit Curve* if two parameters, namely, *Win_length* and *Moving Average Interval (Win_len2)* are chosen properly.
- Several issues in the research have been investigated. As for the parameter selection, optimization methods such as Genetic Algorithm (Beasley et al., 1993) and Neural Networks (Cichocki and Unbehauen, 1993) can be used. As for weight selection, although some research has been carved out, it is felt that the experience of the expert is necessary for its determination.

- In addition, *WARS* can be used to build a trading system. Through its application in S&P 500 index, it can be seen that this indicator is effective and promising.
- *WARS* can be used to reflect the uncertainty of the market. When the magnitude of *WARS* approaches 1, the market is a strong trending state and therefore more investment can be carried out.

In the following sections, 6 well-established indicators are described. Because of their wide application in the technical analysis, brief introduction are presented.

6.6.3 MACD – Moving Average Convergence/Divergence

The MACD ("Moving Average Convergence/Divergence") is a trend following momentum indicator that shows the relationship between two moving averages of prices. The MACD is the difference between a 26-day and 12-day exponential moving average. A 9-day exponential moving average called the "signal" (or "trigger") line is plotted on top of the MACD to show buy/sell opportunities. The MACD proves most effective in wide-swinging trading markets. Figure 6.14 shows Whirlpool and its MACD.

In Figure 6.14, the "buy" arrows and "sell" arrows are drawn when the MACD rose above and fell below its signal line, respectively. Figure 6.14 shows that the MACD is truly a trend following indicator - sacrificing early signals in exchange for keeping one on the right side of the market. When a significant trend has been developed, such as in October 1993 and February 1994, the MACD was able to capture the majority of the

move. When the trend was short lived, such as in January 1993, the MACD proved unprofitable.



Fig. 6.14 MACD Indicator

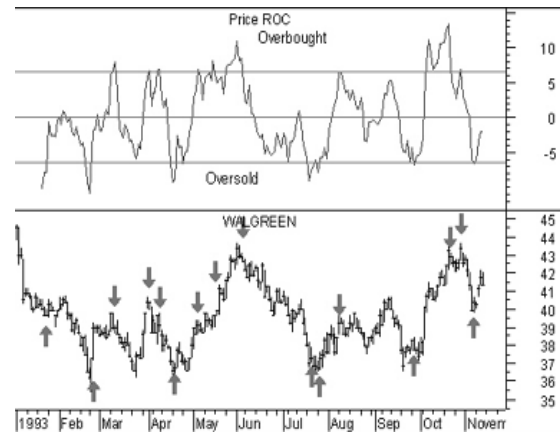


Fig. 6.15 ROC Indicator

6.6.4 ROC – Price Rate of Change

The ROC has a wave-like motion form and it measures the amount that prices have changed over a given time period. As prices increase, the ROC rises; as prices fall, the ROC falls. The greater the change in prices, the greater the change in the ROC.

The time period used to calculate the ROC may range from 1-day (which results in a volatile chart showing the daily price change) to 200-days (or longer). The most popular time periods are the 12- and 25-day ROC for short to intermediate-term trading. These time periods were popularized by Gerald Appel and Fred Hirschler in their book, *Stock Market Trading Systems* (1980). Figure 6.15 shows the 12-day ROC of Walgreen expressed in percentage form.

The "buy" arrows are drawn each time the ROC fell below, and then rose above, the oversold level of -6.5. The "sell" arrows are drawn each time the ROC rose above, and then fell below, the overbought level of +6.5. Here the optimum overbought/oversold levels (e.g. ± 6.5) vary depending on the security being analyzed and overall market conditions.

6.6.5 Stochastic Oscillator

The Stochastic Oscillator is displayed as two lines. The main line is called "%K." The second line, called "%D," is a moving average of %K. The %K line is usually displayed as a solid line and the %D line is usually displayed as a dotted line.

There are several ways to interpret a Stochastic Oscillator. Three popular methods include:

1. Buy when the Oscillator (either %K or %D) falls below a specific level (e.g., 20) and then rises above that level. Sell when the Oscillator rises above a specific level (e.g. 80) and then falls below that level.
2. Buy when the %K line rises above the %D line and sell when the %K line falls below the %D line.

Figure 6.16 shows Avon Products and its 10-day Stochastic.

The "buy" arrows are generated when the %K line fell below, and then rose above, the level of 20. Similarly, the "sell" arrows are generated when the %K line rose above, and then fell below, the level of 80.

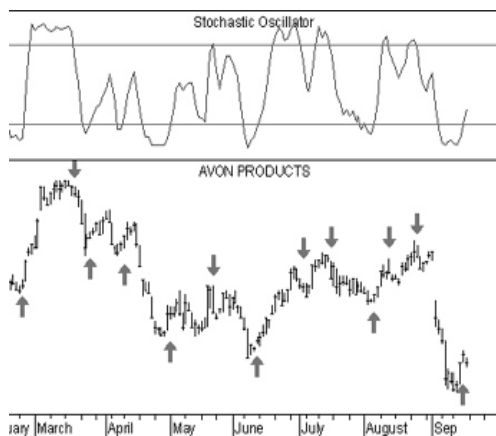


Fig. 6.16 Stochastic Oscillator Indicator



Fig. 6.17 RSI Indicator

6.6.6 RSI – Relative Strength Index

The Relative Strength Index ("RSI") is a popular oscillator. Step-by-step instructions on calculating and interpreting the RSI are provided in Wilder's (1978) book - *New Concepts in Technical Trading Systems*.

When Wilder introduced the RSI, he recommended using a 14-day RSI. Since then, the 9-day and 25-day RSIs have also gained popularity. The fewer days used to calculate the RSI, the more volatile the indicator.

The RSI is a price-following oscillator that ranges between 0 and 100. A popular method of analyzing the RSI is to look for a divergence in which the security is making a new high, but the RSI is failing to surpass its previous high. This divergence is an indication of an impending reversal. When the RSI then turns down and falls below its most recent trough, it is said to have completed a "failure swing." The failure

swing is considered a confirmation of the impending reversal. Further information on the RSI can be found in Wilder's book (1978).

Figure 6.17 shows PepsiCo and its 14-day RSI. A bullish divergence occurred during May and June as prices were falling while the RSI was rising. Prices subsequently corrected and trended upward.

6.6.7 DI – Directional Indicator

The basic Directional Movement trading system involves comparing the 14-day +DI ("Directional Indicator") and the 14-day -DI. This can be done by plotting the two indicators on top of each other or by subtracting the +DI from the -DI. Wilder (1978) suggests buying when the +DI rises above the -DI and selling when the +DI falls below the -DI.

Wilder qualifies these simple trading rules with the "extreme point rule." This rule is designed to prevent whipsaws and reduce the number of trades. The extreme point rule requires that on the day that the +DI and -DI cross, one notes the "extreme point." When the +DI rises above the -DI, the extreme price is the high price on the day the lines cross. When the +DI falls below the -DI, the extreme price is the low price on the day the lines cross.

The extreme point is then used as a trigger point at which one should implement the trade. For example, after receiving a "buy" signal (the +DI rose above the -DI), one should then wait until the security's price rises above the extreme point (the high price

on the day that the +DI and -DI lines crossed) before buying. If the price fails to rise above the extreme point, one should continue to hold your short position.

Figure 6.18 shows Texaco and the +DI and -DI indicators. The "buy" arrows are drawn when the +DI rose above the -DI and "sell" arrows are drawn when the +DI fell below the -DI. Only the significant crossings are labeled.

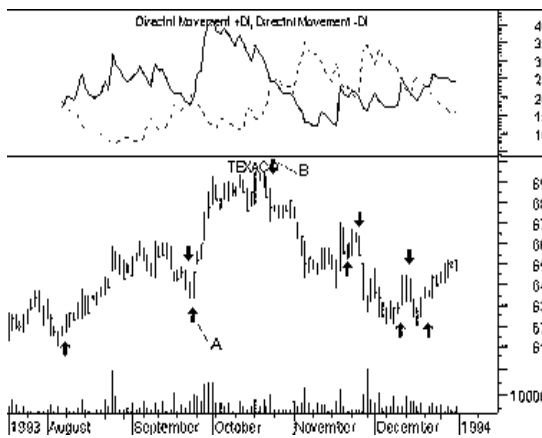


Fig. 6.18 Directional Indicator



Fig. 6.19 Linear Regression Indicator

6.6.8 Linear regression lines

Linear regression is a statistical tool used to predict future values from past values. In the case of security prices, it is commonly used to determine when prices are overextended.

A Linear Regression trendline uses the least squares method to plot a straight line through prices so as to minimize the distances between the prices and the resulting trendline.

A popular method of using the Linear Regression trendline is to construct Linear Regression Channel lines. Two parallel, equidistant lines above and below a Linear Regression trendline are plotted to construct the channel. The distance between the channel lines to the regression line is the greatest distance that any one closing price is from the regression line. Regression Channels contain price movement, with the bottom channel line providing support and the top channel line providing resistance. Prices may extend outside of the channel for a short period of time. However, if prices remain outside the channel for a longer period of time, a reversal in trend may be imminent.

A Linear Regression trendline shows where equilibrium exists. Linear Regression Channels show the range prices can be expected to deviate from a Linear Regression trendline. Figure 6.19 shows the Japanese Yen with a Linear Regression Channel.

Table 6.2 Definitions of the 6 indicators

Indicators	Definition
MACD	$\sum_{i=1}^n EMA_{20}(i) - \sum_{i=1}^n EMA_{40}(i)$ <p>where $EMA_n(i) = a * p(i) + (1-a) * EMA_n(i-1)$, $a = 1/n$</p>
ROC	$(p(i) - p(i-n)) / p(i) * 100$
Stochastic Oscillator	$\sum_{i=1}^n (p(i) - h(i)) / (h(i) - l(i))$
RSI	$RSI = 100 - \frac{100}{1 + RS}$; where $RS = \sum_{i=1}^n (p(i) - p(i-1))^+ / \sum_{i=1}^n (p(i) - p(i-1))^-$
DI	$DI^+ = +DM / TR, DI^- = -DM / TR$ <p>DM – Directional Momentum; TR – True Range</p>
Linear regression lines	$(n * \sum_{i=1}^n i * p(i) - \sum_{i=1}^n i * \sum_{i=1}^n p(i)) / (n * \sum_{i=1}^n i^2 - (\sum_{i=1}^n i)^2)$

where $p(i)$ – close price; $h(i)$ – High price; $l(i)$ – Low price
 n – number of points

Table 6.2 gives the definition of overall 6 indicators.

6.7 Summary

In this chapter, the temporal rule discovery problem is discussed. Since the conventional RST cannot extract temporal rules from a Temporal Information System (TIS), the TIS must be converted to an Information System. There are two methods concerning this conversion. They are “mobile window method” and “columnizing method”. The “columnizing method” is more suitable in this project according to the characteristics of our research objects – financial market forecasting. Between two model-based forecasting methods, technical and fundamental analysis, the technical analysis is chosen because of the lack of necessary information for fundamental analysis. The tools for technical analysis are market indicators, which can reflect the market change accurately. In this project, *WARS (Weighted Accumulated Reconstruction Series)*, a trend-following indicator, has been found to be able to track market changes accurately. By comparison with daily profit curve, the *WARS* can indicate the behavior of *Daily Profit Curve* if two parameters - *Win_length* and *Moving Average Interval (Win_len2)* are chosen properly. In addition, *WARS* can be used to build a trading system. Through the application on S&P 500 index, it can be seen that this indicator is effective and promising. Another 6 well-established indicators are also briefly introduced in this chapter.

Having completed the conversion from TIS to Information System, the next step is to apply the RST to forecast the financial market. In the next chapter, 4 trading systems are presented to check whether using the RST is applicable to forecast the time series.

Chapter 7

Time Series Forecasting Experiments and Discussions

7.1 The Process

In Chapter 6, the temporal rule discovery problem has been discussed. In this chapter, the RST is applied to trading system building problem. The process concerned is illustrated in Figure 7.1.

