Rule Based Systems for Classification in Machine Learning Context

by

Han Liu

The thesis is submitted in partial fulfilment of the requirements for the award of the degree of Doctor of Philosophy of the University of Portsmouth

October 2015

Abstract

This thesis introduces a unified framework for design of rule based systems for classification tasks, which consists of the operations of rule generation, rule simplification and rule representation. This thesis also stresses the importance of combination of different rule learning algorithms through ensemble learning approaches.

For the three operations mentioned above, novel approaches are developed and validated by comparing with existing ones for advancing the performance of using this framework. In particular, for rule generation, Information Entropy Based Rule Generation is developed and validated through comparing with Prism. For rule simplification, Jmid-pruning is developed and validated through comparing with J-pruning and Jmax-pruning. For rule representation, rule based network is developed and validated through comparing with decision tree and linear list. The results show that the novel approaches complement well the existing ones in terms of accuracy, efficiency and interpretability.

On the other hand, this thesis introduces ensemble learning approaches that involve collaborations in training or testing stage through combination of learning algorithms or models. In particular, the novel framework Collaborative and Competitive Random Decision Rules is created and validated through comparing with Random Prisms. This thesis also introduces the other novel framework Collaborative Rule Generation which involves collaborations in training stage through combination of multiple learning algorithms. This framework is validated through comparing with each individual algorithm. In addition, this thesis shows that the above two frameworks can be combined as a hybrid ensemble learning framework toward advancing overall performance of classification. This hybrid framework is validated through comparing with Random Forests.

Finally, this thesis summarises the research contributions in terms of theoretical significance, practical importance, methodological impact and philosophical aspects. In particular, theoretical significance includes creation of the framework for design of rule based systems and development of novel approaches relating to rule based classification. Practical importance shows the usefulness in knowledge discovery and predictive modelling and the independency in application domains and platforms. Methodological impact shows the advances in generation, simplification and representation of rules. Philosophical aspects include the novel understanding of data mining and machine learning in the context of human research and learning, and the inspiration from information theory, system theory and control theory toward methodological innovations. On the basis of the completed work, this thesis provides suggestions regarding further directions toward advancing this research area.

Acknowledgement

The author would like to thank the University of Portsmouth for awarding him the PhD studentship to conduct the research activities that produced the results disseminated in this thesis. The author needs to particularly thank his supervision team- Dr Alexander Gegov, Dr Mihaela Cocea and Dr Mohamed Bader for the continuous support, encouragement and advice. Special thanks must go to his parents Wenle Liu and Chunlan Xie as well as his brother Zhu Liu for the financial support during his academic studies in the past as well as the spiritual support and encouragement for his embarking on the PhD research in the past two and half an years. In addition, the author would also like to thank his best friend Yuqian Zou for the continuous support and encouragement during his PhD study that have facilitated significantly his involvement in the writing process for this thesis.

The author would like to thank Prof Timothy Ross from University of New Mexico for the authored book entitled "Fuzzy Logic with Engineering Applications" which brings in his gain of knowledge in fuzzy logic theory and applications. The author would also like to thank the academic editor for the Springer book series "Studies in Big Data" Prof Janusz Kacprzyk and the executive editor for this series Dr Thomas Ditzinger for offering him the great chance in publishing in this series the research monograph entitled "Rule Based Systems for Big Data: A Machine Learning Approach". In addition, the author also needs to give thanks to Prof Witold Pedrycz from University of Alberta and Prof Shyi-Ming Chen from National Taiwan University of Science and Technology as well as Prof Vladimir Jotsov from State University for Library Studies and Information Technologies for offering him chances in publishing several book chapters in edited volumes of Springer.

Declaration of Authorship

Whilst registered as a candidate for the above degree, I have not been registered for any other research award. The results and conclusions embodied in this thesis are the work of the named candidate and have not been submitted for any other academic award.

Word Count: 49620

Table of Contents

List of Tables

List of Figures

Abbreviations

Bagging: Bootstrap aggregating

Boosting: Adaboost

CCRDR: Collaborative and Competitive Decision Rules

CRG: Collaborative Rule Generation

CART: Classification and Regression Trees

DT: Decision Trees

KNN: K Nearest Neighbours

IEBRG: Information Entropy Based Rule Generation

LDA: Linear Discriminant Analysis

LL: Linear Lists

NN: Neural Networks

PCA: Principal Component Analysis

RBN: Rule Based Networks

RF: Random Forests

SVM: Support Vector Machines

TDIDT: Top-Down Induction of Decision Trees

UCI: University of California, Irvine

Disseminations

Research Monographs

H. Liu, A. Gegov and M. Cocea, Rule Based Systems for Big Data: A Machine Learning Approach. Studies in Big Data 13, Springer, 2016.

Book Chapters

H. Liu, M. Cocea and A. Gegov, Interpretability of Computational Models for Sentiment Analysis. In: W. Pedrycz and S. M. Chen (eds.) Sentiment Analysis and Ontology Engineering: An Environment of Computational Intelligence, Studies in Computational Intelligence, Springer. (In press)

H. Liu and A. Gegov, Induction of Modular Classification Rules by Information Entropy Based Rule Generation. In: V. Sgurev, R. Yager, J. Kacprzyk and V. Jotsov (eds.) Innovative Issues in Intelligent Systems, Studies in Computational Intelligence 623, Springer. (In press)

H. Liu and A. Gegov, Collaborative Decision Making by Ensemble Rule Based Classification Systems. In: W. Pedrycz and S. M. Chen (eds.) Granular Computing and Decision Making: Interactive and Iterative Approaches, Studies in Big Data 10, Springer, pp.245-264, 2015.

H. Liu, A. Gegov and F. Stahl, Unified Framework for Construction of Rule Based Classification Systems. In: W. Pedrycz and S. M. Chen (eds.) Information Granularity, Big Data and Computational Intelligence, Studies in Big Data 8, Springer, pp.209-230, 2015.

Journal Articles

H. Liu, A. Gegov and M. Cocea, Collaborative Rule Generation: An Ensemble Learning Approach. Journal of Intelligent and Fuzzy Systems, IOS Press. (In press) IF: 1.81

H. Liu, A. Gegov and F. Stahl, J-measure Based Hybrid Pruning for Complexity Reduction in Classification Rules. WSEAS Transactions on Systems, 12(9), pp. 433-446, 2013.

Conference Papers

H. Liu, A. Gegov and M. Cocea, Network Based Rule Representation for Knowledge Discovery and Predictive Modelling. In: IEEE International Conference on Fuzzy Systems, 2-5 August 2015, Istanbul, Turkey, pp.1-8.

H. Liu, A. Gegov and M. Cocea, Hybrid Ensemble Learning Approach for Generation of Classification Rules. In: International Conference on Machine Learning and Cybernetics, 12-15 July 2015, Guangzhou, China, pp.377-382.

H. Liu, A. Gegov and F. Stahl, Categorization and Construction of Rule Based Systems. In: International Conference on Engineering Applications of Neural Networks, 5 - 7 September 2014, Sofia, Bulgaria, Springer, pp.183-194.

Presentations

"Rule Based Systems for Big Data: A Machine Learning Approach", Computing Seminar, University of Portsmouth, 30 September 2015.

"Hybrid Ensemble Learning Approach for Generation of Classification Rules", the International Conference on Machine Learning and Cybernetics 2015, IEEE Systems, Man and Cybernetics Society, Guangzhou, China, 14 July 2015.

"Challenges and Opportunities of Machine Learning", Institute of Cosmology and Gravitation Research Seminar, University of Portsmouth, 21 April 2015.

"Data Mining and Machine Learning: Scientific and Philosophical Perspectives", School of Computing Research Seminar, University of Portsmouth, 7 January 2015.

"Construction of Rule Based Systems in Machine Learning Context", Faculty of Technology Research Conference, University of Portsmouth, 3 July 2014.

"Rule Based Systems: A Machine Learning Approach", Computing Seminar, University of Portsmouth, 26 March 2014.

"Knowledge Discovery and Predictive Modelling by Rule Based Networks", School of Computing Research Seminar, University of Portsmouth, 9 October 2013.

"Healthcare Process Modelling by Rule Based Networks", Healthy Computing Workshop, University of Portsmouth, 22 May 2013.

Posters

"Reduction of Bias and Variance by Ensemble Learning Approaches", Faculty of Technology Research Conference, University of Portsmouth, 3 June 2015.

"Reduction of Bias and Variance in Machine Learning", School of Computing Student Conference, University of Portsmouth, 18 March 2015.

"Construction of Rule Based Systems in Machine Learning Context", Faculty of Technology Research Conference, University of Portsmouth, 3 July 2014.

"Generation, Simplification and Representation of Classification Rules", Faculty of Technology Research Conference, University of Portsmouth, 26 June 2013.

Chapter 1 Introduction

1.1 Background

Expert systems have been increasingly popular for commercial applications. A rule based system is a special type of expert system. The development of rule based systems began in the 1960's but became popular in the 1970's and 1980's (Partridge & Hussain, 1994). A rule based system consists of a set of rules, which can serve many purposes such as decision support or predictive decision making in real applications. One of the main challenges in this area is the construction of such systems which could be based on both expert knowledge and data. Thus the design techniques can be divided into two categories: expert based design and data based design. The former follows traditional engineering approaches while the later follows machine learning approaches. For both types of approaches, the designed rule based systems could be used for practical tasks such as classification, regression and association.

This thesis recommends the use of data based approaches instead of expert based approaches. This is because expert based approaches have some limitations which can usually be overcome by using data based approaches. For example, expert knowledge is likely to be incomplete or inaccurate; some of experts' viewpoints may be biased; engineers may misunderstand requirements or have technical designs with defects. When problems with high complexity are solved, both domain experts and engineers are difficult to have all possible cases considered or to have perfect technical designs. Once a failure arises with an expert system, experts or engineers may have to find and fix the problem through reanalysis or redesign. However, the real world has been filled with Big Data. Some previously unknown information or knowledge could be discovered from data. Data would potentially be used as supporting evidence to reflect some useful and important pattern by using modelling techniques. More importantly, the model could be revised automatically in accordance with the update of database in real time if a data based modelling technique is used. Therefore, data based approaches would be more suitable than expert based approaches for design of complex rule based systems. This thesis mainly focuses on theoretical and empirical studies of rule based systems for classification in the context of machine learning.