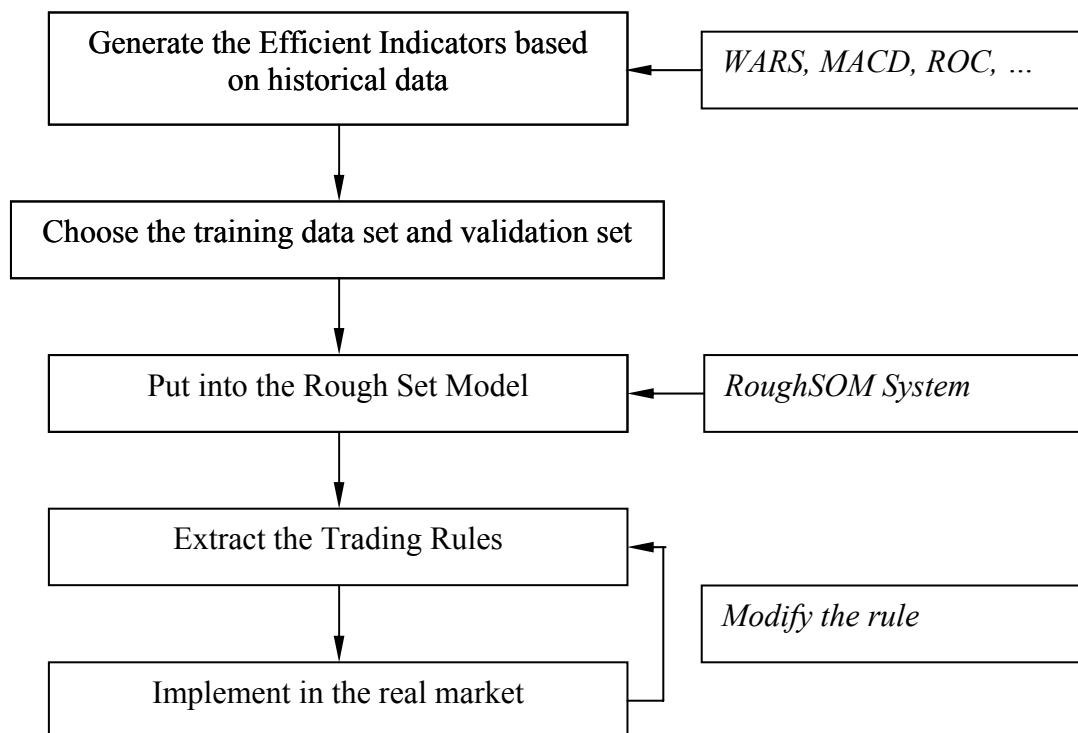


Fig. 7.1 The process of stock market data prediction and analysis

First of all, the historical data should be transferred to rough set objects which can be processed by RoughSOM system. This has been done based on the “columnizing” method, which converts the Temporal Information System to an Information System. All together 7 indicators are chosen to compose the condition attributes in decision table. After that, the decision table is ready for reducts and rules generation. At this step, the decision table is processed using the RoughSOM system. This part may be considered to be the “heart” of the entire process because in this step, there are many factors will affect the final results, such as rule selection, threshold values. These factors will be discussed in the following sections. Through implementing the trading rules on real data, the performance of the RST will be evaluated.

7.2 Data Preparation

In Chapter 6, there are 7 indicators to compose the condition attributes of the decision table were presented. There is also need to define the decision table which is future direction of the data set. The decision attribute is constructed as follows:

$$Dec_att = \left\{ \sum_{i=1}^{20} (21-i) \cdot \text{sign}[\text{close}(i) - \text{close}(0)] \right\} / \sum_{i=1}^{20} i \quad (7.1)$$

where $\text{close}(0)$ is today's close price and $\text{close}(i)$ is the i th close price in the future.

Eq. (7.1) specifies a range for Dec_att of -1 to $+1$. A value of $+1$ indicates that, every day, up to 20 days in the future, the market closed higher than today. Similarly, a -1 indicates that, every day, up to 20 days in the future, the market closed lower than

today. Figure 7.2 gives a snapshot of the S&P 500 index for the period covering from January 1999 to July 1999 and the fluctuation of the *Dec_att*.

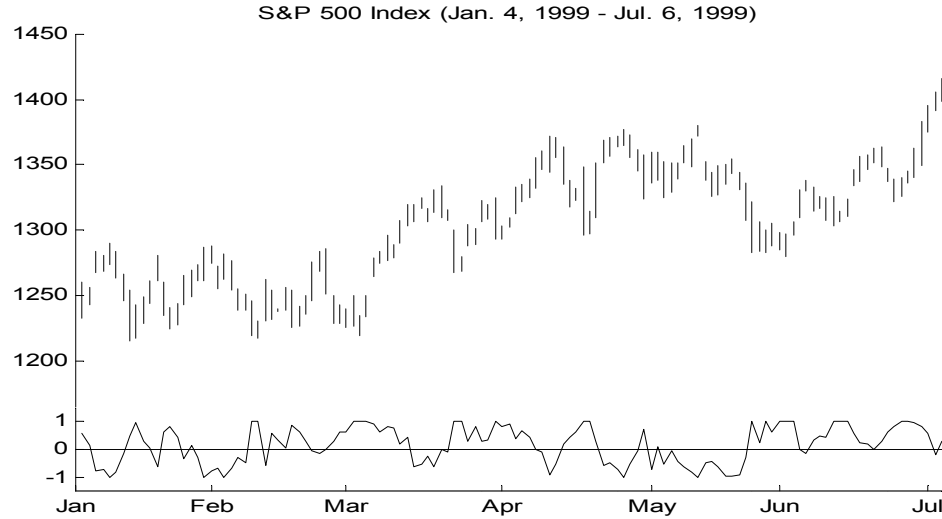


Fig. 7.2 S&P 500 index covering period Jan. through Jul. 1999. The *Dec_att* is shown below the S&P 500 index.

In this analysis, the *Dec_att* is divided into 3 parts, namely, +1 corresponding to buy, 0 to hold and -1 to sell. The different unsupervised discretization methods for dividing the *Dec_att* into 3 parts according to the value of Sharpe Ratio (Sharpe, 1994) and trading numbers are studied. Finally, the equal-frequency-interval method is chosen to implement the division. The Sharpe Ratio is defined as follow:

$$\text{Sharpe ratio} = \frac{\text{mean of the returns}}{\text{standard deviation of the returns}} \quad (7.1)$$

In this project, 4 futures were studied. They are S&P 500 index (Standard & Poor 500 stock index futures), MATIF-CAC index (French government stock index futures), EUREX-BOND (German 10-year government bond) and CBOT-US (United States 30-year government bond), which are provided by Man-Drapeau Pte Ltd. The details on these 4 data sets are given in Table 7.1. There are two types of data used in this

experiment. They are 15-minutes-bar data and daily-bar data, which sampling frequency are 15 minutes and daily respectively. The 15-minutes-bar data is used to calculate the condition attributes. Each indicator is calculated using the data points within one day. Hence, each indicator reflects the fluctuation on the daily basis. The decision attribute is calculated using daily-bar data. In the end, the condition attributes and decision attribute keep the same time frequency. It is unnecessary to shift the data either in row or in column.

Table 7.1 Data sets information

Data Set	Starting date	Ending data	Number of points (before/after 1999)
S&P 500	Jan. 04, 1988	Jul. 26, 1999	2929 (2781 / 248)
MATIF-CAC	May 25, 1993	Jul. 06, 1999	1539 (1388 / 151)
EUREX-BOND	Jan. 02, 1991	Aug. 12, 1999	2155 (2002 / 153)
CBOT-US	Oct. 01, 1990	Jul. 06, 1999	2219 (2072 / 147)

7.3 Rules Extraction

After construction of decision table, these data sets are divided into a rule-extraction set which covers the period before 1999, and a validation set which covers the period after 1999. In both extraction and validation data sets, the rows of data corresponding to $1/6(\max_wars - \min_wars) < WARS < 5/6(\max_wars - \min_wars)$ are removed because the aim of the analysis is to determine the distinguished rules to generate stronger trading signals. This threshold is set to limit the number of the training sets less than 1000 (because the modified Chi2 algorithm introduced in Chapter 4 is suitable for discretizing medium-size data set (<1000)). The threshold varies for different cases. Figure 7.3 provides a snapshot of the S&P 500 index for the period covering from January 1999 to July 1999 and the fluctuation of the *WARS*.

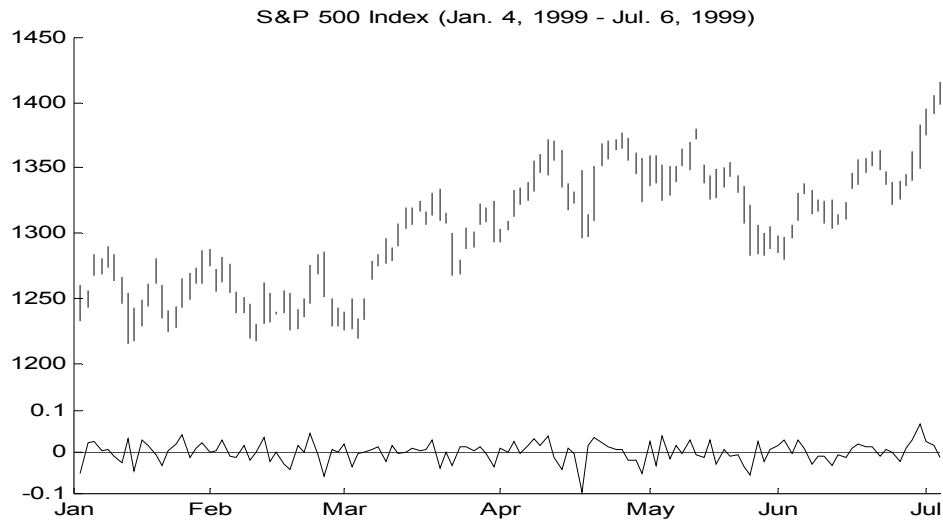


Fig. 7.3 S&P 500 index covering period Jan. through Jul. 1999. The *WARS* indicator is shown below the S&P 500 index.

The selected training data sets are discretized using the modified Chi2 algorithm which has been described in Chapter 4. After discretization, the decision table is sent into the RoughSOM system to generate rules. The format of rules is as follows.

Rule #1:	If	A(2)==1	Then	D=-1
Rule #2:	If	A(4)==15	Then	D=-1
...
...
Rule #50:	If	A(1)==15 & A(6)==65	Then	D=1
Rule #51:	If	A(3)==7 & A(6)==65	Then	D=1
...
...
Rule #300:	If	A(2)==17 & A(6)==6	Then	D=1
Rule #301:	If	A(5)==34 & A(6)==6	Then	D=1
...
...
...

where A – condition attribute; A(1) – directional index;
A(2) – Price rate of change; A(3) – Linear regression lines; A(4) – MACD;
A(5) – RSI indicator; A(6) – Stochastic Oscillator; A(7) – WARS.
D – Decision attribute; -1 – Sell; 0 – Hold; 1 – Buy;

For S&P 500 index, the boundary values for each indicator using the modified Chi2 algorithm on the training data set are as follows.

A(1)	$[-\infty$	-0.632107	-0.622222	-0.548387	-0.533981	
	-0.528438	-0.527851	-0.515982	-0.337580	-0.278431	
	-0.250000	0.245098	0.251462	0.354839	0.358209	
	0.577465	0.582543	0.707071	0.714286	0.991770	$+\infty]$
A(2)	$[-\infty$	-2.874319	-2.296875	-1.899093	-1.223550	
	-1.201639	-0.900386	-0.891027	-0.682880	-0.676382	
	0.088158	0.558154	0.579531	0.649016	0.980279	
	0.981666	1.441516	1.740203	1.754879	7.625146	$+\infty]$
A(3)	$[-\infty$	-0.115781	-0.114774	-0.098352	-0.097802	
	-0.067125	0.154823	0.313858	0.331532	2.387808	$+\infty]$
A(4)	$[-\infty$	-9.763467	-6.150903	-5.940674	-4.869065	
	-4.311108	-2.173778	-2.052539	-1.662823	-1.590944	
	-0.607464	-0.595292	-0.580730	-0.525853	-0.508249	
	-0.477388	-0.425225	-0.410660	-0.001054	0.410532	
	0.421743	1.119033	1.186100	9.843738	$+\infty]$	
A(5)	$[-\infty$	0.173267	0.186147	0.243243	0.248641	
	0.259843	0.283784	0.289157	0.293750	0.310976	
	0.311594	0.335366	0.338883	0.356000	0.360684	
	0.361004	0.370861	0.372917	0.381395	0.381703	
	0.394231	0.394558	0.433673	0.440678	0.550584	
	0.551852	0.650696	0.692857	0.701389	0.702532	
	0.707424	0.707921	0.712644	0.726115	0.887273	$+\infty]$
A(6)	$[-\infty$	0.276610	0.290920	0.313060	0.323843	
	0.330097	0.357194	0.362226	0.364924	0.367089	
	0.368333	0.391667	0.392072	0.403527	0.403704	
	0.406519	0.407104	0.408053	0.410419	0.413914	
	0.414396	0.430886	0.432729	0.453866	0.456790	
	0.458629	0.461420	0.462532	0.463239	0.472542	
	0.473684	0.476013	0.476893	0.490802	0.492342	
	0.500467	0.501372	0.504080	0.506580	0.507064	
	0.509434	0.509655	0.509812	0.510648	0.529863	
	0.531145	0.534979	0.541526	0.542636	0.543896	
	0.546296	0.548309	0.549074	0.555937	0.559335	
	0.559509	0.560920	0.567494	0.570128	0.570692	
	0.571245	0.575949	0.582504	0.582948	0.591435	
	0.595588	0.601107	0.601496	0.602020	0.604056	
	0.606061	0.607720	0.612963	0.613290	0.623670	
	0.623829	0.625377	0.632019	0.632459	0.641481	
	0.642868	0.643857	0.644679	0.647673	0.648389	
	0.648396	0.660354	0.661637	0.684003	0.728283	
	0.730399	0.811581	$+\infty]$			
A(7)	$[-\infty$	-0.044267	-0.044032	-0.041794	-0.041724	
	-0.032739	-0.032458	-0.031871	-0.030384	-0.030098	
	-0.028878	-0.027979	-0.027573	-0.027402	-0.027287	
	-0.027237	-0.026863	-0.026465	-0.025846	-0.025575	

-0.022151	-0.022130	-0.020366	-0.020063	-0.019891	
-0.018841	-0.018732	-0.018485	-0.018372	-0.016144	
-0.014146	-0.013977	-0.013948	0.015235	0.015492	
0.015826	0.015854	0.015945	0.016027	0.017397	
0.017456	0.017697	0.018345	0.018981	0.019191	
0.020161	0.020240	0.022646	0.022892	0.023176	
0.023306	0.023309	0.024057	0.025066	0.026752	
0.027675	0.035256	0.035301	0.037002	0.037051	
0.038633	0.038711	0.048772	0.050115	0.160593	$+\infty]$

The obtained rules are then used to build the trading system. Several runs using different settings, that is, *strength* threshold and methods to solve unseen cases, for each of these are usually necessary in order to obtain a sense of the quality of the extracted rules. Here *strength* threshold is applied to determine the rule number and the generalization of each rule. In case there is no rule matching the new objects, two methods are used. One is by calculating the Euclidean distance, which measures the similarity between the training data and testing data. The category is assigned to this unseen object with the maximal frequency in case that many equal distance are obtained. The other is to assign the unseen object a value 0 since in the special case of building trading system, 0 represents a hold (i.e. without any action). In this way, the trading number will be reduced.

7.4 Results Discussion

The derived rules are encoded into a trading system. The performance of this trading system, for the period January 1988 to July 1999 of S&P 500 Index is shown in Table 7.1-7.2. The commissions and slippage have not been included.

Table 7.2 Performance benchmark of Buy-hold strategy and original decision attribute for period Jan. 4, 1999 to Jul. 26, 1999 of S&P 500 Index

Method		Performance	
		1988-1998	1999
Buy-hold		1146.30	163.00
Dec_att	Net_profit	4624.50	539.60
	max_win	157.50	79.50
	max_loss	-13.25	-1.20
	Trading_number	323	15
	Winning_Trade	293	14
	Sharpe_ratio	0.68	1.31

The results are first compared to the Buy-hold strategy. The Buy-hold strategy is defined as follows:

Buy – hold strategy: A buy-hold strategy assumes that one buy on the first day loaded in the chart and hold the position. The profit is calculated by using the price on the first day and the price on the last day.

From Table 7.2 and 7.3, it can be seen that for the whole test period covering 1988 to 1999, the trading system built using the RST is always better than that of the Buy–hold strategy because it generates more net profit. However, for the validation set, after 1999, it is worse than the Buy-hold strategy. The trading system always loses money and gets a negative Sharpe ratio.

Comparing the different performance corresponding to different *strength* threshold, it can be seen that with the increase of the threshold, the test accuracy rate over the whole period does not fluctuate too much (always around 0.58) which means that about 58% objects can be classified correctly. However, this does not mean a high profit since the order of the buy, hold and sell can be re-organized. The best threshold is 1.0 in this case study with the highest profit and Sharpe ratio. In addition, its performance in the validation set, after 1999, is also better than the others.

Table 7.3 Performance of the trading system for period Jan.4 1999 to Jul.26 1999 of S&P 500 Index

Threshold	Test Accuracy rate	Rule No.	Parameters	Performance			
				1988-1998		1999	
				Eu*	unEu ⁺	Eu	unEu
0.5	0.57786	3307	Net_profit	1509.50	1403.20	33.10	22.10
			max_win	98.50	98.50	45.50	45.50
			max_loss	-31.00	-36.50	-31.00	-36.50
			Trading_number	777	745	42	40
			Winning_Trade	385	362	19	18
			Sharpe_ratio	0.16	0.15	0.04	0.02
1.0	0.58542	1044	Net_profit	1692.60	1479.60	<u>168.70</u>	20.70
			max_win	94.00	94.00	57.50	44.70
			max_loss	-31.00	-37.70	-31.00	-36.50
			Trading_number	825	611	44	38
			Winning_Trade	418	310	23	18
			Sharpe_ratio	0.17	0.18	0.16	0.02
1.5	0.58542	894	Net_profit	1563.60	1261.90	104.00	-79.70
			max_win	94.00	113.50	57.50	44.70
			max_loss	-32.00	-37.70	-32.00	-36.50
			Trading_number	849	557	46	36
			Winning_Trade	427	281	23	14
			Sharpe_ratio	0.16	0.16	0.10	-0.10
2.0	0.58336	362	Net_profit	1418.80	1415.40	-40.60	-197.50
			max_win	94.00	214.50	57.50	44.70
			max_loss	-49.00	-40.50	-49.00	-40.50
			Trading_number	849	415	49	29
			Winning_Trade	435	213	23	9
			Sharpe_ratio	0.15	0.19	-0.04	-0.30
2.5	0.58371	336	Net_profit	1390.60	1128.20	-69.00	-125.50
			max_win	94.00	214.50	57.50	44.70
			max_loss	-49.00	-40.50	-49.00	-40.50
			Trading_number	835	385	51	25
			Winning_Trade	427	189	24	8
			Sharpe_ratio	0.15	0.16	-0.07	-0.22
3.0	0.58439	152	Net_profit	1323.90	1384.40	-115.40	-144.50
			max_win	94.00	206.30	57.50	75.00
			max_loss	-40.00	-88.30	-40.00	-59.10
			Trading_number	829	270	51	21
			Winning_Trade	432	149	22	6
			Sharpe_ratio	0.14	0.23	-0.11	-0.22

*Eu – using Euclidean distance to determine the category of unseen objects;

⁺unEu – assign the unseen object a value 0.

From Table 7.3, it can also be seen that the real trading number is much greater than that generated by *Dec_att*. In Figure 7.4, the trading system covering the validation set at 3.0 threshold is plotted to be compared with the *Dec_att*.

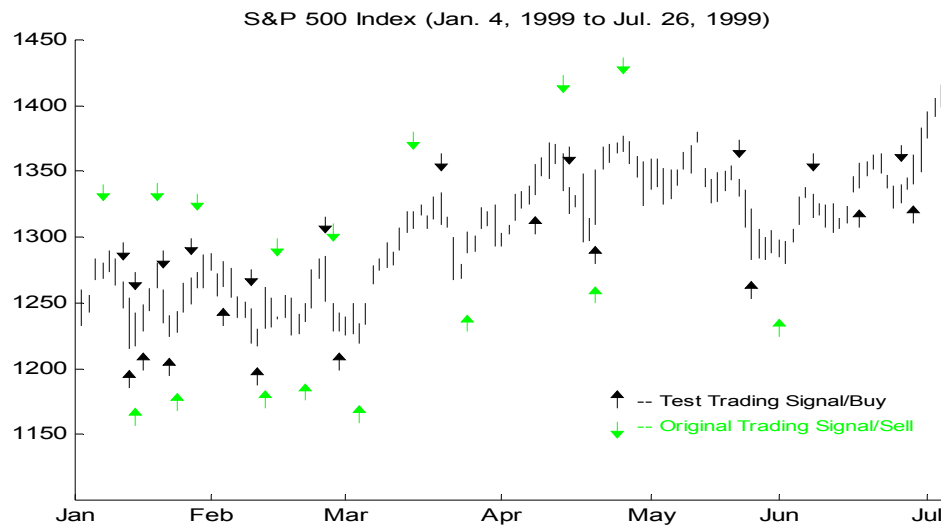


Fig. 7.4 Trading system by using threshold=3.0 and by *Dec_att* covering period Jan.4 to Jul. 26, 1999

From Figure 7.4, it can be seen that some test trading signals are the same as the original signals, which is more favorable to the users. In addition, the test trading signals catch more turning points than the original signal does, which means this trading system is sensitive to trend reverse and it catches more information from this reverse. Correspondingly, there are more trading signals generated from January to March of 1999, which is a more volatile period compared to March to July of 1999. This is also favorable to the users. As it is known that it is difficult to earn money in volatile market for inexperienced users, this trading system provides them more chances to gain. However, there are more trades than the original signals and some trading signals are generated behind the turning points, which means it stays while the trends have changed from up to down or vice versa. From another point view, this may be caused by the noise in time series.

Chapter 8

Conclusions and Recommendations

8.1 Introduction

This chapter describes the conclusions and suggested future works. The conclusions are drawn from the discussion in the preceding chapters.

8.2 Conclusions and Contributions

This thesis has explored the *pros and cons* of applying the RST in the data mining problem. Targeting at applying the RST to time series forecasting problem, a system was built to implement the whole process.

The first step is to convert the Temporal Information System to an Information System, which can be processed using the traditional rough set model. Here the “columnizing” method is applied to implement this process. Each column corresponds to an indicator. For a special case of time series forecasting – trading system construction using financial data – *WARS (Weighted Accumulated Reconstruction Series)* is used to classify the trend of the market. Another 6 well established indicators are included to compose the Information System and in doing so, the Temporal Information System is converted to the traditional rough set objects.

Secondly, the composed decision table is discretized, since traditional RST cannot be applied to continuous values directly. A modified Chi2 algorithm is proposed to implement this process. The modified Chi2 algorithm is adapted from ChiMerge proposed by Randy Kerber (1992), which was advanced to Chi2 algorithm by Liu and Setiono (1997). It overcame the inaccuracy that existed in the original Chi2 algorithm and replaced the stopping criterion to coordinate with the rough set model. Through these two modifications, a completely automatic discretization method is realized with a higher accuracy rate.

The discrete decision table is subsequently sent to the RST to generate reducts and rules. After the rules generation, new objects are classified using these rules. In classfying a new object by matching its description to decision rules, there is a situation that has been overlooked. This is the case whereby the *rule strengths* of both classes are the same and hence there is no clear indication on which class this object belongs to. This is caused by uncertainty and imprecision contained in the data. Although the RST is a powerful tool in dealing with granularity of information and has no requirement of exterior information, it ignores the inner relationships of the data. Therefore, a cluster method – SOM (Self-Organizing Map) – is applied to determine these inner relationships so as to increase the classification accuracy of the RST. Following this idea, a new method termed RoughSOM which combines the RST and SOM is proposed. Through experiments carried out on 10 data sets, it has been shown that this new method removes the uncertainty from the system and increases the predictive accuracy on the test data sets.