Machine learning is a branch of artificial intelligence and involves two stages: training and testing. Training aims to learn something from known properties by using learning algorithms and testing aims to make predictions on unknown properties by using the knowledge learned in training stage. From this point of view, training and testing are also known as learning and prediction respectively. In practice, a machine learning task aims to build a model that is further used to make predictions by adopting learning algorithms. This task is usually referred to as predictive modelling. Machine learning could be divided into two types: supervised learning and unsupervised learning, in accordance with the form of learning. Supervised learning means learning with a teacher. This is because all instances from a data set are labelled. The aim of this type of learning is to build a model by learning from labelled data and then to make predictions on other unlabelled instances with regard to the value of a predicted attribute. The predicted value of an attribute could be either discrete or continuous. Therefore, supervised learning could be involved in both classification and regression tasks for categorical prediction and numerical prediction respectively. In contrast, unsupervised learning means learning without a teacher. This is because all instances from a data set are unlabelled. The aim of this type of learning is to find previously unknown

patterns from data sets. It includes association, which aims to identify correlations between attributes, and clustering, which aims to group objects based on similarity measures. On the other hand, machine learning algorithms are popularly used in data mining tasks to discover some previously unknown pattern. This task is usually referred to as knowledge discovery. From this point of view, data mining tasks also involve classification, regression, association and clustering. Both classification and regression can be used to reflect the correlation between multiple independent variables and a single dependent variable. The difference between classification and regression is that the former typically reflects the correlation in qualitative aspects whereas the latter reflects in quantitative aspects. Association is used to reflect the correlation between multiple independent variables and multiple dependent variables in both qualitative and quantitative aspects. Clustering can be used to reflect patterns in relation to grouping of objects.

In data mining and machine learning, automatic induction of classification rules has become increasingly popular in commercial applications such as decision support and decision making. In this context, the methods of rule generation can be divided into two categories: 'divide and conquer' (Quinlan, 1993) and 'separate and conquer' (Michalski R. S., 1969). The former is also known as Top-Down Induction of Decision Trees (TDIDT), which generates classification rules in the intermediate form of a decision tree such as ID3, C4.5 and C5.0 (Quinlan, 1993). The latter is also known as covering approach (Fürnkranz, 1999), which generates if-then rules directly from training instances such as Prism (Cendrowska, 1987). A series of experiments have shown that Prism, which acts as a representative of the methods that follow 'separate and conquer' approach, achieves a similar level of accuracy compared with TDIDT in most cases and can even outperform TDIDT in other cases especially in noise domain (Bramer, 2000).

However, a principal problem (Bramer, 2002) that arises with most methods for generation of classification rules is the overfitting of training data, the solution of which is likely to result in a bias termed as overfitting avoidance bias in (Fürnkranz, 1999; Schaffer, 1993; Wolpert, 1993). In some cases, the overfitting problem may result in the generation of a large number of complex rules. This may not only increase computational cost but also lower the accuracy in predicting further unseen instances. This has led to the development of pruning algorithms with respect to the reduction of overfitting. Pruning methods could be subdivided into two categories- pre-pruning (Fürnkranz, 1999; Bramer, 2007), which truncates rules during rule generation, and post-pruning (Fürnkranz, 1999; Bramer, 2007), which generates a whole set of rules and then removes a number of rules and rule terms, by using statistical or other tests (Bramer, 2002). A family of pruning algorithms are based on J-measure used as information theoretic means of quantifying the information content of a rule (Smyth & Goodman, 1991). This is based on the working hypothesis (Bramer, 2002) that rules with high information content (value of J-measure) are likely to have a high level of predictive accuracy. Two existing J-measure based pruning algorithms are J-pruning (Bramer, 2002) and Jmax-pruning (Stahl & Bramer, 2011; Stahl & Bramer, 2012), which have been successfully applied to Prism for the reduction of overfitting.

The main objective in prediction stage of machine learning is to find the first rule that fires by searching through a rule set. As efficiency is concerned, a suitable structure is required to effectively represent a rule set. The existing rule representations include decision tree and linear list. Tree representation is mainly used to represent rule sets generated by 'divide and conquer' approach in the form of decision trees. It has root and internal nodes representing

attributes and leaf nodes representing classifications as well as branches representing attribute values. On the other hand, list representation is commonly used to represent rules generated by 'separate and conquer' approach in the form of 'if-then' rules.

Each classification algorithm would have its own strengths and limitations and possibly perform well on some datasets but poorly on the others due to its suitability to particular datasets. This has led to the development of ensemble learning concepts for the purpose of increasing overall classification accuracy of a classifier by generating multiple base classifiers and combining their classification results (Stahl & Bramer, 2013; Stahl & Bramer, 2011; Stahl F. , Gaber, Liu, Bramer, & Yu, 2011; Stahl F. , et al., 2012).

The above description is to specify ways to address the common issues that arise in data mining and machine learning areas through scaling up algorithms (Brain, 2003; Stahl & Bramer, 2013). However, the outcome of machine learning tasks does not only depend on the performance of learning algorithms but also on the characteristics of data set such as dimensionality and sample size. In other words, the performance of a particular machine learning algorithm usually depends on its suitability to the characteristics of a data set. For example, some algorithms are unable to directly deal with continuous attributes such as Original Prism introduced in (Cendrowska, 1987). For this kind of algorithms, it is required to discretise continuous attributes prior to training stage. A popular method of discretization of continuous attributes is ChiMerge (Kerber, 1992). The discretization of continuous attributes usually helps speed up the process of training greatly. This is because the attribute complexity is reduced through discretising the continuous attributes (Brain, 2003). However, it is also likely to lead to loss of accuracy. This is because information usually gets lost to some extents after a continuous attribute is discretized as mentioned in (Brain, 2003). In addition, some algorithms prefer to deal with continuous attributes such as K Nearest Neighbour (KNN) (Altman, 1992) and Support Vector Machine (SVM) (Tan, Steinbach, & Kumar, 2006; Press, Teukolsky, Vetterling, & Flannery, 2007).

On the other hand, the performance of an algorithm is also subject to data dimensionality and sample size. If they are massively large, it would usually result in huge computational costs. It is also likely to generate hypothesis that over-fits training data but under-fits test data due to the presence of noise and coincidental pattern that exists in the training data. This is due to the following issues. A large training set is likely to have coincidental patterns included, which are not scientifically reliable. In this case, a generated model that over-fits training data usually performs poor accuracy on test data. In contrast, if the size of a sample is too small, it is likely to learn bias from training data as the sample could only have a small coverage for the scientific pattern. Therefore, it is necessary to effectively choose representative samples for training data. With regard to dimensionality, it is objectively possible that not all of the attributes are relevant to making classifications. In this case, some attributes need to be removed from the training set by feature selection techniques if the attributes are irrelevant. Therefore, it is necessary to examine the relevance of attributes in order to effectively reduce data dimensionality. The above descriptions mostly explain why an algorithm may perform better on some data sets but worse on others. All of these issues mentioned above often arise in machine learning tasks so the issues also need to be taken into account by rule based classification algorithms in order to improve classification performance. On the basis of above descriptions, it is necessary to have data set pre-processed prior to training stage. This is usually referred to as scaling down data (Brain, 2003). This mainly consists of dimensionality reduction and sampling. For dimensionality

reduction, some popular existing methods include Principle Component Analysis (PCA) (Jolliffe, 2002), Linear Discriminant Analysis (LDA) (Yu & Yang, 2001) and Information Gain based methods (Azhagusundari & Thanamani, 2013). Some popular sampling methods include simple random sampling (Yates, Moore, & Starnes, 2008), probabilistic sampling (Deming, 1975) and cluster sampling (Kerry & Bland, 1998).

In addition to predictive accuracy, interpretability is also a significant aspect if the machine learning approaches are adopted in data mining tasks for the purpose of knowledge discovery. As mentioned above, machine learning methods can be used for two main purposes. One is to build a predictive model that is used to make predictions. The other one is to discover some meaningful and useful knowledge from data. For the latter purpose, the knowledge discovered is later used to provide insights for a knowledge domain. For example, a decision support system is built in order to provide recommendations to people with regard to a decision. People may not trust the recommendations made by the system unless they are convinced through seeing the reasons of the decision making. From this point of view, it is required to have an expert system which works in a white box manner. This is in order to make the expert system transparent so that people can understand the reasons why the output is derived from the system.

As mentioned above, rule based system is a special type of expert systems. This type of expert systems works in a white box manner. Higgins justified in (Higgins, 1993) that interpretable expert systems need to be able to provide the explanation with regard to the reason of an output and that rule based knowledge representation makes expert systems more interpretable with the following arguments:

- A network was conceived in (Uttley, 1959), which needs a number of nodes exponential in the number of attributes in order to restore the information on conditional probabilities of any combination of inputs. It is argued in (Higgins, 1993) that the network restores a large amount of information that is mostly less valuable.
- Another type of networks known as Bayesian Networks introduced in (Kononenko I. , 1989) needs a number of nodes which is the same as the number of attributes. However, the network only restores the information on joint probabilities based on the assumption that each of the input attributes is totally independent of the others. Therefore, it is argued in (Higgins, 1993) that this network is unlikely to predict more complex relationships between attributes due to the lack of information on correlational probabilities between attributes.
- There are some other methods that fill the gaps that exist in Bayesian Networks by deciding to only choose some higher-order conjunctive probabilities, such as the first neural networks (Rosenblatt, 1962) and a method based on correlation/dependency measure (Ekeberg & Lansner, 1988). However, it is argued in (Higgins, 1993) that these methods still need to be based on the assumption that all attributes are independent of each other.