The proposed techniques are applied in a case study to forecast a time series and building a trading system. Four futures are chosen as subjects. By comparing with the buy-hold strategy with benchmark of net profit, it shows the trading system built by RoughSOM is efficient and profitable. The results show it did find the inherent rules of the financial market and using the RST to forecast the time series is applicable and promising.

8.3 Recommendations for Future Work

The realization of a general purpose, fully automated data mining system is still far from reach. There are many issues which should be studied further. The following are some interesting topics for future research.

The selection of the training data and testing data will be studied. In studies on time series forecasting problem, what is a reasonable time range limit for the training data? All these problems concern the time series forecasting modeling, such as Markov property. Certainly the experience of the experts will be helpful.

The generated rules should be evaluated and analyzed further. In this project, the final rules were not presented. On one hand, there are a lot of them and so it is impractical to present all of them in this thesis. On the other hand, since the author is unable to collect the information concerning these indicators in the related market and some indicators are seldom studied by other researchers, these indicators' value cannot be judged by comparison with others' work or explained by the experts.

The benchmark for evaluating the performance of the forecasting problem may be changed a little. Conventionally, the predictive accuracy is used to evaluate the forecasting performance. However, for the special case of building a trading system, the final benchmark is the Sharpe ratio and net profit. In this project, the author obtained several examples with similar predictive accuracy but with different Sharpe ratio and net profit. It shows that the predictive accuracy is not proportional to the Sharpe ratio or net profit (this is not valid for predictive accuracy of 100%). This is different from the classification problem. It is an interesting problem to be studied further on.

As mentioned in the summaries of Chapter 4 and 5, the modified Chi2 algorithm cannot be applied into large size data sets. This problem will be studied further with the aim to offer a solution. Similarly, for large classes, the predictive accuracy obtained by RoughSOM is not so satisfactory. It is worth continuing the study on it.

Tables 7.4-7.9 present the performance details on three other stocks using trading system generated by RoughSOM. The performance on these three stocks supports previous analysis.

Table 7.4 Performance benchmark of Buy-hold strategy and original decision attribute for period Jan. 4, 1999 to Jul. 6, 1999 of MATIF-CAC Index

Method		Performance	
		1993-1998	1999
Buy-hold		2821.00	539.50
Dec_att	Net_profit	18951.50	3669.50
	max_win	1031.00	404.00
	max_loss	-170.00	0.00
	Trading_number	144	19
	Winning_Trade	134	19
	Sharpe_ratio	0.82	1.96

Table 7.5 Performance of the trading system for period Jan.4 to Jul. 6 1999 of MATIF-CAC Index

Threshold	Test Accuracy rate	Rule No.	Parameters	Performance			
				1993-1998		1999	
				Eu*	unEu ⁺	Eu	unEu
0.5	0.76103	2719	Net_profit	13727.00	13848.00	1021.00	1021.00
			max_win	790.00	790.00	385.00	385.00
			max_loss	-170.00	-170.00	-141.50	-141.50
			Trading_number	355	347	50	50
			Winning_Trade	238	236	27	27
			Sharpe_ratio	0.40	0.41	0.21	0.21
1.0	0.75905	891	Net_profit	13441.00	12036.00	1002.00	707.00
			max_win	790.00	790.00	379.00	379.00
			max_loss	-170.00	-170.00	-139.00	-141.50
			Trading_number	357	329	52	48
			Winning_Trade	240	212	28	24
			Sharpe_ratio	0.39	0.37	0.20	0.14
1.5	0.75115	745	Net_profit	13148.00	11296.00	943.00	571.00
			max_win	790.00	790.00	379.00	379.00
			max_loss	-170.00	-170.00	-139.00	-141.50
			Trading_number	363	313	52	46
			Winning_Trade	241	191	28	22
			Sharpe_ratio	0.38	0.35	0.19	0.11
2.0	0.7472	326	Net_profit	12658.00	11840.00	643.00	95.00
			max_win	790.00	790.00	379.00	379.00
			max_loss	-170.00	-203.00	-139.00	-203.00
			Trading_number	365	231	49	34
			Winning_Trade	244	161	25	17
			Sharpe_ratio	0.37	0.40	0.13	0.02
2.5	0.74589	288	Net_profit	12818.00	11966.00	785.00	470.00
			max_win	790.00	790.00	379.00	379.00
			max_loss	-170.00	-170.00	-139.00	-141.50
			Trading_number	385	221	51	31
			Winning_Trade	258	156	26	17
			Sharpe_ratio	0.36	0.43	0.16	0.12
3.0	0.74457	125	Net_profit	12608.00	10369.00	1005.00	300.00
			max_win	790.00	1031.00	379.00	379.00
			max_loss	-170.00	-269.00	-150.00	-269.00
			Trading_number	359	149	43	17
			Winning_Trade	246	111	24	10
			Sharpe_ratio	0.38	0.44	0.23	0.10
3.5	0.74852	110	Net_profit	12879.00	8594.00	1143.00	-735.00
			max_win	790.00	1031.00	379.00	141.00
			max_loss	-170.00	-269.00	-150.00	-269.00
			Trading_number	333	104	43	15
			Winning_Trade	226	80	24	4
			Sharpe_ratio	0.40	0.48	0.25	-0.38

*Eu – using Euclidean distance to determine the category of unseen objects;

⁺unEu – assign the unseen object a value 0.

Table 7.6 Performance benchmark of Buy-hold strategy and original decision attribute for period Jan. 4 to Aug. 12, 1999 of EUREX-BUND Index

Method		Performance	
		1991-1998	1999
Buy-hold		27.30	-7.47
Dec_att	Net_profit	217.89	16.41
	max_win	7.23	7.23
	max_loss	-0.88	-0.45
	Trading_number	200	10
	Winning_Trade	173	8
	Sharpe_ratio	0.88	0.70

Table 7.7 Performance of the trading system for period Jan.4 to Aug.12, 1999 of EUREX-BUND Index

Threshold	Test Accuracy rate	Rule No.	Parameters	Performance			
				1991-1998		1999	
				Eu*	unEu ⁺	Eu	unEu
0.5	0.5843	1896	Net_profit	72.85	75.91	-2.02	-3.02
			max_win	3.35	4.38	1.76	1.76
			max_loss	-1.68	-1.50	-1.25	-1.50
			Trading_number	535	471	44	42
			Winning_Trade	277	237	18	17
			Sharpe_ratio	0.20	0.22	-0.07	-0.10
1.0	0.57598	522	Net_profit	73.19	75.35	-4.46	-5.34
			max_win	3.35	4.31	1.76	1.75
			max_loss	-1.68	-1.50	-1.25	-1.50
			Trading_number	597	346	46	40
			Winning_Trade	296	174	16	14
			Sharpe_ratio	0.18	0.25	-0.14	-0.18
1.5	0.53672	239	Net_profit	87.65	74.76	-7.54	-9.82
			max_win	4.53	13.19	1.94	0.00
			max_loss	-2.61	-7.73	-1.24	-7.73
			Trading_number	556	43	45	3
			Winning_Trade	305	33	15	0
			Sharpe_ratio	0.22	0.55	-0.21	-0.84
2.0	0.53672	107	Net_profit	85.59	39.26	-4.38	-9.82
			max_win	4.36	16.04	1.94	0.00
			max_loss	-2.61	-7.73	-1.24	-7.73
			Trading_number	586	17	47	3
			Winning_Trade	319	13	18	0
			Sharpe_ratio	0.21	0.39	-0.12	-0.84
2.5	0.53718	56	Net_profit	84.95	0	-4.58	0
			max_win	4.36	0	1.94	0
			max_loss	-2.61	0	-1.74	0
			Trading_number	584	0	49	0
			Winning_Trade	319	0	20	0
			Sharpe_ratio	0.21	0	-0.12	0

*Eu – using Euclidean distance to determine the category of unseen objects;

⁺unEu – assign the unseen object a value 0.

Table 7.8 Performance benchmark of Buy-hold strategy and original decision attribute for period Jan. 4 to Jul. 6, 1999 of CBOT-US Index

Method		Performance	
		1990-1998	1999
Buy-hold		24.875	-12.125
Dec_att	Net_profit	430.966	28.375
	Max_win	10.875	8.469
	Max_loss	-1.312	0.00
	Trading_number	187	9
	Winning_Trade	168	9
	Sharpe_ratio	1.04	1.20

Table 7.9 Performance of the trading system for period Jan.4 to Jul. 6, 1999 of CBOT-US Index

Threshold	Test Accuracy rate	Rule No.	Parameters	Performance			
				1990-1998		1999	
				Eu*	unEu ⁺	Eu	unEu
0.5	0.52296	1126	Net_profit	49.59	46.96	-0.81	-0.81
			max_win	4.66	4.66	3.22	3.22
			max_loss	-3.12	-3.12	-1.50	-1.50
			Trading_number	768	758	46	46
			Winning_Trade	338	335	16	16
			Sharpe_ratio	0.07	0.07	-0.02	-0.02
1.0	0.5116	428	Net_profit	50.29	58.62	-6.00	-0.13
			max_win	4.66	4.66	2.56	3.22
			max_loss	-3.12	-3.12	-2.375	-2.375
			Trading_number	786	684	48	42
			Winning_Trade	356	308	16	15
			Sharpe_ratio	0.07	0.09	-0.12	-0.00
1.5	0.51023	360	Net_profit	51.32	65.57	-3.88	5.75
			max_win	4.66	5.34	2.34	3.22
			max_loss	-3.12	-3.12	-2.38	-2.38
			Trading_number	756	616	44	32
			Winning_Trade	342	281	14	13
			Sharpe_ratio	0.07	0.10	-0.09	0.14
2.0	0.51114	184	Net_profit	41.59	72.03	1.69	9.69
			max_win	4.72	5.34	2.31	3.22
			max_loss	-3.16	-3.13	-1.38	-2.25
			Trading_number	767	538	44	26
			Winning_Trade	337	256	17	13
			Sharpe_ratio	0.06	0.12	0.04	0.28
2.5	0.51069	158	Net_profit	41.04	85.75	3.38	8.19
			max_win	5.25	5.38	2.53	3.22
			max_loss	-3.16	-3.13	-1.41	-2.25
			Trading_number	755	470	42	24
			Winning_Trade	339	230	18	13
			Sharpe_ratio	0.06	0.15	0.09	0.27
3.0	0.51069	89	Net_profit	32.15	86.15	-0.41	5.19
			max_win	8.50	8.50	3.94	3.31
			max_loss	-4.12	-3.97	-1.41	-2.25
			Trading_number	717	249	37	12
			Winning_Trade	312	124	13	7
			Sharpe_ratio	0.04	0.20	-0.01	0.29

*Eu – using Euclidean distance to determine the category of unseen objects;

⁺unEu – assign the unseen object a value 0.

7.5 Summary

In this chapter, a case study using the RST to build a trading system in financial market is presented. Through a detailed analysis on the S&P 500 Index, the procedure to apply the RST to solve the temporal rule discovery problem is presented. The performance of the trading systems built by RoughSOM system shows this new knowledge discovery tool does find some inherent rules contained in the data. This can be seen from the Sharpe ratio and net profit compared with the Buy-hold strategy. It is a promising alternative to the conventional methods. The performance on three other stocks also supports the author's analysis. However, because of the lack of aid from the experts, generated rules can not be evaluated and validated. In addition, trading system cannot be further improved. The author believes that if expert's experience is available, it will generate more promising results.

Chapter 8

Conclusions and Recommendations

8.1 Introduction

This chapter describes the conclusions and suggested future works. The conclusions are drawn from the discussion in the preceding chapters.

8.2 Conclusions and Contributions

This thesis has explored the *pros and cons* of applying the RST in the data mining problem. Targeting at applying the RST to time series forecasting problem, a system was built to implement the whole process.