On the basis of above arguments, Higgins recommended the use of rule based knowledge representation due mainly to the advantage that rules used to interpret relationships between attributes can provide explanations with regard to the decision of an expert system (Higgins, 1993). Therefore, Higgins argues the significance of interpretability, i.e. the need to explain the output of an expert system based on the reasoning of that system. From this point of view, rule based systems have high interpretability in general. However, in machine learning context, due to the presence of massively large data, it is very likely to have a complex system built, which makes the knowledge extracted from such a system cumbersome and

less readable for people. In this case, it is necessary to represent the system in a way that has a high level of interpretability. On the other hand, different people would usually have different levels of cognitive capability. In other words, the same message may make different meaning to different people due to their different levels of capability of reading and understanding. In addition, different people would also have different levels of expertise and different preferences with regard to the way of receiving information. All these issues raised above make it necessary that knowledge extracted from a rule based system needs to be represented in a way that suits people to read and understand. This indicates the necessity of proper selection of rule representation techniques. Some representation techniques are introduced in Chapters 2 and 3 and discussed with respect to their advantages and disadvantages. In addition, Chapter 6 outlines a list of impact factors and evaluation criteria for interpretability of rule based systems.

This thesis mainly introduce adopting the way by scaling up algorithms to address the issues that arise with rule generation, rule simplification and ensemble learning with respect to improving the result of machine learning tasks. This thesis also introduces the way by proper selection of rule representation to address the issues on interpretability of rule based systems. However, the thesis does not include sections on scaling down data in depth. This is because all data sets used in the experiments are retrieved from popular machine learning repositories which are used for researchers to validate their newly developed algorithms. The data sets are relatively small and particularly pre-processed by experts in accordance with sampling and feature relevance. Therefore, it is not relevant to particularly scale down the data again in this PhD thesis. However, this part on scaling down data is further incorporated into the research methodology that is introduced in Chapter 3 and the way to achieve that is also specified in Chapter 6 as a further direction of this research.

1.2 Aim and Objectives

The aim of this research is to create a theoretical framework for design of rule based classification systems for the purpose of knowledge discovery and predictive modelling.

To fulfil the aim, it is necessary to achieve the objectives as follows:

1. To develop advanced methods and techniques in relation to rule based classification including generation, simplification and representation of classification rules.
2. To create two advanced frameworks of ensemble learning for classification.
3. To validate the methods and techniques described in objectives 1 and 2.

Besides, this thesis also includes discussions in philosophical aspects in Chapter 6. This is in order to introduce novel understanding of the concepts and methodologies introduced in the thesis in relation to data mining and machine learning. The recommendations on evaluation and improvement of interpretability are also described in Chapters 5 and 6.

1.3 Theoretical Preliminaries

Section 1.1 introduces the background in relation to rule based systems and machine learning. However, there are some fundamental concepts that strongly relate to the two subjects such as discrete mathematics, statistics, if-then rules, algorithms and logic. This section describes these concepts in detail in order to help readers understand those more technical sections in Chapter 2 and 3.

1.3.1 Discrete Mathematics

Discrete mathematics is a branch of mathematical theory, which includes three main topics namely mathematical logic, set theory and graph theory. In this thesis, the research methodology introduced in Chapter 3 is strongly based on Boolean logic that is a theoretical application of mathematical logic in computer science. As mentioned in Section 1.1, a rule based system consists of a set of rules. In other words, rules are basically stored in a set, which is referred to as rule set. In addition, the data used in machine learning tasks is usually referred to as dataset. Therefore, set theory is also strongly related to the research methodology in this thesis. The development of rule based network, which is introduced in Chapter 2 and Chapter 3, is fundamentally based on graph theory. On the basis of the above description, this subsection introduces in more detail the three topics as part of discrete mathematics with respects to their concepts and connections to the research methodology.

As introduced in (Simpson, 2013; Ross, 2004), mathematical logic includes the propositional connectives namely conjunction, disjunction, negation, implication and equivalence. Conjunction is also referred to as AND logic in computer science and denoted by $F=a \wedge b$. The conjunction could be illustrated by the conjunction truth table below:

Table 1.1 Conjunction Truth Table 1

a	b	F
0	0	0
0	1	0
1	0	0
1	1	1

Table 1.1 essentially implies that if the output is positive if and only if all the inputs are positive in AND logic. In other words, if any one of the inputs is negative, it would result in a negative output. In practice, the conjunction is widely used to make judgements especially on safety critical judgement. For example, it can be used for security check systems and the security status is positive if and only if all parameters relating to the security are positive. In this thesis, the conjunction is typically used to judge if a rule is firing and more details about it are presented in Section 1.3.3.

Disjunction is also referred to as OR logic in computer science and denoted by $F= a \vee b$. The disjunction is illustrated by the disjunction truth table below:

Table 1.2 Disjunction Truth Table

a	b	F
0	0	0
0	1	1
1	0	1
1	1	1

Table.1.2 essentially implies that the output would be negative if and only if all of the inputs are negative in OR logic. In other words, if any one of the inputs is positive, then it would result in a positive output. In practice, it is widely used to make judgements on alarm

systems. For example, an alarm system would be activated if any one of the parameters appears to be negative.

Implication is popularly used to make deduction, and denoted by F= a⟶ b. The implication is illustrated by the truth table below:

Table 1.3 Implication Truth Table

a	b	F
0	0	1
0	1	1
1	0	0
1	1	1

Table 1.3 essentially implies that 'a' is defined as an antecedent and 'b' as a consequent. In this context, it supposes that the consequent would be deterministic if antecedent is satisfied. In other words, 'a' is seen as the adequate but not necessary condition of 'b', which means if 'a' is true then 'b' would definitely be true but b may be either true or false otherwise. In contrast, if 'b' is true, it is not necessarily due to that 'a' is true. This can also be proved as follows:

$F= a\longrightarrow b \Leftrightarrow \neg a \vee b$

It can be seen from Table 1.3 and 1.4 that the outputs from the two tables are exactly same. Therefore, Table 1.3 indicates that if an antecedent is satisfied then it would be able to determine the consequent. Otherwise, the consequent would be non-deterministic. In this thesis, the concept of implication is typically used in the form of if-then rules for predicting classes. The concept of if-then rules is introduced in Section 1.3.3.

Table 1.4 Negation Truth Table 4

a	b	¬ a	F
0	0	1	1
0	1	1	1
1	0	0	0
1	1	0	1

Besides, negation and equivalence are actually not applied to the research methodology in this thesis. Therefore, they are not introduced in detail here and more details about these two concepts are available in (Simpson, 2013; Ross, 2004).

Set theory is another part of discrete mathematics as mentioned earlier. A set is defined as a collection of elements. The elements maybe numbers, points and names etc., which are not ordered nor repetitive, i.e. the elements can be stored in any order and are distinct each other. As introduced in (Schneider, 2001; Aho, Hopcraft, & Ullman, 1983), an element has a membership in a set, which is denoted by 'e ∈ S' and pronounced by that element 'e' belongs to set 'S' or denoted by 'e ∉ S' and pronounced by that element 'e' does not belong to set 'S'. In this thesis, set theory is used in the management of data and rules, which are

referred to as a data set and a rule set respectively. A data set is used to store data and each element represents a data point. In this thesis, data points are usually referred to as instances. A rule set is used to store rules and each element represents a rule. In addition, a set can have a number of subsets depending on the number of elements. The maximum number of subsets for a set would be 2^n, where n is the number of elements in the set. There are also some operations between sets such as union, intersection and difference, which are not actually applied to the research methodology in this thesis. Therefore, the concepts relating to these operations are not introduced here and more details are available in (Schneider, 2001).

On the other hand, there may be relations existing between sets. A binary relation exists when two sets are related. For example, there are two sets denoted as 'Student' and 'Course' respectively. In this context, there would be a mapping from students and courses and each mapping is known as an ordered pair. In the University of Portsmouth, each student can register on one course only but a course could have many students or no students, which means that each element in the set 'Student' is only mapped to one element in the set 'Course' but an element in the latter set may be mapped from many elements in the former set. Therefore, this is a many-to-one relation. This type of relations are also known as functions. In contrast, if the university regulations allow that a student may register on more than one course, the relation would become many-to-many and is not considered as a function any more. Therefore, a function is generally defined as a many-to-one relation. In the above example, the set 'Student' is regarded as the domain and the set 'Course' as range. In this thesis, each rule in a rule set actually acts as a particular function to reflect the mapping from an input space (domain) to an output space (range).

Graph theory is also a part of discrete mathematics as mentioned earlier in this subsection. It is popularly used in data structures such as binary search trees and directed or undirected graphs. A tree typically consists of a root node and some internal nodes as well as some leaf nodes as illustrated in Fig.1.1. In this figure, node A is the root node of the tree; node B and C are two internal nodes; and node D, E, F and G are four leaf nodes. A tree could be seen as a top-down directed graph. This is because the search strategy applied to trees is in the top-down approach from the root node to the leaf nodes. The search strategy could be divided into two categories: depth first search and breadth first search. In the former strategy, the search is going through in the approach: A→B→D→E→C→F→G. In contrast, in the latter strategy, the search would be in the approach: A→C→C→D→E→F→G. In this thesis, the tree structure is applied to the concept of decision trees to graphically represent a set of rules. More details about this are presented in Chapter 2.

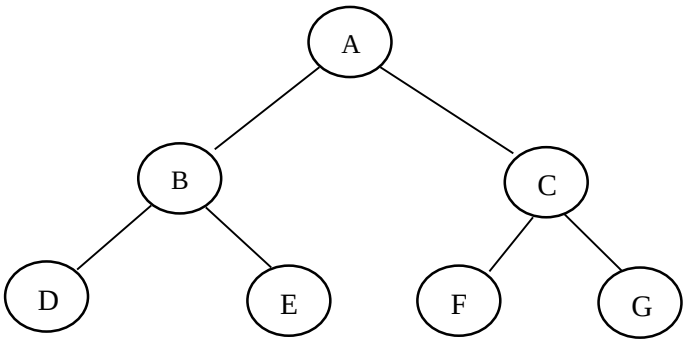

Fig.1.1 Example of Tree Structure

In contrast to trees, there is also a type of horizontally directed graphs in one/two way(s) as illustrated in Fig.1.2 and Fig.1.3. For example, a feed-forward neural network is seen as a one way directed graph and a feedback neural network as a two way directed graph.

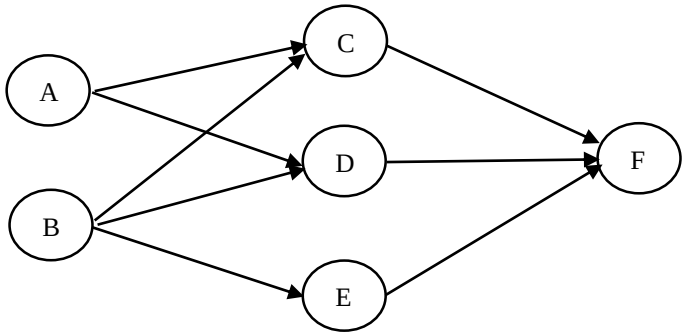

Fig.1.2 Example of one way directed graph

In a directed graph, it could be judged on the reachability between nodes depending on the existence of connections. For example, looking at Fig.1.2, it can only be judged that it is reachable from node A to node C but unreachable in the opposite way. This is because there is only a one way connection from node A to node C. In contrast, there is a two way connection between node A and node C through looking at Fig.1.3. Therefore, it can be judged that it is reachable between the two nodes, i.e. it is reachable in both ways (A→C and C→A). In this thesis, the concept of directed graphs is applied to a special type of rule representation known as rule based network for the purpose of predictive simulation. Related details are presented in Chapter 2 and 3.

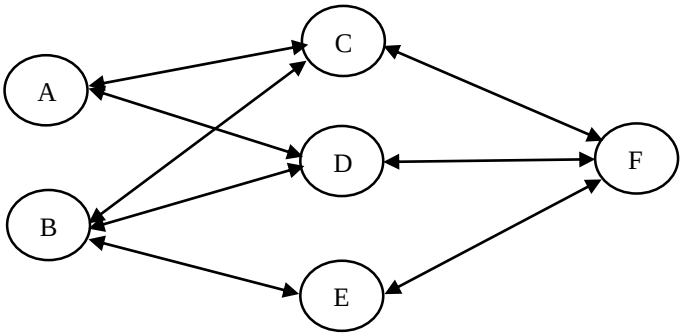

Fig.1.3 Example of two way directed graph

In addition, a graph could also be undirected, which means that in a graphical representation the connections between nodes would become undirected. This concept is also applied to network based rule representation but the difference to application of directed graphs is that the purpose is for knowledge representation. More details about this are introduced in Chapter 3.

1.3.2 Probability Theory

Probability theory is another branch of mathematics, which is a concept involved in all type of activities (Murdoch & Barnes, 1973). Probability is seen as a measure of uncertainty for a particular event. In general, there are two extreme cases. The first one is that if an event A is exact, then the probability of the event, denoted by P (A), is equal to 1. The other case is that if the event is impossible, then the corresponding probability would be equal to 0. In reality, most events are supposed to be random and their corresponding probabilities would be

ranged between 0 and 1. These events typically include independent events and mutually exclusive events.

Independent events generally mean that for two or more events the occurrence of one does not affect that of the other(s). However, the events would be mutually exclusive if the occurrence of one event results in that the other(s) will exactly not occur. In addition, there are also some events that are neither independent nor mutually exclusive. In other words, the occurrence of one event may result in the occurrence of the other(s) with a probability. The corresponding probability is referred to as conditional probability, which is denoted by P (A|B). The P (A|B) is pronounced as that the probability of A given B as a condition. According to Bayes's theorem (Michiel, 2001), P (A) is seen as a prior probability, which indicates the pre-degree of certainty for event A, and P (A|B) as a posterior probability, which indicates the post-degree of certainty for event A after taking into consideration event B. In this thesis, the concept of probability theory introduced above is related to the essence of the methods of rule generation introduced in Chapter 2 and 3, namely ID3, Prism and Information Entropy Based Rule Generation. In addition, the concept is also related to an information theoretic measure called J-measure.

Probability theory is typically jointly used with statistics. For example, it can well contribute to the theory of distribution (Murdoch & Barnes, 1973) with respect to probability distribution. As mentioned in the book (Murdoch & Barnes, 1973), a probability distribution is often transformed from a frequency distribution. When different events have the same probability, the probability distribution is in the case of normal distribution. In the context of statistics, normal distribution occurs while all possible outcomes have the same frequency resulting from a sampling based investigation. Probability distribution also helps predict the expected outcome out of all the possible ones in a random event. This could be achieved by weighted majority voting, while the random event is discrete, or by weighted averaging, while the event is continuous. In the above context, probability is actually used as the weight and the expected outcome is referred to as mathematical expectation. In addition, the probability distribution also helps measure the approximate distance between the expected outcome and the actual outcome, while the distance among different outcomes is precise such as rating from 1 to 5. This could be achieved by calculating the variance or standard deviation to reflect the volatility with regard to the possible outcome. In this thesis, probability distribution is related to a technique of information theory, which is known as entropy and used as a measure of uncertainty in classification. In addition, the concept on mathematical expectation is used to measure the expected accuracy by random guess in classification and variance/ standard deviation can be used to measure the randomness of an algorithm of ensemble learning.

1.3.3 If-Then Rules

As mentioned in Section 1.1, a rule based system typically consists of a set of if-then rules. Ross (2004) stated that there are many different ways for knowledge representation in the area of artificial intelligence but the most popular one would perhaps be in the form of if-then rules denoted by the expression: IF cause (antecedent) THEN effect (consequent).

The expression above typically indicates the inference that if a condition (cause, antecedent) is known then the outcome (effect, consequent) can be derived (Ross, 2004). It is introduced in the tutorial (Gegov, 2013) that both the antecedent and the consequent of a rule could be made up by multiple terms (inputs/outputs). In this context, an antecedent with multiple

inputs that are linked by ‘and’ connectives is called conjunctive antecedent whereas the inputs that are linked by ‘or’ connectives would make up a disjunctive antecedent. The same concept is also applied to rule consequent. In addition, it is also introduced in (Gegov, 2013) that rules may be conjunctive, if all of the rules are connected by logical conjunction, or disjunctive, if the rules are connected by logical disjunction. On the other hand, a rule may be inconsistent, which indicates that the antecedent of a rule may be mapped to different consequents. In this case, the rule could be expressed in a form with a conjunctive antecedent and a disjunctive consequent.

In this thesis, if-then rules are used to make predictions in classification tasks. In this context, each of the rules is referred to as a classification rule, which can have multiple inputs but only a single output. In a classification rule, the consequent with a single output represents the class predicted and the antecedent with a single/multiple input(s) represents the adequate condition to have this class predicted. A rule set that is used to predict class consists of disjunctive rules which may be overlapped. This means that different rules may have the same instances covered. However, if the overlapped rules have different consequents (classification), it would raise a problem referred to as conflict of classification. In this case, conflict resolution is required to solve the problem according to some criteria such as weighted voting (Bramer, 2007) and fuzzy inference (Ross, 2004). When a rule is inconsistent, it would result in a clash problem in classification. This is because the prediction of class becomes non-deterministic when this problem arises. More details about conflict resolution and clash handling are presented in Chapter 2 and 3.

Another concept relating to if-then rules is known as a rule base. In general, a rule base consists of a number of rules which have common input and output variables. For example, a rule base has two inputs: x_1 and x_2 and one output y as illustrated by Fig.1.4:

Fig.1.4 rule base with inputs x_1 and x_2 and output y

If x_1, x_2 and y all belong to {0, 1}, the rule base can have up to four rules as listed below:

If x_1=0 and x_2= 0 then y ∈{0, 1}
If x_1=0 and x_2= 1 then y ∈{0, 1}
If x_1=1 and x_2= 0 then y ∈{0, 1}
If x_1=1 and x_2= 1 then y ∈{0, 1}

In practice, a rule base can be used to effectively and efficiently manage rules with respects to their storage and retrieval. For example, if a particular rule is searched for, it could be efficiently retrieved by locating at the rule base in which the rule is restored. This is a significant difference to a rule set for retrieval purpose. As mentioned earlier in this section, a set is used to restore a collection of elements which are not ordered nor grouped properly. From this point of view, it is not efficient to look for a particular rule in a rule set. The only way to deal with that is to linearly go through the rules one by one in the rule set until the target rule is found. In the worst case, it may be required to go through the whole set due to that the target rule is restored as the last element of the rule set. Therefore, the use of rule

base would improve the efficiency in predicting classes on unseen instances in testing stage. More details about the use of rule base are introduced in Chapter 3 and 6.

1.3.4 Algorithms

Aho et al (1983) defined that *"algorithm is a finite sequence of instructions, each of which has a clear meaning and can be performed with a finite amount of effort in a finite length of time"*. In general, an algorithm acts as a step by step procedure for problem solving. An algorithm may have no inputs but must have at least one output with regard to solving a particular problem. In practice, a problem can usually be solved by more than one algorithm. In this sense, it is necessary to make comparison between algorithms to find the one which is more suitable to a particular problem domain. An algorithm could be evaluated against the following aspects:

- Accuracy, which refers to the correctness in terms of correlation between inputs and outputs.
- Efficiency, which refers to the computational cost required.
- Robustness, which refers to the tolerance to incorrect inputs.
- Readability, which refers to the interpretability to people.

Accuracy would usually be the most important factor in determining whether an algorithm is chosen to solve a particular problem. It can be measured by providing the inputs and then checking the outputs.

Efficiency is another important factor to measure if the algorithm is feasible in practice. This is because if an algorithm is computationally expensive then the implementation of the algorithm may be crashed on a hardware device. Efficiency of an algorithm can usually be measured by checking the time complexity of the algorithm in theoretical analysis. In practice, it is usually measured by checking the actual runtime on a machine.

Robustness can usually be measured by providing a number of incorrect inputs and checking the extent to which the accuracy with regard to outputs is affected.

Readability is also important especially when an algorithm is theoretically analysed by experts or read by practitioners for application purpose. This problem can usually be solved by choosing a suitable representation for the algorithm to make it easier to read. Some existing representations include flow chart, UML activity diagram, pseudo code, text and programming language.

This thesis addresses the four aspects in Chapter 2, 3 and 4 in the way of theoretical analysis and empirical validation as well as algorithm representation with regard to algorithm analysis.