The first step is to convert the Temporal Information System to an Information System, which can be processed using the traditional rough set model. Here the “columnizing” method is applied to implement this process. Each column corresponds to an indicator. For a special case of time series forecasting – trading system construction using financial data – *WARS (Weighted Accumulated Reconstruction Series)* is used to classify the trend of the market. Another 6 well established indicators are included to compose the Information System and in doing so, the Temporal Information System is converted to the traditional rough set objects.

Secondly, the composed decision table is discretized, since traditional RST cannot be applied to continuous values directly. A modified Chi2 algorithm is proposed to implement this process. The modified Chi2 algorithm is adapted from ChiMerge proposed by Randy Kerber (1992), which was advanced to Chi2 algorithm by Liu and Setiono (1997). It overcame the inaccuracy that existed in the original Chi2 algorithm and replaced the stopping criterion to coordinate with the rough set model. Through these two modifications, a completely automatic discretization method is realized with a higher accuracy rate.

The discrete decision table is subsequently sent to the RST to generate reducts and rules. After the rules generation, new objects are classified using these rules. In classfying a new object by matching its description to decision rules, there is a situation that has been overlooked. This is the case whereby the *rule strengths* of both classes are the same and hence there is no clear indication on which class this object belongs to. This is caused by uncertainty and imprecision contained in the data. Although the RST is a powerful tool in dealing with granularity of information and has no requirement of exterior information, it ignores the inner relationships of the data. Therefore, a cluster method – SOM (Self-Organizing Map) – is applied to determine these inner relationships so as to increase the classification accuracy of the RST. Following this idea, a new method termed RoughSOM which combines the RST and SOM is proposed. Through experiments carried out on 10 data sets, it has been shown that this new method removes the uncertainty from the system and increases the predictive accuracy on the test data sets.

The proposed techniques are applied in a case study to forecast a time series and building a trading system. Four futures are chosen as subjects. By comparing with the buy-hold strategy with benchmark of net profit, it shows the trading system built by RoughSOM is efficient and profitable. The results show it did find the inherent rules of the financial market and using the RST to forecast the time series is applicable and promising.

8.3 Recommendations for Future Work

The realization of a general purpose, fully automated data mining system is still far from reach. There are many issues which should be studied further. The following are some interesting topics for future research.

The selection of the training data and testing data will be studied. In studies on time series forecasting problem, what is a reasonable time range limit for the training data? All these problems concern the time series forecasting modeling, such as Markov property. Certainly the experience of the experts will be helpful.

The generated rules should be evaluated and analyzed further. In this project, the final rules were not presented. On one hand, there are a lot of them and so it is impractical to present all of them in this thesis. On the other hand, since the author is unable to collect the information concerning these indicators in the related market and some indicators are seldom studied by other researchers, these indicators' value cannot be judged by comparison with others' work or explained by the experts.

The benchmark for evaluating the performance of the forecasting problem may be changed a little. Conventionally, the predictive accuracy is used to evaluate the forecasting performance. However, for the special case of building a trading system, the final benchmark is the Sharpe ratio and net profit. In this project, the author obtained several examples with similar predictive accuracy but with different Sharpe ratio and net profit. It shows that the predictive accuracy is not proportional to the Sharpe ratio or net profit (this is not valid for predictive accuracy of 100%). This is different from the classification problem. It is an interesting problem to be studied further on.

As mentioned in the summaries of Chapter 4 and 5, the modified Chi2 algorithm cannot be applied into large size data sets. This problem will be studied further with the aim to offer a solution. Similarly, for large classes, the predictive accuracy obtained by RoughSOM is not so satisfactory. It is worth continuing the study on it.

References

- Øhrn, A. Discernibility and Rough Sets in Medicine: Tools and Applications. Ph.D Thesis. Dept. of Computer Science and Information Science, Norwegian University of Science and Technology (NTNU). 1999.
- Achelis, S.B. Technical Analysis from A to Z: covers every trading tool - from the Absolute Breadth Index to the Zig Zag. Chicago: Probus Publisher. 1995.
- Ahn, B.S., S. Cho and C. Kim. The integrated methodology of rough set theory and artificial neural network for business failure prediction, *Expert Systems with Applications*, 18, pp.65-74. 2000.
- Ananyan, S. Data Mining for Direct Marketing, *DM Review*, Jan. 2000.
(Downloadable from <http://www.dmreview.com/master.cfm?NavID=55&EdID=1766>)
- Ankerst, M. Visual Data Mining. Ph.D Thesis. University of Munich. 2000.
- Appel, G. and Hitschler, F. Stock Market Trading Systems. Homewood, IL: Dow Jones-Irwin. 1980.
- Baltzersen, J. K. An attempt to Predict Stock Market Data: A Rough Sets Approach. Diploma Thesis. Knowledge Systems Group, Department of Computer Systems and Telematics, The Norwegian Institute of Technology, University of Trondheim. 1996.
- Barkoulas J. and N. Travlos. Chaos in an emerging capital market? The case of the Athens Stock Exchange, *Applied Financial Economics*, 8, pp.231-243. 1998.
- Bazan, J.G. A Comparison of Dynamic and non-Dynamic Rough Set Methods for Extracting Laws from Decision Table. In: *Rough Sets in Knowledge Discovery*, Vol. 1, ed by L. Polkowski and A. Skowron, Chapter 17, pp. 321-365. Heidelberg: Physica-Verlag. 1998.

- Bazan, J.G. and M. Szczuka. RSES and RSESLib – A collection of tools for rough set computations. In: Proceedings of the Second International Conference on Rough Sets and Current Trends in Computing (RSCTC'2000), 2000, Banff, Canada, pp. 74-81.
- Bazan, J.G., A. Skowron and P. Synak. Market data analysis: A Rough Set Approach. ICS Research Reports 6/94, Warsaw University of Technology. 1994.
- Beasley, D., D. R. Bull and R. Martin. An Overview of Genetic Algorithm, University Computing, 15, pp. 170-181. 1993.
- Bishop, G. W. Charles H. Dow and the Dow theory. New York: Appleton-Century-Crofts. 1960.
- Bjorvand, A. T. Time Series and Rough Sets. Diploma Thesis. Department of Computer System and Telematics, The Norwegian Institute of Technology, The University of Trondheim. 1996.
- Brachman, R., T. Khabaza, W. Kloesgen, G. Piatetsky-Shapiro and E. Simoudis. Mining Business Databases, Communications of ACM, 39 (11), pp. 42-48. 1996.
- Chmielewski, M.R. and J.W. Grzymala-Busse. Global Discretization of Continuous Attributes as Preprocessing for Machine Learning, International Journal of Approximate Reasoning, 15 (4), pp. 319 – 331. 1996.
- Cichocki, A. and R. Unbehauen. Neural networks for optimization and signal processing. Chichester: John Wiley & Sons. 1993.
- Dagel, J.F. and R.N. Brady. Diesel Engine & Fuel System Repair. 4th Ed. N.J.: Prentice Hall Press. 1998.
- Demartines, P. and J. Herault. Curvilinear Component Analysis: A Self-Organizing Neural Network for Nonlinear Mapping of Data Sets, IEEE Trans. on Neural Networks, 8 (1), pp. 148-154. 1997.

- Derry, J.F. Database Mining/Knowledge Discovery in Financial Database: An Overview, *Journal of Computational Intelligence in Finance*, 5 (3), pp. 5-9. 1997.
- Dimitras, A.I., R. Slowinski, R. Susmaga and C. Zopounidis. Business Failure Prediction using Rough Sets, *European Journal of Operational Research*, 114, pp. 263 – 280. 1999.
- Dimitras, A.I., S.H. Zanakakis and C. Zopounidis. A survey of business failure with an emphasis on prediction methods and industrial applications, *European Journal of Operational Research*, 90, pp. 487-513. 1996.
- Dougherty, J., R. Kohavi and M. Sahami. Supervised and Unsupervised Discretization of Continuous Features. In: *Machine Learning: Proceedings of the 12th International Conference*, 1995, San Francisco, Calif.: Morgan Kaufmann, pp.194-202.
- Eiben, A.E., T.J. Euverman, W. Kowalczyk and F. Slisser. Modelling customer retention with statistical techniques, rough data models and genetic programming. In: *Rough-Fuzzy Hybridization: A New Trend in Decision-Making*, Chapter 15, ed by A. Skowron and S.K. Pal, pp.330-348. Berlin: Springer. 1998.
- Fayyad, U. and K.B. Irani. Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning. In: *Proc. 13th International Joint Conference on Artificial Intelligence*, 1993, pp. 1022-1027,
- Fayyad, U. and K.B. Irani. On the Handling of Continuous-Valued Attributes in Decision Tree Generation, *Machine Learning*, 8, pp. 87-102. 1992.
- Fayyad, U., G. Piatetsky-Shapior, P. Smyth and R. Uthurusamy (eds). *Advances in Knowledge Discovery and Data Mining*. California: AAAI Press / The MIT Press. 1996.

- Fayyad, U., G. Piatetsky-Shapiro and P. Smyth. From data mining to knowledge discovery: an overview. In: *Advances in Knowledge Discovery and Data Mining*, ed by U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy, pp. 1-29. California: AAAI Press / The MIT Press. 1996.
- Frank, M. and T. Stengos. Measuring the Strangeness of Gold and Silver Rates of Return, *Review of Economic Studies*, 56, pp. 553-567. 1989.
- Freisleben, B. Stock market prediction with backpropagation networks. In: *Industrial and Engineering Applications of Artificial Intelligence and Expert System*, ed by F. Belli and F. J. Radermacher, June 9-12, 1992, Paderborn, Germany, pp. 451-460.
- Frizelle, G. and E. Woodcock. Measuring Complexity as an Aid to Developing Operational Strategy, *International Journal of Operations and Production Management*, 15, pp. 26-39. 1995.
- Furness, P. New pattern analysis methods for database marketing, *Journal of Database Marketing*, 1 (3), pp. 220-232. 1994.
- Golan, R. and D. Edwards. Temporal rules discovery using datalogic/R+ with stock market data. In: *Rough Sets, Fuzzy Sets and Knowledge Discovery*, Proc. International Workshop on Rough Sets and Knowledge Discovery, RSKD'93, October 12-15, 1993, Banff, Alberta, Canada, pp.74-81.
- Golan, R. Stock Market Analysis Utilizing Rough Set Theory. Ph.D. Thesis. Department of Computer Science, University of Regina, Canada. 1995.
- Goonatilake, S. and P. Treleaven (eds). *Intelligent Systems for Finance and Business*, New York: John Wiley & Sons. 1995

- Greco, S., B. Matarazzo and R. Slowinski. A New Rough Set Approach to Evaluation of Bankruptcy Risk. In: Operational Tools in the Management of Financial Risks, ed by C. Zopounidis, pp.121-136. Kluwer Academic Publishers. 1998.
- Greco, S., B. Matarazzo and R. Slowinski. Dealing with missing data in rough set analysis of multi-attribute and multi-criteria decision problems, In: Decision Making: Recent Development and Worldwide Applications, ed by S.H. Zanakis, G. Doukidis and C. Zopounidis, pp.295-316. Dordrecht: Kluwer Academic Publishers. 2000a.
- Greco, S., B. Matarazzo and R. Slowinski. Exploitation of a Rough Approximation of the Outranking Relation in Multicriteria Choice and Ranking. In: Trends in multicriteria decision making, Proc. 13th International Conference on Multiple Criteria Decision Making, January 1997a, Cape Town, South Africa, pp. 45-60.
- Greco, S., B. Matarazzo and R. Slowinski. Fuzzy Extention of the Rough Set Approach to Multicriteria and Multiattribute Sorting. In: Preferences and Decisions under Incomplete Knowledge, ed by J. Fodor, B. Baets and P. Perny, pp. 131-151. Physica-Verlag. 2000b.
- Greco, S., B. Matarazzo and R. Slowinski. Rough Set Approach to Multi-attribute Choice and Ranking Problems. In: Multiple criteria decision making: Proc. 12th International Conference, 1997b, Berlin: Spring-Verlag, pp. 318-329.
- Greco, S., B. Matarazzo and R. Slowinski. Rough Set Processing of Vague Information Using Fuzzy Similarity Relations. In: Finite versus Infinite: contributions to an eternal dilemma, ed by S.C. Calude and G. Paun, pp. 149-174, New York: Springer-Verlag. 2000c.
- Greco, S., B. Matarazzo and R. Slowinski. The use of rough sets and fuzzy sets in MCDM, In: Multicriteria decision making: advances in MCDM models,