1.3.5 Logic

It is stated in the literature (Ross, 2004) that logic is a small part of capability of human reasoning, which is used to assist people in making decisions or judgements. Section 1.3.1 introduces mathematical logic which is also referred to as Boolean logic in computer science. As mentioned in Section 1.3.1, in the context of Boolean logic, each variable is only assigned a binary truth value: 0 (false) or 1 (true). It indicates that reasoning and judgement are made under certainty resulting in deterministic outcomes. From this point of view, this type of logic is also referred to as deterministic logic. However, in reality, people usually

can only make decisions, judgement and reasoning under uncertainty. Therefore, the other two types of logic, namely probabilistic logic and fuzzy logic, are used more popularly, both of which can be seen as an extension of deterministic logic. The main difference is that the truth value is not binary but continuous between 0 and 1. The truth value implies a probability of truth between true and false in probabilistic logic and a degree of that in fuzzy logic. The rest of the subsection describes the essence of the three types of logic and their difference as well as the association to the concept of rule based systems.

Deterministic logic deals with any events under certainty. For example, when applying deterministic logic for the outcome of an exam, it could be thought that a student will exactly pass or fail a unit. In this context, it means the event is exact to happen.

Probabilistic logic deals with any events under probabilistic uncertainty. For the same example about exams, it could be thought that a student has 80% chances to pass, i.e. 20% chances to fail, for a unit. In this context, it means the event is highly probable to happen.

Fuzzy logic deals with any events under non-probabilistic uncertainty. For the same example about exams, it could be thought that a student has 80% factors of passing, i.e. 20% factors of failing, for a unit with regard to all factors in relation to the exam. In this context, it means the event is highly likely to happen.

A scenario is used to illustrate the above description as follows: students need to attempt the questions on four topics in a maths test. They can pass if and only if they pass all of the four topics. For each of the topics, they have to get all answers correct to pass. The exam questions do not cover all aspects that students are taught but should not be outside the domain nor be known to students. Table 1.5 reflects the depth of understanding of a student in each of the topics.

Table 1.5 Depth of understanding for each topic

Topic 1	Topic 2	Topic 3	Topic 4
80%	60%	70%	20%

In this scenario, deterministic logic is not applicable because it is never deterministic with regard to the outcome of the test. In other words, it is not an exact event that a student will pass or not.

In probabilistic logic, the depth of understanding is supposed to be the probability of the student passing. This is because of the assumption that the student would exactly gain full marks for which questions the student is able to work out. Therefore, the probability of passing would be: $p= 0.8\times0.6\times0.7\times0.2=0.0672$.

In fuzzy logic, the depth of understanding is supposed to be the weight of the factors for passing. For example, for topic 1, the student has 80% factors for passing but it does not imply that the student would have 80% chance to pass. This is because in reality the student may feel unwell mentally, physically and psychologically. All of these issues may make it possible that the student makes mistakes as a result of that the student fails to gain marks for which questions that he/she is able to work out. The fuzzy truth value of passing is 0.2= min (0.8, 0.6, 0.7, 0.2). In this context, the most likely outcome for failing would be that the student only fails one topic resulting in a failure of maths. The topic 4 would be obviously the one which is most likely to fail with the fuzzy truth value 0.8. In all other cases, the

fuzzy truth value would be less than 0.8. Therefore, the fuzzy truth value for passing is 0.2=1-0.8.

In the context of rule base systems, a deterministic rule based system would have each rule either fire or not. If it fires, the consequence would be deterministic. A probabilistic rule based system would have a firing probability for each rule. The consequence would be probabilistic depending on posterior probability of it given specific antecedents. A fuzzy rule based system would have a firing strength for each rule. The consequence would be weighted depending on the fuzzy truth value of the most likely outcome. More details about the concepts on rule based systems above are presented in Chapter 3.

1.3.6 Statistical Measures

In this thesis, some statistical measures are used as heuristics for development of rule learning algorithms and evaluation of rule quality. This subsection presents several measures namely entropy, J-measure, confidence, lift and leverage.

Entropy is introduced by Shannon in (Shanno, 1948), which is an information theoretic measure of uncertainty. Entropy *E* can be calculated as illustrated in equation (1):

$$E = -\sum_{i=0}^{n} p_i \cdot \log_2 p_i \quad (1)$$

where *p* is read as probability that an event occurs and *i* is the index of the corresponding event.

J-measure is introduced in (Smyth & Goodman, 1991), which is an information theoretic measure of average information content of a single rule. J-measure is essentially the product of two terms as illustrated in equation (2):

$$J(Y, X = x) = P(x) \cdot j(Y, X = x) \quad (2)$$

where the first term *P(x)* is read as the probability that the rule antecedent (left hand side) occurs and considered as a measure of simplicity (Smyth & Goodman, 1992). In addition, the second term is read as j-measure, which is first introduced in (Blachman, 1968) but later modified in (Smyth & Goodman, 1992) and considered as a measure of goodness of fit of a single rule (Smyth & Goodman, 1992). The j-measure is calculated as illustrated in equation (3):

$$j(Y, X = x) = P(y \mid x) \cdot \log_2\left(\frac{P(y \mid x)}{P(y)}\right) + (1 - P(y \mid x)) \cdot \log_2\left(\frac{1 - P(y \mid x)}{1 - P(y)}\right) \quad (3)$$

where *P(y)* is read as prior probability that the rule consequent (right hand side) occurs and *P(y |x)* is read as posterior probability that the rule consequent occurs given the rule antecedent as the condition.

In addition, j-measure has an upper bound referred to as jmax as indicated in (Smyth & Goodman, 1992) and illustrated in equation (4):

$$j(Y, X = x) \leq \max(P(y \mid x) \cdot \log_2(\frac{1}{P(y)}), (1 - P(y \mid x)) \cdot \log_2(\frac{1}{1 - P(y)})) \quad (4)$$

However, if it is unknown to which class the rule is assigned as its consequent, then the j-measure needs to be calculated by taking into account all possible classes as illustrated in equation (5):

$$j(Y, X = x) = \sum_{i=0}^{n} P(y_i \mid x) \cdot \log_2(\frac{P(y_i \mid x)}{P(y_i)}) \quad (5)$$

In this case, the corresponding jmax is calculated in the way illustrated in equation (6):

$$j(Y, X = x) \leq \max_i(P(y_i \mid x) \cdot \log_2(\frac{1}{P(y_i)})) \quad (6)$$

Confidence is introduced in (Agrawal, Imielinski, & Swami, 1993), which is considered as predictive accuracy of a single rule, i.e. to what extent the rule consequent is accurate while the rule antecedent is met. The confidence is calculated as illustrated in equation (7):

$$Conf = \frac{P(x, y)}{P(x)} \quad (7)$$

where *P(x, y)* is read as the joint probability that the antecedent and consequent of a rule both occur and *P(x)* is read as prior probability as same as used in J-measure above.

Lift is introduced in (Brin, Motwani, Ullman, & Tsur, 1997), which measures to what extent the actual frequency of joint occurrence for the two events X and Y is higher than expected if X and Y are statistically independent (Hahsler, 2015). The lift is calculated as illustrated in equation (8):

$$Lift = \frac{P(x, y)}{P(x) \cdot P(y)} \quad (8)$$

where *P(x, y)* is read as the joint probability of x and y as same as mentioned above and *P(x)* and *P(y)* are read as the coverage of rule antecedent and consequent respectively.

Leverage is introduced in (Piatetsky-Shapiro, 1991), which measures the difference between the actual joint probability of x and y and the expected one (Hahsler, 2015). The leverage is calculated as illustrated in equation (9):

$$Leverage = P(x, y) - P(x) \cdot P(y) \quad (9)$$

where P(x, y), P(x) and P(y) are read as same as in equation (8) above.

More detailed overview of these statistical measures can be found in (Tan, Kumar, & Srivastava, 2004; Geng & Hamilton, 2006). In this thesis, entropy is used as a heuristic for rule generation and J-measure is used for both rule simplification and evaluation. In addition, confidence, lift and leverage are all used for evaluation of rule quality. More details on this are described in Chapter 2 and 3.

1.4 Chapter Overview

This thesis consists of six main chapters namely, introduction, literature review, research methodology, quantitative validation, qualitative evaluation and conclusion. The rest of the thesis is organized as follows:

Chapter 2 introduces some existing methods and techniques in relation to rule based classification including generation, simplification and representation of classification rules and identifies their strengths and limitations to show the necessities for the development of advanced methods and techniques as mentioned in the research objectives in Chapter 1.

Chapter 3 introduces a unified theoretical framework for design of rule based classification systems as well as advanced methods and techniques which are developed in the PhD thesis and can be successfully integrated into the framework. This chapter also introduces two advanced frameworks for ensemble learning. Besides, all these methods and techniques are discussed critically and comparatively with respects to improvements in performance of classification.

Chapter 4 introduces the ways in which the research methodology is validated and specifies the existing methods and techniques with which the proposed ones are compared. This chapter also describes the data sets used for validation of proposed methods and techniques and justifies the suitability of the chosen data sets. The results are presented and discussed comparatively against existing rule based methods and techniques reviewed in Chapter 2.

Chapter 5 evaluates the completed work against the objectives mentioned in Chapter 1. This chapter also critically reflects the strengths and limitations of the completed work towards identification of further directions. Contributions in the thesis are also highlighted.

Chapter 6 describes the contributions in this thesis in detail with respects to theoretical significance, practical importance, methodological impact and philosophical aspects. Future work is also highlighted towards further improvement of the research methodology with references to the evaluation against objectives of this research mentioned in Chapter 5.

Chapter 2 Literature Review

2.1 Introduction

As mentioned in Chapter 1, the focus of this thesis is on the description of rule based classification and ensemble learning as well as the discussion on some existing methods and techniques. This chapter describes some existing methods and techniques in relation to rule based classification including rule generation, rule simplification and rule representation. This chapter also includes the description of ensemble learning concepts and some of ensemble rule based methods in this context.

2.2 Single Rule Based Systems

As mentioned in Chapter 1, most methods for generation of classification rules may result in overfitting of training data, which means that the constructed classifier consists of a large number of complex rules and may lower both classification accuracy and computational efficiency. This makes it necessary to simplify rules for reduction of overfiting. In addition, the same classifier may perform different levels of efficiency in testing stage if the classifier is represented in different structures. This makes it relevant to find a suitable representation for a particular classifier. Therefore, this section is subdivided into three subsections in order to introduce some existing methods or techniques in three different aspects, namely rule generation, rule simplification and rule representation.