- algorithms, theory, and applications, ed by T. Gal, T. Stewart and T. Hanne, Chapter 14, pp. 1-59. Dordrecht: Kluwer Academic Publishers. 1999.
- Greco, S., L.S. Cascio, B. Matarazzo. Rough Set Approach to Stock Selection: An Application to the Italian Market, In: Modelling techniques for financial markets and bank management, ed by M. Bertocchi, E. Cavalli and S. Komlosi, pp. 192-211. Heidelberg: Physica-Verlag. 1996.
- Grzymala-Busse, J.W. and X. Zou. Classification Strategies Using Certain and Possible Rules. In: Rough Sets and Current Trends in Computing, Proc. RSCTC'98 Conference, Warsaw 1998, pp. 37-44.
- Grzymala-Busse, J.W. Knowledge Acquisition under Uncertainty - a Rough Set Approach. Journal of Intelligent and Robotic System, 1, pp. 3-16. 1988.
- Grzymala-Busse, J.W. LERS - A Knowledge Discovery System. In: Rough Sets in Knowledge Discovery, Vol. 2, ed by L. Polkowski and A. Skowron, pp.562-565. Physica-Verlag, 1998.
- Grzymala-Busse, J.W. LERS - A system for learning from examples based on rough sets, In: Intelligent Decision Support - Handbook of Applications and Advances of the Rough Sets Theory, ed by R. Slowinski, Chapter 1, pp. 3-18, Kluwer Academic Publisher, 1992.
- Grzymala-Busse, J.W. Rough-Set and Dempster-Shafer approaches to knowledge acquisition under uncertainty- A comparison. In: Managing Uncertainty in Expert System, pp. 167. Bosten: Kluwer Academic Publishers. 1991.
- Hampton, J. Risk Management: the Equity Curve Revisited. Journal of Computational Intelligence in Finance, 6, pp. 47-50. 1998a.
- Hampton, J. Rough Set Theory: The Basics (Part 1), Journal of Computational Intelligence in Finance, 5 (6), pp. 25-29. 1997.

- Hampton, J. Rough Set Theory: The Basics (Part 2), *Journal of Computational Intelligence in Finance*, 6 (1), pp. 40-42. 1998b.
- Hampton, J. Rough Set Theory: The Basics (Part 3), *Journal of Computational Intelligence in Finance*, 6 (2), pp. 35-37. 1998c.
- Hashemi, R., L.A. Le Blanc, C.T. Rucks and A. Rajaratnam. A hybrid intelligent system for predicting bank holding structures, *European Journal of Operational Research*, 109 (2), pp. 390-402.1998.
- Hertz, J., A. Krogh and R.G. Palmer. *Introduction to the Theory of Neural Computation*. Redwood City, CA: Addison-Wesley, 1991.
- Hiemstra, Y. A stock market forecasting support system based on fuzzy logic. In: *Proceedings of the 27th Annual Hawaii International Conference on System Sciences*, ed by T. Mudge and B.D. Shriver, Jan. 4-7, 1994, Wailea, HI, USA, pp. 281-287.
- Holden, K., D. A. Peel and J. L. Thompson. *Economic forecasting: an introduction*. Cambridge University Press, 1990.
- Hosmer, D.W. and S. Lemeshow. *Applied Logistic Regression*. New York: JohnWiley & Sons, 1989.
- Huang, D. Vibration Analysis and Fault Diagnosis for Diesel Engine, *Journal of Vibration Engineering*, 8 (2), pp.144-149. 1995 (in Chinese).
- Johnston, R.B. From Efficiency to Flexibility: Entropic Measures of Market Complexity and Production Flexibility, *Complexity International*, 3. 1996.
- Kapur, J. and H. Kesavan. *Entropy optimization principles with applications*, Boston: Academic Press. 1992.
- Kaski, S and T. Kohonen. Exploratory data analysis by the self-organizing map: Structures of welfare and poverty in the world. In: *Neural Networks in Financial*

- Engineering, ed by A.N. Refenes, Y. Abu-Mostafa, J. Moody and A. Weigend, pp. 498-507, Singapore: World Scientific. 1996.
- Kaski, S and T. Kohonen. Structures of Welfare and Poverty in the World Discovered by the Self-Organizing Map. Technical Report A24, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland. 1995.
- Kerber, R. ChiMerge: Discretization of Numeric Attributes. In: Proc. 9th International Conference on Artificial Intelligence (AAAI-92), 1992, pp.123-128.
- Kittler, R. and W. Wang. METROLOGY / TEST: The emerging role for data mining, Solid State Technology, 42(11), pp. 44-48. 1999
- Kohavi, R. and M. Sahami. Error-Based and Entropy-Based Discretization of Continuous Features. In: Proc. 2nd International Conference on Knowledge Discovery & Data Mining, 1996, pp. 114-119.
- Kohavi, R. Bottom-up induction of oblivious read-once decision graphs: strengths and limitation. In: Proc. 12th National Conference on Artificial Intelligence, 1994, pp. 613-618.
- Kohonen, T. Self-Organizing Maps. Springer Series in Information Sciences, Vol. 30, Berlin: Springer-Verlag. 1995.
- Komorowski, J., Z. Pawlak, L. Polkowski and A. Skowron. Rough sets: A tutorial. In: Rough fuzzy hybridization: A new trend in decision-making, ed by S.K. Pal and A. Skowron, pp. 3-98. Singapore: Springer-Verlag. 1999.
- Kowalczyk, W. and F. Slisser. Modelling Customer Retention with Rough Data Models. In: Proc. 1st European Symposium, PKDD'97, June 24-27, 1997, Trondheim, Norway, pp.4-13.

- Kowalczyk, W. and Z. Piasta. Rough Set-Inspired Approach to Knowledge Discovery in Business Databases. In: Proc. 2nd Pacific-Asia Conference, PAKDD-98, April 15-17, 1998, Melbourne, Australia, pp.186-197.
- Kowalczyk, W. Rough Data Modelling: A New Technique for Analyzing Data. In: Rough Sets in Knowledge Discovery, Vol. 1, Chapter 20, ed by L. Polkowski and A. Skowron, pp.400-421. Physica-Verlag. 1998a.
- Kowalczyk, W. TRANCE: A Tool for Rough Data Analysis, Classification, and Clustering, In: Proc. 4th International Workshop on Rough Sets, Fuzzy Sets and Machine Discovery, RSDF'96, 1996, Tokyo, pp. 269-275.
- Kowalczyk, W. TRANCE: A Tool for Rough Data Analysis, Classification, and Clustering. In: Rough Sets in Knowledge Discovery, Vol. 2, ed by L. Polkowski and A. Skowron, pp.566-568. Physica-Verlag. 1998b.
- Krawiec, K., R. Slowinski and D.Vanderpooten. Construction of Rough Classifiers Based on application of a Similarity Relation. In: Proc. 4th International Workshop on Rough Sets, Fuzzy Sets and Machine Discovery, RFSD'96, 1996, Tokyo, Japan, pp. 23-30.
- Kryszkiewicz, M. Properties of Incomplete Information Systems in the Framework of Rough Sets. In: Rough Sets in Knowledge Discovery, Vol. 1, Chapter 21, ed by L. Polkowski and A. Skowron, pp. 422-450. Physica-Verlag. 1998.
- Kumar, A. New Techniques for Data Reduction in a Database System for Knowledge Discovery Applications. Journal of Intelligent Information Systems, 10 (1), pp. 31-48. 1998.
- Lenarcik, A. and Z. Piasta. Discretization of condition attributes space, In: Intelligent Decision Support - Handbook of Applications and Advances of the Rough Sets Theory, ed by R. Slowinski, pp. 373-389. Kluwer Academic Publishers. 1992.

- Lenarcik, A. and Z. Piasta. Probabilistic approach to decision algorithm generation in the case of continuous condition attributes, *Foundations of Computing and Decision Sciences*, 18 (3-4), pp. 213-223. 1993a.
- Lenarcik, A. and Z. Piasta. Probabilistic Rough Classifiers with mixture of discrete and continuous attributes. In: *Rough Sets and Data Mining. Analysis of Imprecise Data*, ed by T.Y. Lin and N. Cercone, pp. 373-383. Kluwer Boston: Academic Publishers. 1997.
- Lenarcik, A. and Z. Piasta. ProbRough - A System for Probabilistic Rough Classifiers Generation. In: *Rough Sets in Knowledge Discovery*, Vol. 2, ed by L. Polkowski and A. Skowron, pp.569-571. Physica-Verlag. 1998.
- Lenarcik, A. and Z. Piasta. Rough classifiers. In: *Rough Sets, Fuzzy Sets and Knowledge Discovery*, Proc. International Workshop on Rough Sets and Knowledge Discovery (RSKD'93), October 12-15, 1993b, Banff, Alberta, Canada, pp.298-316.
- Lin, T.Y. and A.J. Tremba. Attribute Transformations on Numeric Databases and Its Applications to Stock Market and Economic Data. In: *Proc. 4th Pacific-Asia Conference, PAKDD 2000*, April 18-20, 2000, Kyoto, Japan, pp. 181-192.
- Lin, T.Y. and N. Cercone (ed). *Rough Sets and Data Mining: Analysis for Imprecise Data*. Boston: Kluwer Academic Publishers. 1997.
- Lin, T.Y. and Y.Y. Yao. Mining Soft Rules Using Rough Sets and Neighborhoods. In: *Proc. Symposium on Modeling, Analysis and Simulation, Computational Engineering in Systems Applications (CESA'96)*, IMACS Multi Conference, Vol. 2 of 2, July 9-12, 1996, Lille, France, pp. 1095-1100.
- Lin, T.Y. Neighborhood Systems and Approximation in Database and Knowledge Base Systems. In: *Proc. 4th International Symposium on Methodologies of*

- Intelligent Systems, Poster Session, October 12-15, 1989, Charlotte, USA, pp. 75-86.
- Lin, T.Y. Rough Set Theory in very large database. In: Proc. Symposium on Modeling, Analysis and Simulation, Computational Engineering in Systems Applications (CESA'96), IMACS Multi Conference, Vol. 2 of 2, July 9-12, 1996, Lille, France, pp. 936-941.
- Lin, T.Y., T. Hinke, D. Marks and B. Thurasingham. Security and Database Mining, In: Proc. 9th Annual IFIP TC11 Working Conference on Database Security, August 1995, Rensselaerville, USA, pp. 391-399.
- Liu, H. and R. Setiono, Feature Selection via Discretization of Numeric Attributes, IEEE Trans. Knowledge and Data Engineering, 9 (4), pp.642-645. 1997.
- Liu, H. On the Integration and Extraction of Diagnostic Information. Ph.D Thesis, Xi'an Jiaotong University. 1997 (in Chinese).
- Mayfield E. and B. Mizrach. On Determining the Dimension of Real-Time Stock-Price Data, Journal of Business & Economic Statistics, 10, pp. 367-374. 1992.
- Meng, J. Some Advanced Techniques in Fault Feature Extraction for Large Rotating Machinery, Ph.D Thesis, Xi'an Jiaotong University, 1996 (in Chinese).
- Merz, C.J. and P.M. Murphy, UCI Repository of machine learning databases, <http://www.ics.uci.edu/~mlearn/MLRepository.html>.
- Michalski, R.S. A Theory and Methodology of Inductive Learning, Artificial Intelligence, 20, pp. 111-161. 1983.
- Michalski, R.S. A Theory and Methodology of Inductive Learning. In: Machine learning: an artificial intelligence approach, ed by R.S. Michalski, J.G. Carbonell and T.M. Mitchell, pp. 83-134. Palo Alto: Tioga Pub. Co. 1983.