2.2.1 Rule Generation Methods

As mentioned in Chapter 1, rule generation can be done by two approaches. The two approaches are usually referred to as 'divide and conquer' and 'separate and conquer' respectively. The former approach can generate a rule set in the form of a decision tree and thus is also called Top-Down Induction of Decision Trees (TDIDT). However, it has been criticised by Cendrowska (1987) as a major cause of overfitting. Therefore, the latter approach is recommended to be used instead and Prism is a representative method developed for this motivation. This subsection focuses on the introduction on TDIDT and Prism methods.

Decision trees have been a popular method for generation of classification rules and they are based on the fairly simple but powerful TDIDT algorithm (Bramer, 2007). The basic idea of this algorithm can be illustrated in Fig.2.1.

Input: A set of training instances, attribute A_i, where is the index of the attribute A, value V_j, where j is the index of the value V
Output: A decision tree.

if the stopping criterion is satisfied **then**
  create a leaf that corresponds to all remaining training instances
else
  choose the best (according to some heuristics) attribute A_i
  label the current node with A_i
  for each value V_j of the attribute A_i **do**
   label an outgoing edge with value V_j
   recursively build a subtree by using a corresponding subset of training instances
  end for
end if

Fig.2.1 Decision tree learning algorithm (Kononenko & Kukar, 2007)

One popular method of attribute selection for 'else' branch illustrated in Fig.2.1 is based on average entropy of attribute (Bramer, 2007), which is to select the attribute that can minimize the value of entropy for the current subset being separated, thus maximize information gain.

As mentioned in Chapter 1, entropy is a measure of the uncertainty in discriminating multiple classifications. It can be calculated in the following way (Bramer, 2007):

- To calculate the entropy for each attribute-value pair in the way that the entropy of training set is denoted by E and is defined by the formula illustrated in equation (1) (See Section 1.3.6) summed over the classes for which $p_i \neq 0$ (p denotes the probability of class i) if there are k classes.
- To calculate the weighted average for entropy of resulting subsets.

For the conditional entropy of an attribute-value pair, p_i denotes the posterior probability for class i when given the particular attribute-value pair as a condition. On the other hand, for initial entropy, the p_i denotes the priori probability for class i. The information gain is calculated by subtracting the initial entropy from the average entropy of a given attribute.

ID3 is an example of TDIDT, which bases attribute selection on entropy. The procedure of ID3 is illustrated below using a data set that is named contact-lenses (Cendrowska, 1987) and retrieved from UCI repository (Lichman, 2013). The details of this data set are illustrated in Table 2.1.

Table 2.1. Contact lenses data

age	prescription	astigmatic	Tear production rate	class
young	myope	no	reduced	no lenses
young	myope	no	normal	soft lenses
young	myope	yes	reduced	no lenses
young	myope	yes	normal	hard lenses
young	hypermetrope	no	reduced	no lenses
young	hypermetrope	no	normal	soft lenses
young	hypermetrope	yes	reduced	no lenses
young	hypermetrope	yes	normal	hard lenses
pre-presbyopic	myope	no	reduced	no lenses
pre-presbyopic	myope	no	normal	soft lenses
pre-presbyopic	myope	yes	reduced	no lenses
pre-presbyopic	myope	yes	normal	hard lenses
pre-presbyopic	hypermetrope	no	reduced	no lenses
pre-presbyopic	hypermetrope	no	normal	soft lenses
pre-presbyopic	hypermetrope	yes	reduced	no lenses
pre-presbyopic	hypermetrope	yes	normal	hard lenses
presbyopic	myope	no	reduced	no lenses
presbyopic	myope	no	normal	soft lenses
presbyopic	myope	yes	reduced	no lenses
presbyopic	myope	yes	normal	hard lenses
presbyopic	hypermetrope	no	reduced	no lenses
presbyopic	hypermetrope	no	normal	soft lenses
presbyopic	hypermetrope	yes	reduced	no lenses
presbyopic	hypermetrope	yes	normal	hard lenses

As mentioned above, ID3 makes attribute selection based on entropy. Therefore, it is necessary to create a frequency table for each attribute.

Table 2.2. Frequency table for age

Class label	age= young	age= pre-presbyopic	age= presbyopic
No lenses	4	4	4
Soft lenses	2	2	2
Hard lenses	2	2	2
Total	8	8	8

Table 2.3. Frequency table for spectacle prescription

Class label	prescription= myope	prescription= hypermetrope
No lenses	6	6
Soft lenses	3	3
Hard lenses	3	3
Total	12	12

Table 2.4. Frequency table for astigmatic

Class label	astigmatic=yes	astigmatic=no
No lenses	6	6
Soft lenses	6	0
Hard lenses	0	6
Total	12	12

For ID3, the average entropy for each of the attributes is in the following:

$$E(\text{age}) = 1/3\times(-(1/2)\times\log_2(1/2) - (1/4)\times\log_2(1/4) - (1/4)\times\log_2(1/4)) + 1/3\times(-(1/2)\times\log_2(1/2) - (1/4)\times\log_2(1/2) - (1/4)\times\log_2(1/2)) + 1/3\times(-(1/2)\times\log_2(1/2) - (1/4)\times\log_2(1/2) - (1/4)\times\log_2(1/2)) = 3/2$$

$$E(\text{prescription}) = 1/2\times(-(1/2)\times\log_2(1/2) - (1/4)\times\log_2(1/4) - (1/4)\times\log_2(1/4)) + 1/2\times(-(1/2)\times\log_2(1/2) - (1/4)\times\log_2(1/4) - (1/4)\times\log_2(1/4)) = 3/2$$

$$E(\text{astigmatic}) = 1/2\times(-(1/2)\times\log_2(1/2) - (1/2)\times\log_2(1/2)) + 1/2\times(-(1/2)\times\log_2(1/2) - (1/2)\times\log_2(1/2)) = 1$$

$$E(\text{tear production rate}) = 1/2\times(-(1/2)\times\log_2(1/2) - (1/2)\times\log_2(1/2)) + 1/2\times(-1\times\log_2(1)) = ½$$

As E (tear production rate) is the minimum, the data set illustrated in Table 2.1 is split on the attribute *tear production rate*. This results in two subsets illustrated in Table 2.6 and 2.7.

It can be seen from Table 2.6 that all instances belong to the class *no lenses*, which indicates there is no uncertainty remaining in the subset. Therefore, it results in an incomplete decision tree as illustrated in Fig.2.2.

Table 2.6 Subset 1 for contact lenses data

age	prescription	astigmatic	Tear production rate	class
young	myope	no	reduced	no lenses
young	myope	yes	reduced	no lenses
young	hypermetrope	no	reduced	no lenses
young	hypermetrope	yes	reduced	no lenses
pre-presbyopic	myope	no	reduced	no lenses
pre-presbyopic	myope	yes	reduced	no lenses
pre-presbyopic	hypermetrope	no	reduced	no lenses
pre-presbyopic	hypermetrope	yes	reduced	no lenses
presbyopic	myope	no	reduced	no lenses
presbyopic	myope	yes	reduced	no lenses
presbyopic	hypermetrope	no	reduced	no lenses
presbyopic	hypermetrope	yes	reduced	no lenses

Table 2.7 Subset 2 for contact lenses data

age	prescription	astigmatic	Tear production rate	class
young	myope	no	normal	soft lenses
young	myope	yes	normal	hard lenses
young	hypermetrope	no	normal	soft lenses
young	hypermetrope	yes	normal	hard lenses
pre-presbyopic	myope	no	normal	soft lenses
pre-presbyopic	myope	yes	normal	hard lenses
pre-presbyopic	hypermetrope	no	normal	soft lenses
pre-presbyopic	hypermetrope	yes	normal	hard lenses
presbyopic	myope	no	normal	soft lenses
presbyopic	myope	yes	normal	hard lenses
presbyopic	hypermetrope	no	normal	soft lenses
presbyopic	hypermetrope	yes	normal	hard lenses

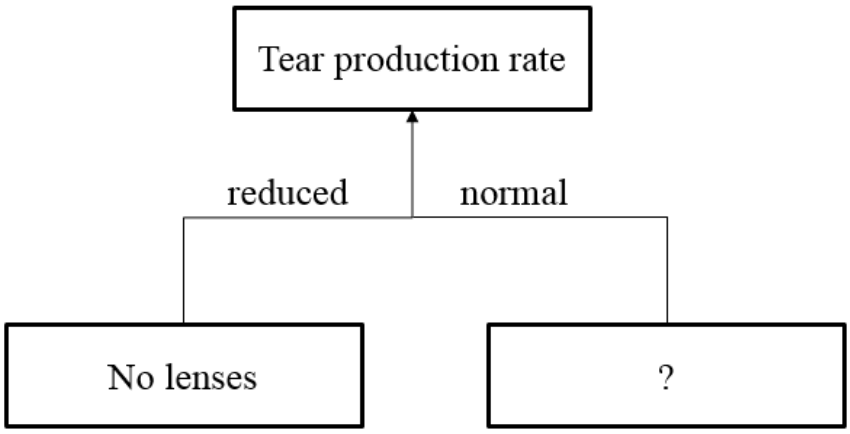

Fig.2.2 Incomplete decision tree comprising attribute *Tear production rate*

The left branch comprising *tear production rate = reduced* is terminated by giving a leaf node labeled *no lenses*. The right branch comprising *tear production rate = normal* is still not terminated, which means it is required to select another attribute other than *tear production rate* to be split at the node which is labeled with a question mark. For this, it is needed to create a frequency table for each of the rest of the attributes namely age, prescription and astigmatic from the subset 2 for the data set illustrated in Table 2.

Table 2.8 Frequency table for age at the second iteration

Class label	age= young	age= pre-presbyopic	age= presbyopic
No lenses	0	0	0
Soft lenses	2	2	2
Hard lenses	2	2	2
Total	4	4	4

Table 2.9 Frequency table for spectacle prescription at the second iteration

Class label	prescription= myope	prescription= hypermetrope
No lenses	0	0
Soft lenses	3	3
Hard lenses	3	3
Total	6	6

Table 2.10 Frequency table for astigmatic at the second iteration

Class label	astigmatic=yes	astigmatic=no
No lenses	0	0
Soft lenses	0	6
Hard lenses	6	0
Total	6	6

According to Table 3.8- 3.10, the average entropy for each of the attributes is shown below.

$$E(\text{age}) = 1/2\times(-(1/2)\times\log_2(1/2) - (1/2)\times\log_2(1/2)) + 1/2\times(-(1/2)\times\log_2(1/2) - (1/2)\times\log_2(1/2)) = 1$$

$$E(\text{prescription}) = 1/2\times(-(1/2)\times\log_2(1/2) - (1/2)\times\log_2(1/2)) + 1/2\times(-(1/2)\times\log_2(1/2) - (1/2)\times\log_2(1/2)) = 1$$

$$E(\text{astigmatic}) = 1/2\times(-1\times\log_2(1)) + 1/2\times(-1\times\log_2(1)) = 0$$

Therefore, the attribute *astigmatic* is selected to be split at the node that is labeled with a question mark in Fig.2.2, which means that the subset 2 for the data set illustrated in Table 2.7 is split on the *astigmatic* attribute resulting in two further subsets.

Table 2.11 Subset 2.1 for contact lenses data

age	prescription	astigmatic	Tear production rate	class
young	myope	no	normal	soft lenses
young	hypermetrope	no	normal	soft lenses
pre-presbyopic	myope	no	normal	soft lenses
pre-presbyopic	hypermetrope	no	normal	soft lenses
presbyopic	myope	no	normal	soft lenses
presbyopic	hypermetrope	no	normal	soft lenses

It is clear that both subsets illustrated above have all instances belong to the same class and thus remains no uncertainty. The complete decision tree is generated as illustrated in Fig.2.3.

Table 2.12 Subset 2.2 for contact lenses data

age	prescription	astigmatic	Tear production rate	class
young	myope	yes	normal	hard lenses
young	hypermetrope	yes	normal	hard lenses
pre-presbyopic	myope	yes	normal	hard lenses
pre-presbyopic	hypermetrope	yes	normal	hard lenses
presbyopic	myope	yes	normal	hard lenses
presbyopic	hypermetrope	yes	normal	hard lenses

As mentioned above, decision tree representation is a major cause of overfitting. This is due to a principal drawback on replicated sub-tree problem that is identified in (Cendrowska, 1987) and is illustrated in Fig.2.4.

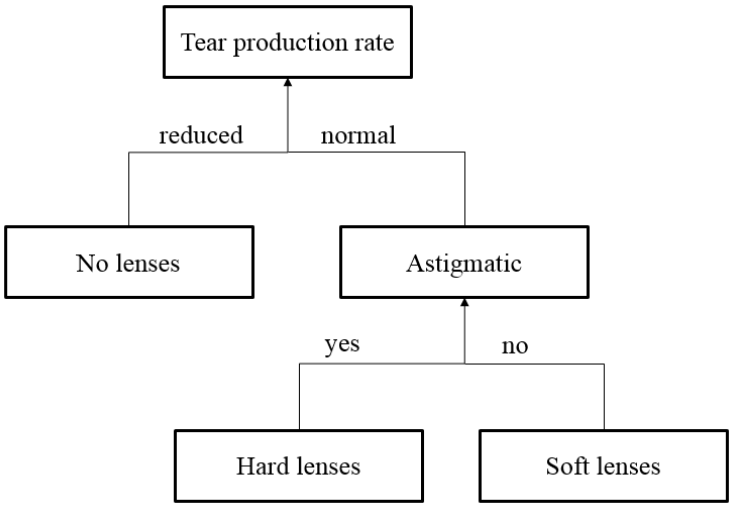

Fig.2.3 Complete decision tree

As summarized by Han and Kamber (2006), decision tree learning is so popular due to the following reasons:

Firstly, the generation of decision trees does not need any prior knowledge in a domain nor parameters setting. Therefore, decision tree learning is seen as an appropriate approach for knowledge discovery.

Secondly, the decision tree learning algorithms are able to effectively deal with training data in high dimensionality.

Thirdly, the decision tree representation is interpretable. In other words, knowledge extracted from a decision tree is easily communicated to people.

Fourthly, the training by the induction algorithm is not expensive and the prediction by a decision tree classifier is straightforward and efficient.

Fifthly, the decision tree learning algorithms can generate accurate classifiers in general.

Finally, the learning algorithm is not domain dependent but data dependent. In this context, the decision tree learning algorithm can be used broadly in many different application areas

such as medicine, finance and biology. However, the performance in a particular task is highly dependent on the suitability to the data used.

Although decision tree learning has some advantages listed in (Han & Kamber, 2006) and described above, it was pointed out in (Cendrowska, 1987; Deng, 2012) that decision tree learning is difficult to manipulate for expert systems as it is required to examine the whole tree in order to extract rules about a single classification. It has been partially solved by converting a tree to a set of individual rules but there are some rules that are not easily fit into a tree structure as is the replicated sub-tree problem mentioned above. In a medical diagnosis system, this problem may lead to unnecessary surgery (Cendrowska, 1987; Deng, 2012). The reasons identified in (Cendrowska, 1987; Deng, 2012) are the following:

- The decision tree is attribute oriented.
- Each iteration in the generation process chooses the attribute on which to be split aiming at minimizing the average entropy, i.e. measuring the average uncertainty. However, this does not necessarily mean that the uncertainty for each rule is reduced.
- An attribute might be highly relevant to one particular classification but irrelevant to the others. Sometimes only one value of an attribute is relevant.

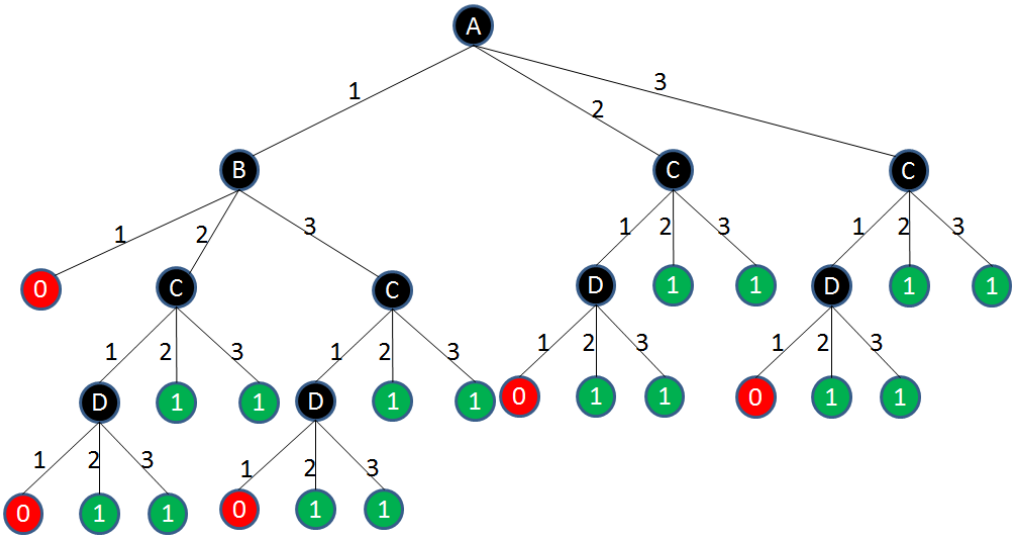

Fig.2.4 Cendrowska's replicated subtree example

As the presence of the above problem, Prism method was introduced in the paper (Cendrowska, 1987) and primarily aimed at avoiding the replicated sub-tree problem. The basic procedure of the underlying Prism algorithm is illustrated in Fig.2.5.

On the other hand, the original Prism algorithm does not take clashes into account, i.e. a set of instances in a subset of a training set that are identical apart from being assigned to different classes but cannot be separated further (Stahl & Bramer, 2011; Stahl & Bramer, 2012). However, the Inducer Software implementation (Bramer, 2005) of Prism can handle clashes and the strategy of handling a clash is illustrated in Fig.2.6.

Input: a training set *T*, a subset $T' \subseteq T$, an instance $t \in T$, dimensionality *d*, an attribute a_x (*x* is the index of *a*), class C_i (*i* is the index of *C*), number of classes *n*
Output: a rule set *RS*, a result set of instances *T''* covered by a rule $R \in RS$
Initialize:
$T'= T$, $T''= \emptyset$, $i=0$;
for $i<n$ **do**
do generate rules for class C_i
while $\exists$ t: t $\in T' \wedge$ t $\notin C_i$ **do**
$x=0$;
while $x < d$ **do**
for each value *v* of a_x **do**
Calculate P(C_i| $a_x= v$);
end for
$x++$;
end while
assign $a_x= v$ to *R* as a rule term, while P(C_i| $a_x= v$) is max;
$\forall$t: $T''\cap\{t\}$, if *t* comprise $a_x= v$;
end while
$RS= RS \cap \{R\}$;
$T'= T'- T''$;
while $\forall$t: t $\in T' \wedge$ t $\notin C_i$
$i++$;
end for

Fig.2.5 Prism Algorithm

Another problem that arises with Prism is tie-breaking, i.e. if there are two or more attribute-value pairs which have equal highest probability in a subset (see step 3 in Fig.2.5). The original Prism algorithm makes an arbitrary choice in step 4 as illustrated in Fig. 1 whereas the Inducer Software makes the choice using the highest total target class frequency (Bramer, 2007).

If a clash occurs while generating rules for class *i*:
1. Determine the majority class for the subset of instances in the clash set.
2. If this majority class is *class i*, then compute the induced rule by assigning all instances in the clash set to class *i*. If it is not, discard the whole rule.
3. If the induced rule is discarded, then all instances that match the target class should be deleted from the training set before the start of the next rule induction. If the rule is kept, then all instances in the clash set should be deleted from the training data.

Fig.2.6 Dealing with clashes in Prism (Stahl & Bramer, 2011; Stahl & Bramer, 2012)

In addition, Bramer pointed out that the original Prism algorithm always deletes instances covered by those rules generated so far and then restores the training set to its original size after the completion of rule generation for class *i* and before the start for class *i+1*. This undoubtedly increases the number of iterations resulting in high computational cost (Bramer, 2000), especially when the training data is very large. For the purpose of increasing the computational efficiency, a modified version of Prism, called PrismTCS, was developed by Bramer (Bramer, 2000). PrismTCS always chooses the minority class as the target class pre-assigned to a rule being generated as its consequence. Besides this, it does not reset the dataset to its original state and introduces an order to each rule according to its importance (Bramer, 2000; Stahl & Bramer, 2011; Stahl & Bramer, 2012). Therefore, PrismTCS is not only faster in generating rules compared with the original Prism, but also provides a similar level of classification accuracy (Bramer, 2000; Stahl & Bramer, 2011; Stahl & Bramer, 2012).