- Mienko, R., R. Slowinski, J. Stefanowski and R. Susmaga. RoughFamily - software implementation of rough set based data analysis and rule discovery techniques. In: Proc. 4th International Workshop on Rough Sets, Fuzzy Sets and Machine Discovery, RSDF'96, 1996, Tokyo, pp. 437-440.
- Mills, D. Finding the likely buyer using rough sets technology. American Salesman, 38 (8), pp. 3-5. 1993.
- Montgomery, D.C. and G.C. Runger. Applied Statistics and Probability for Engineers. 2nd Edition, New York: John Wiley & Sons. 1999.
- Mrozek, A. and K. Skabek. Rough Sets in Economic Applications. In: Rough Sets in Knowledge Discovery, Vol. 2, Chapter 13, ed by L. Polkowski and A. Skowron, pp.238-271. Physica-Verlag. 1998.
- Nguyen, H.S. and A. Skowron. Boolean Reasoning for Feature Extractions Problem. In: Proc. 10th International Symposium on Methodologies for Information Systems (ISMIS'97), 1997, Charlotte, NC, USA, pp.117-126.
- Nguyen, H.S. and A. Skowron. Quantization of Real Values Attributes, Rough set and Boolean Reasoning Approaches. In: Proc. International Workshop on Rough Sets and Soft Computing at Second Joint Conference on Information Sciences, JCIS'95, 1995, Wrightsville Beach, NC, USA, pp. 34-37.
- Nguyen, H.S. and S.H. Nguyen. Discretization Methods for Data Mining. In: Rough Sets in Knowledge Discovery, Vol. 1, Chapter 22, ed by L. Polkowski and A. Skowron, pp.451-482. Physica-Verlag. 1998.
- Nguyen, H.S. Discretization of real value attributes: Boolean reasoning approach, Ph.D thesis, Warsaw University, 1997.

- Nguyen, H.S. Discretization Problems for Rough Set Methods. In: Rough Sets & Current Trend in Computing, Proc. 1st International Conference, RSCTC'98, June 1998, Warsaw, Poland, pp. 545-552.
- Nguyen, H.S. From Optimal Hyperplanes to Optimal Decision Trees, *Fundamenta Informaticae*, 34 (1-2), pp. 145-174. 1998a.
- Nowicki, R., R. Slowinski and J. Stefanowski. Evaluation of vibroacoustic diagnostic symptoms by means of the rough sets theory, *Computers in Industry*, 20, pp. 141-152. 1992.
- Pardo, R. Design, Testing and Optimization of Trading Systems, John Wiley & Sons. 1992.
- Pawlak, Z. and R. Slowinski. Rough set approach to multi-attribute decision analysis, *European Journal of Operational Research*, 72, pp. 443-459. 1994.
- Pawlak, Z. Rough Sets, *International Journal of Computer and Information Sciences*, 11 (5), pp. 341-356. 1982.
- Pawlak, Z. Rough Sets, Theoretical Aspects of Reasoning about Data. Dordrecht: Kluwer Academic Publishers. 1991.
- Pawlak, Z. Rough Sets. In: Rough Sets and Data Mining, ed by T.Y. Lin and N. Cercone, pp. 3-8. Kluwer Academic Publisher. 1997.
- Piasta, Z. and A. Lenarcik. Learning Rough Classifiers from Large Databases with Missing Values. In: Rough Sets in Knowledge Discovery, Vol. 1, Chapter 23, ed by L. Polkowski and A. Skowron, pp.483-499. Physica-Verlag. 1998.
- Piasta, Z. and A. Lenarcik. Rule induction with probabilistic rough classifiers. ICS Research Report 24/96, Warsaw University of Technology. 1996. Also in Machine Learning (to appear).

- Piatetsky-Shapiro, G. Data Mining and Knowledge Discovery in Business Databases. In: Proc. 9th International Symposium ISMIS-96, 1996, Zakopane, Poland, pp. 56-67.
- Poel, D. and Z. Piasta. Purchase Prediction in Database Marketing with the ProbRough System. In: Rough Sets and Current Trends in Computing, Proc. 1st International Conference, RSCTC'98, June 22-26, 1998, Warsaw, Poland, pp. 593-600.
- Poel, D. Rough Sets for Database Marketing. In: Rough Sets in Knowledge Discovery 2, Chapter 17, ed by L. Polkowski and A. Skowron, pp.324-335. Physica-Verlag. 1998.
- Polkowski, L. and A. Skowron (eds.). Rough Sets in Knowledge Discovery, Vol. 1: methodology and applications. Vol. 2: applications, case studies, and software systems. Physica-Verlag. 1998.
- Predki, B., R. Slowinski, J. Stefanowski, R. Susmaga and S. Wilk. ROSE - Software Implementation of the Rough Set Theory. In: Rough Sets and Current Trends in Computing, Proc. RSCTC'98 Conference, 1998, Warsaw, pp. 605-608.
- Qu, L. and J. Meng. Fault-diagnosis Techniques and Current Advanced Sciences. In: Proc. 95' Conference on Equipment Diagnostic Techniques, 1995, P. R. China, pp. 9-34 (in Chinese).
- Qu, L. and Y. Shen. Orbit complexity: a new criterion for evaluating the dynamic quality of rotor systems, Journal of Mechanical Engineering Sciences, 207, pp. 325-334. 1993.
- Qu, L. and Z. He. Machinery Fault Diagnostics. Shanghai Science Press. 1982
- Quinlan, J.M. C4.5: Programs for Machine Learning, San Mateo, Calif.: Morgan Kaufmann, 1993.

- Quinlan, J.M. Improved Use of Continuous Attributes in C4.5, *Journal of Artificial Intelligence Research*, 4, pp.77-90. 1996.
- Quinlan, R. Induction of decision trees, *Machine Learning*, 1, pp. 81-106. 1986.
- Risvik, K. M. Discretization of Numerical Attributes, *Preprocessing for Machine Learning*. Project Report, Knowledge Systems Group, Department of Computer Systems and Telematics, The Norwegian Institute of Technology, University of Trondheim. 1997.
- Ronen, B. and R. Karp. An Information Entropy Approach to the Small-Lot Concept, *IEEE Transactions on Engineering Management*, 41, pp. 89-92. 1994.
- Roubens, M. and P.H. Vincke. *Preference Modelling*. Lecture Notes in Economics and Mathematical Systems, Vol. 250, Berlin: Springer-Verlag. 1985.
- Roy, B. Main sources of inaccuracy determination, uncertainty and imprecision in decision models, *Mathematical and Computer Modeling*, 12, pp. 1245-1254. 1989.
- Ruggiero, M. How to build a system framework, *Futures*, 23 (12), pp. 50-56. 1994b.
- Ruggiero, M. Rules are made to be traded, *AI in Finance*, Fall, pp. 35-40. 1994a.
- Ruggiero, M. Turning the Key, *Futures*, 23 (14), pp. 38-40. 1994c.
- Sakai, H. and A. Okuma. An Algorithm for Checking Dependencies of Attributes in a Table with Non-deterministic Information: A Rough Sets Based Approach. In: *Proc. 6th Pacific Rim International Conference on Artificial Intelligence*, August 28-September 1, 2000, Melbourne, Australia, pp. 219-229.
- Sakai, H. and A. Okuma. An Algorithm for Finding Equivalence Relation from Table with Non-deterministic Information. In: *New directions in rough sets, data mining, and granular-soft computing*, *Proc. 7th International Workshop, RSFDGrC'99*, November 1999, Yamaguchi, Japan, pp. 64-72.

- Shan, N., H.J. Hamilton, W. Ziarko and N. Cercone. Discretization of continuous valued attributes in classification systems. In: Proc. 4th International Workshop on Rough Sets, Fuzzy Sets and Machine Discovery, RFSD'96, 1996, Tokyo, Japan, pp. 74-81.
- Sharpe, W.F. The Sharpe Ratio, *Journal of Portfolio Management*, pp. 49-58. Fall 1994.
- Shavlik, J.W. and T.G. Dietterich (eds). *Readings in machine learning*. Morgan Kaufmann Publishers. 1990.
- Shen, L., E.H. Tay, L. Qu and Y. Shen. Fault Diagnosis Using Rough Sets Theory, *Computers in Industry*, 43 (1), pp. 61-72. 2000.
- Skalko, C. Rough Sets Help Time the OEX, *Journal of Computational Intelligence in Finance*, 4 (6), pp. 20-27. 1996.
- Skowron, A. and C. Rauszer, The discernibility matrices and functions in information systems. In: *Intelligent Decision Support - Handbook of Applications and Advances of the Rough Sets Theory*, ed by R. Slowinski, pp. 331-362. Kluwer Academic Publishers. 1992.
- Skowron, A. and J. Stepaniuk. Information granules and approximation spaces. In: *Proc. 7th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU'98*, 1998, Paris, France, pp. 1354-1361.
- Skowron, A. and J. Stepaniuk. Tolerance approximation spaces, *Fundamenta Informaticae*, 27 (2-3), pp. 245-253. 1996.
- Skowron, A. Boolean Reasoning for Decision Rules Generation. In: *Methodologies for Intelligent Systems*, ed by J. Komorowski and Z.W. Ras, pp. 295-305. Berlin: Springer-Verlag. 1993.

- Slowinski, K. Rough Classification of HSV Patients. In: Intelligent Decision Support - Handbook of Applications and Advances of the Rough Sets Theory, ed by R. Slowinski, Chapter 6, pp.77-94, Kluwer Academic Publishers, 1992.
- Slowinski, R. (ed). Intelligent Decision Support - Handbook of Applications and Advances of the Rough Sets Theory. Kluwer Academic Publishers. 1992.
- Slowinski, R. and C. Zopounidis. Application of the rough set approach to evaluation of bankruptcy risk, International Journal of Intelligent Systems in Accounting, Finance & Management, 4 (1), pp. 27-41. 1995.
- Slowinski, R. and C. Zopounidis. Rough-Set Sorting of Firms According to Bankruptcy Risk. In: Applying Multiple Criteria Aid for Decision to Environmental Management, ed by M. Paruccini, pp. 339-357. Kluwer Academic Publishers. 1994.
- Slowinski, R. and C. Zopounidis. Application of the rough set approach to evaluation of bankruptcy risk, Intelligent Systems in Accounting, Finance and Management, 4, pp. 27-41. 1995,
- Slowinski, R. and D. Vanderpooten. A Generalized Definition of Rough Approximations based on Similarity, IEEE Transactions on Knowledge and Data Engineering, 12 (2), pp. 331-336. 2000.
- Slowinski, R. and D. Vanderpooten. Similarity Relation as a basis for Rough Approximation. In: Advances in Machine Intelligence & Soft Computing, ed by P.P. Wang, pp. 17-33. Raleigh NC: Bookwright. 1995.
- Slowinski, R. and J. Stefanowski. On limitations of Using Rough Set Approach to Analyze Non-Trivial Medical Information Systems. In: Proc. 4th International Workshop on Rough Sets, Fuzzy Sets and Machine Discovery, RSDF'96, 1996, Tokyo, pp. 176-183.

- Slowinski, R. and J. Stefanowski. Rough Classification with Valued Closeness Relation. In: New Approaches in Classification and Data Analysis, ed by E. Diday, pp. 482-488. Berlin: Springer-Verlag. 1994.
- Slowinski, R. and J. Stefanowski. Rough Family - Software Implementation of the Rough Set Theory. In: Rough Sets in Knowledge Discovery, Vol. 2, ed by L. Polkowski and A. Skowron, pp.581-586. Physica-Verlag. 1998.
- Slowinski, R. and J. Stefanowski. RoughDAS and RoughClass software implementation of the rough sets approach. In: Intelligent Decision Support - Handbook of Applications and Advances of the Rough Sets Theory, ed by R. Slowinski, pp.445-456. Kluwer Academic Publishers. 1992.
- Slowinski, R. Rough set learning of preferential attitude in multi-criteria decision making, In: Methodologies for Intelligent System, ed by J. Komorowski and Z.W. Ras, pp. 642-651. Berlin: Springer-Verlag. 1993.
- Slowinski, R., C. Zopounidis and A.I. Dimitras. Prediction of Company Acquisition in Greece by means of the Rough Set Approach, European Journal of Operational Research, 100, pp. 1-15. 1997.
- Slowinski, R., C. Zopounidis, A.I. Dimitras and R. Susmaga. Rough Set Predictor of Business Failure. In: Soft computing in financial engineering, ed by R.A. Ribeiro, H.J. Zimmermann, R.R. Yager and J. Kacprzyk, pp. 402-424. New York: Physica-Verlag. 1999.
- Stefanowski, J. and D. Vanderpooten. A General Two-stage Approach to Inducing rules from Examples. In: Rough Sets, Fuzzy Sets and Knowledge Discovery, ed by W. Ziarko, pp.317-325. Berlin: Springer-Verlag. 1994.

- Stefanowski, J. On Rough Set Based Approaches to Induction of Decision Rules. In: Rough Sets in Knowledge Discovery, Vol. 1, Chapter 24, ed by L. Polkowski and A. Skowron, pp. 500-529. Physica-Verlag. 1998b.
- Stefanowski, J. The Rough Set Based Rule Induction Techniques for Classification Problems. In: Proc. 6th European Congress on Intelligent Techniques and Soft computing, September 7-10, 1998a, Aachen, pp. 109-113.
- Susmaga, R., W. Michalowski and R. Slowinski. Identifying Regularities in Stock Portfolio Titling. Interim Report IR-97-66, International Institute for Applied Systems Analysis. 1997.
- Synak, P. Rough Set Expert System – User's Guide. Version 1.0, 1995.
- Szladow, A. and D. Mills. Tapping Financial Databases, Business Credit, 95 (7), pp. 8. 1993.
- Tan, D. The Vibration Control of Internal Combustion Engine. Southwest Jiaotong University Press, P. R. China. 1995 (in Chinese).
- Thomason, M.R. Dynamic Normalization: Outliers and Time, Journal of Computational Intelligence in Finance, 6, pp. 43-44. 1998.
- Thompson, J. Targeting for response value and profit, Journal of Targeting, Measurement and Analysis for Marketing, 3, pp. 133-146. 1994.
- Tsumoto, S. and H. Tanaka. Automated discovery of medical expert system rules from clinical databases based on Rough Sets. In: Proc. 2nd International Conference on Knowledge Discovery and Data Mining, KDD'96, 1996, pp. 63-69.
- Tsumoto, S., S. Kobayashi, H. Tanaka and A. Nakamura (eds). Proc. 4th International Workshop on Rough Sets, Fuzzy Sets and Machine Discovery, RSDF'96, Tokyo. 1996.

- Webster, A.L. Applied Statistics for Business and Economics. 2nd Ed., McGraw-Hill Inc. 1995.
- Weiss, E. Rough Sets, Rough Neurons, Induction and Data Mining #2, Journal of Computational Intelligence in Finance, 5 (3), pp. 10-11. 1997.
- Weiss, E. Rough Sets, Rough Neurons, Induction and Data Mining, NEUROVE\$T Journal, 4 (6), pp. 28. 1996.
- Weiss, S.M. and C.A. Kulikowski. Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems. Calif.: Morgan Kaufmann. 1990.
- Wilder, J. W. New Concepts in Technical Trading Systems. Greensboro, NC: Trend Research. 1978.
- Wroblewski, J. Finding minimal reducts using genetic algorithm. In: Proc. International Workshop on Rough Sets Soft Computing at 2nd Annual Joint Conference on Information Sciences, JCIS'95, 1995, Wrightsville Beach, North Carolina, USA, pp. 186-189.
- Wroblewski, J. Genetic Algorithm in Decomposition and Classification Problems. In: Rough Sets in Knowledge Discovery, Vol. 2, Chapter 24, ed by L. Polkowski and A. Skowron, pp. 471-487. Physica-Verlag. 1998.
- Ziarko, W. (ed). Rough Sets, Fuzzy Sets and Knowledge Discovery, Proc. International Workshop on Rough Sets and Knowledge Discovery, RSKD'93, October 12-15, 1994. Banff, Alberta, Canada.
- Ziarko, W. Variable Precision Rough Set Model, Journal of Computer and System Sciences, 46, pp. 39-59. 1993.
- Ziarko, W., R. Golan and D. Edwards. An application of Datalogic/R knowledge discovery tool to identify strong predictive rules in stock market data. In: Proc.

AAAI Workshop on Knowledge Discovery in Databases, 1993, Washington D.C., USA, pp. 93-101.

Appendix A

Proof: The χ^2 value of the reconstructed decision table is greater than that of the original table for a 2-class decision table

Table A.1 The original decision table

Decision type	1	2	R_i
1	A_{11}	A_{12}	R_1
2	A_{21}	A_{22}	R_2
C_j	C_1	C_2	N

Table A.2 The reconstructed decision table

Decision type	1	2	3	4	R_i
			(Original DA=1; SOM DA=2)	(Original DA=2; SOM DA=1)	
1	B_{11}	B_{12}	B_{13}	B_{14}	R_1
2	B_{21}	B_{22}	B_{23}	B_{24}	R_2
C_j'	C_1'	C_2'	C_3'	C_4'	N

The χ^2 value of the original decision table and reconstructed decision table:

$$\chi^2_o = \frac{(A_{11} - E_{11})^2}{E_{11}} + \frac{(A_{12} - E_{12})^2}{E_{12}} + \frac{(A_{21} - E_{21})^2}{E_{21}} + \frac{(A_{22} - E_{22})^2}{E_{22}};$$

$$\chi^2_r = \frac{(B_{11} - E_{11}')^2}{E_{11}'} + \frac{(B_{12} - E_{12}')^2}{E_{12}'} + \frac{(B_{13} - E_{13}')^2}{E_{13}'} + \frac{(B_{14} - E_{14}')^2}{E_{14}'}$$

$$+ \frac{(B_{21} - E_{21}')^2}{E_{21}'} + \frac{(B_{22} - E_{22}')^2}{E_{22}'} + \frac{(B_{23} - E_{23}')^2}{E_{23}'} + \frac{(B_{24} - E_{24}')^2}{E_{24}'};$$

where: $A_{11} = B_{11} + B_{13}$; $A_{12} = B_{12} + B_{14}$; $A_{21} = B_{21} + B_{23}$; $A_{22} = B_{22} + B_{24}$;

$$E_{11} = E_{11}' + E_{13}'; \quad E_{12} = E_{12}' + E_{14}'; \quad E_{21} = E_{21}' + E_{23}'; \quad E_{22} = E_{21}' + E_{24}';$$

$$C_1 = C_1' + C_3'; \quad C_2 = C_2' + C_4';$$

Here we only consider $\frac{(A_{11} - E_{11})^2}{E_{11}}$ and $\frac{(B_{11} - E_{11}')^2}{E_{11}'} + \frac{(B_{13} - E_{13}')^2}{E_{13}'}$, we study the

difference of these two items:

$$\begin{aligned} \frac{(A_{11} - E_{11})^2}{E_{11}} &= \frac{A_{11}^2 - 2A_{11}E_{11} + E_{11}^2}{E_{11}} = \frac{A_{11}^2}{E_{11}} - 2A_{11} + E_{11}; \\ \frac{(B_{11} - E_{11}')^2}{E_{11}'} + \frac{(B_{13} - E_{13}')^2}{E_{13}'} &= \frac{B_{11}^2}{E_{11}'} - 2B_{11} + E_{11}' + \frac{B_{13}^2}{E_{13}'} - 2B_{13} + E_{13}'; \\ &= \frac{B_{11}^2}{E_{11}'} + \frac{B_{13}^2}{E_{13}'} - 2A_{11} + E_{11}'; \end{aligned}$$

In the above two equations, the later two parts are the same. Now we calculate the difference between them:

$$\begin{aligned} \frac{B_{11}^2}{E_{11}'} + \frac{B_{13}^2}{E_{13}'} - \frac{A_{11}^2}{E_{11}} &= \frac{B_{11}^2}{R_1 C_1' / N} + \frac{B_{13}^2}{R_1 C_3' / N} - \frac{A_{11}^2}{R_1 C_1 / N} = \frac{1}{R_1 / N} \left(\frac{B_{11}^2}{C_1'} + \frac{B_{13}^2}{C_3'} - \frac{A_{11}^2}{C_1} \right) \\ &= \frac{1}{R_1 / N} \left(\frac{B_{11}^2 C_3' C_1 + B_{13}^2 C_1' C_1 - (B_{11} + B_{13})^2 C_1' C_3'}{C_1' C_3' C_1} \right) \\ &= \frac{1}{R_1 C_1' C_3' C_1 / N} (B_{11}^2 C_3' C_1 + B_{13}^2 C_1' C_1 - B_{11}^2 C_1' C_3' - B_{13}^2 C_1' C_3' - 2B_{11} B_{13} C_1' C_3') \\ &= \frac{1}{R_1 C_1' C_3' C_1 / N} (B_{11}^2 (C_3' C_1 - C_1' C_3') + B_{13}^2 (C_1' C_1 - C_1' C_3') - 2B_{11} B_{13} C_1' C_3') \\ &= \frac{1}{R_1 C_1' C_3' C_1 / N} (B_{11}^2 C_3'^2 + B_{13}^2 C_1'^2 - 2B_{11} B_{13} C_1' C_3') = \frac{1}{R_1 C_1' C_3' C_1 / N} (B_{11} C_3' - B_{13} C_1')^2 \\ &\geq 0 \end{aligned}$$

$$\therefore \frac{(B_{11} - E_{11}')^2}{E_{11}'} + \frac{(B_{13} - E_{13}')^2}{E_{13}'} \geq \frac{(A_{11} - E_{11})^2}{E_{11}};$$

Similarly,

$$\frac{(B_{12} - E_{12})^2}{E_{12}} + \frac{(B_{14} - E_{14})^2}{E_{14}} \geq \frac{(A_{12} - E_{12})^2}{E_{12}};$$

$$\frac{(B_{21} - E_{21}')^2}{E_{21}'} + \frac{(B_{23} - E_{23}')^2}{E_{23}'} \geq \frac{(A_{21} - E_{21})^2}{E_{21}};$$

$$\frac{(B_{22} - E_{22}')^2}{E_{22}'} + \frac{(B_{24} - E_{24}')^2}{E_{24}'} \geq \frac{(A_{22} - E_{22})^2}{E_{22}}.$$

$$\therefore \chi^2_r \geq \chi^2_o$$

Publications of the author

1. Shen, L., E.H. Tay, L. Qu and Y. Shen. Fault Diagnosis Using Rough Sets Theory, Computers in Industry, 43 (1), pp. 61-72. 2000. (Chapter 3)
2. Tay, E.H. and L. Shen. A Modified Chi2 Algorithm for Discretization, IEEE Trans. On Knowledge and Data Engineering, 2001. (to appear) (Chapter 4)
3. Tay, E.H. and L. Shen. Economic and financial prediction using rough sets model, The European Journal of Operational Research, 2001. (to appear) (Chapter 2)
4. Shen, L. and E.H. Tay. A Discretization Method for Rough Sets Theory, Intelligent Data Analysis, 2001. (to appear) (Chapter 4)
5. Tay, E.H. and L. Shen. Utilization of SOM to Remove Uncertainty of Rough Sets Sorting System, Submitted to Knowledge and Information Systems-An International Journal, 2001. (Chapter 5)
6. Tay, E.H. and L. Shen. Contingency Management based on Rough Sets Theory, Submitted to Engineering Applications of Artificial Intelligence, 2001. (Chapter 3)
7. Shen, L. and E.H. Tay. Classifying Market States with WARS. In: Proc. 2nd International Conference on Data Mining, Financial Engineering, and Intelligent Agents - IDEAL 2000, December 13-15, 2000, Shatin, N.T., Hong Kong. Lecture Notes in Computer Science 1983. pp. 280-285. (Chapter 6)
8. Shen, L. and E.H. Tay. Diagnosing Valve Clearance Fault using Multi-Parameter Fusion, In: Proc. 2nd International Conference on Information Fusion, July 6 - 8, 1999, California, USA, Vol. II, pp. 960-965.