Bramer described in (Bramer, 2000) a series of experiments to compare Prism against decision tree with respect to their performance on a number of datasets. He concluded the following (Bramer, 2007):

- Prism algorithm usually gives classification rules at least as good as those generated from TDIDT algorithm but also outperforms TDIDT in terms of noise tolerance.
- There are generally fewer rules but also fewer terms per rule, which is likely to aid their comprehensibility to domain experts and users. This result would seem that Prism generally gives consistently better accuracy than TDIDT.
- The main difference is that Prism generally prefers to leave a test instance unclassified rather than to assign it an incorrect classification.
- The reasons why Prism is more noise-tolerant than TDIDT may be due to the presence of fewer terms per rule in most cases.
- Prism generally has higher computational efficiency than TDIDT, and the efficiency can be further improved by parallelisation whereas TDIDT cannot achieve.

As mentioned earlier, there is a case that only one value of an attribute is relevant to a particular classification and that ID3 (a version of TDIDT) does not take into consideration. It is pointed out in (Deng, 2012) that the Prism method is attribute-value-oriented and pays much attention to the relationship between an attribute-value pair and a particular classification, thus generating fewer but more general rules than the TDIDT.

Although Prism algorithm has obvious advantages, such as noise tolerance, comparing with TDIDT as mentioned above, the algorithm also has some disadvantages in the aspects of classification conflict, clash handling, underfitting, and computational efficiency.

The original version of Prism may generate a rule set which results in a classification conflict in predicting unseen instances. This can be illustrated by the example below:

Rule 1: If x=1and y=1 then class= a
Rule 2: If z=1 then class= b

What should the classification be for an instance with x=1, y=1 and z=1? One rule gives *class a*, the other one gives *class b*. It is required to have a method choose only one classification to classify the unseen instance (Bramer, 2007). Such a method is known as a conflict resolution strategy. Bramer mentioned in the book (Bramer, 2007) that Prism uses the 'take the first rule that fires' strategy in dealing with the conflict problem and therefore it is required to generate the most important rules first. However, the original Prism cannot actually introduce an order to a rule according to its importance as each of those rules with a different target class is independent of each other. As mentioned above, this version of Prism would restore the training set to its original size after the completion of rule generation for class *i* and before the start for class *i+1*. This indicates that the rule generation for each class may be done in parallel so the algorithm cannot directly rank the importance among rules. Thus the 'take the first rule that fires' strategy may not deal with the classification confliction well. The PrismTCS does not restore dataset to its original state unlike original Prism and thus can introduce the order to a rule according to its importance. This problem is partially resolved but PrismTCS may potentially lead to underfitting of a rule set. PrismTCS always chooses the minority class in the current training set as the target class of the rule being generated. Since the training set is never restored to its original size as mentioned

above, it can be proven that one class could always be selected as the target class until all instances of this class have been deleted from the training set. This is because the instances of this minority class covered by the current rule generated should be removed prior to generating the next rule. This case may result in the case that the majority class in the training set may not be necessarily selected as target class to generate a list of rules until the termination of the whole generation process. In this case, there is not even a single rule having the majority class as its consequence (right hand side of this rule). In some implementations, this problem has been partially solved by assigning a default class (usually majority class) in predicting unseen instances when there is not a single rule that can cover this instance. However, this should be based on the assumption that the training set is complete. Otherwise, the rule set could still underfit the training set as the conditions of classifying instances to the other classes are probably not strong enough. On the other hand, if a clash occurs, both the original Prism and PrismTCS would prefer to discard the whole rule rather than to assign the majority class, which is of higher importance, to the rule. As mentioned above, Prism may generally generate more general and fewer rules than TDIDT algorithms. One reason is potentially due to discarding rules. In addition, the clash may happen in two principal ways as follows:

1) One of the instances has at least one incorrect record for its attribute values or its classification (Bramer, 2007).
2) The clash set has both (or all) instances correctly recorded but it is impossible to discriminate between (or among) them on the basis of the attributes recorded and thus it may be required to examine further values of attributes (Bramer, 2007).

When there is noise present in datasets, Prism is more robust than TDIDT as mentioned above. However, if the reason that a clash occurs is not due to noise and the training set covers a large amount of data, then it could result in serious underfitting of the rule set by discarding rules as it will leave many unseen instances unclassified at prediction stage. The fact that Prism would decide to discard the rules in some cases is probably because it uses the so-called 'from effect to cause' approach. As mentioned above, each rule being generated should be pre-assigned a target class and then the conditions should be searched for specialising the rule antecedent by adding terms until the adequacy conditions are met. Sometimes, it may not necessarily receive adequacy conditions even after all attributes have been examined. This indicates that the current rule covers a clash set that contains instances of more than one class. If the target class is not the majority class, this indicates the search of causes is not successful so the algorithm decides to give up by discarding the incomplete rule and deleting all those instances that match the target class in order to avoid the same case to happen all over again (Stahl & Bramer, 2011; Stahl & Bramer, 2012). This actually not only increases the irrelevant computation cost but also results in underfitting of the rule set.

The above descriptions indicate that the limitations of Prism in the above mentioned aspects could result in significant loss of accuracy and unnecessary computational costs. This thus motivates the development of another rule generation method which is further introduced in Chapter 3.

2.2.2 Rule Simplification Methods

As mentioned in Chapter 1, rule simplification can be achieved by using pruning methods. A special type of pruning algorithms is based on J-measure (Smyth & Goodman, 1991) as

mentioned in Chapter 1. Two existing J-measure based pruning algorithms include J-pruning and Jmax-pruning which have been successfully applied on Prism algorithm. This subsection focuses on descriptions of the two pruning algorithms above.

When a rule is being generated, the J-value (value of J-measure) may increase or decrease after specialising the rule by adding a new term. Both pruning algorithms (J-pruning and Jmax-pruning) expect to find the global maximum of J-value for the rule. Each rule has a complexity degree which is the number of terms. The increase of complexity degree may lead the J-value of this rule to increase or decrease. The aim of pruning algorithms is to find the complexity degree corresponding to the global maximum of J-value as illustrated in Fig.2.7 using a fictitious example.

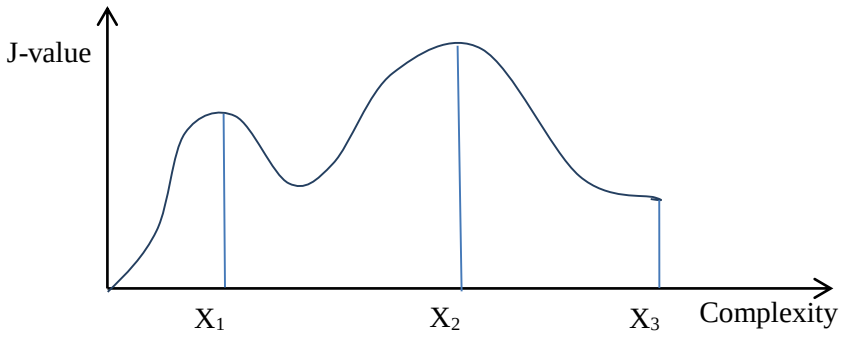

Fig.2.7 Relationship between complexity degree and J-value (case 1)

However, the two pruning algorithms mentioned above search the global maximum of J-value in different strategies:

- J-pruning: monitor the change pattern of J-value and stop rule generation once it goes down. i.e it will stop rule generation when complexity degree is X_1 as illustrated in Fig.2.5 because the J-value is going to decrease afterwards. The final rule generated is with the complexity degree X_1 (having the first X_1 rule terms).
- Jmax-pruning: monitor and record the highest J-value observed so far until the completion of rule's generation, i.e. it will stop rule generation when the complexity is X_3 as illustrated in Fig.2.5 and reduce the complexity degree subsequently until the degree is X_2 by removing those rule terms afterward. The final rule is with the complexity degree X_2.

J-pruning is a pre-pruning method because the pruning action is taken during rule generation. It was developed by Bramer (2002) and its basic idea is illustrated in Algorithm 1.

```
Rule r = new Rule;
Boolean rule_Incomplete = true;
Do While (rule_Incomplete){
    Term t = generate new term;
     compute J_value of r if appending t;
     IF(r.current_J_value > J_value){
       do not append t to r;
       invoke clash handling for r;
       rule_Incomplete = false;
```

```
    }ELSE{
      r.current_J_value = J_value;
      append t to r;}
}
```

Algorithm 1 J-pruning for Prism algorithms

J-pruning achieves relatively good results as indicated in (Bramer, 2002). However, Stahl and Bramer pointed out in the papers (Stahl & Bramer, 2011; Stahl & Bramer, 2012) that J-pruning does not exploit the J-measure to its full potential. This is because this method immediately stops the generation process as soon as the J-measure goes down after a new term is added to the rule as illustrated in Fig.2.7. In fact, it is theoretically possible that the J-measure may go down and go up again after further terms are added to the rule. This indicates that the pruning action may be taken too early. The fact that J-pruning may achieve relatively good results could be explained by the assumption that it does not happen very often that the J-value goes down and then goes up again. A possible case is that there is only one local maximum of J-value as illustrated in Fig.2.8. It also indicates that J-pruning may even result in underfitting due to over-generalised rules. This is because the pruning action may be taken too early resulting in too general rules being generated. The above description motivated the development of a new pruning method, called Jmax-pruning, which was introduced in (Stahl & Bramer, 2011; Stahl & Bramer, 2012), in order to exploit the J-measure to its full potential.

As mentioned in (Stahl & Bramer, 2011; Stahl & Bramer, 2012), Jmax-pruning can be seen as a hybrid between pre-pruning and post-pruning. With regard to each generated rule, each individual rule is actually post-pruned after the completion of the generation for that rule. However, with respect to the whole classifier (whole rule set) it is a pre-pruning approach as there is no further pruning required after all rules have been induced.

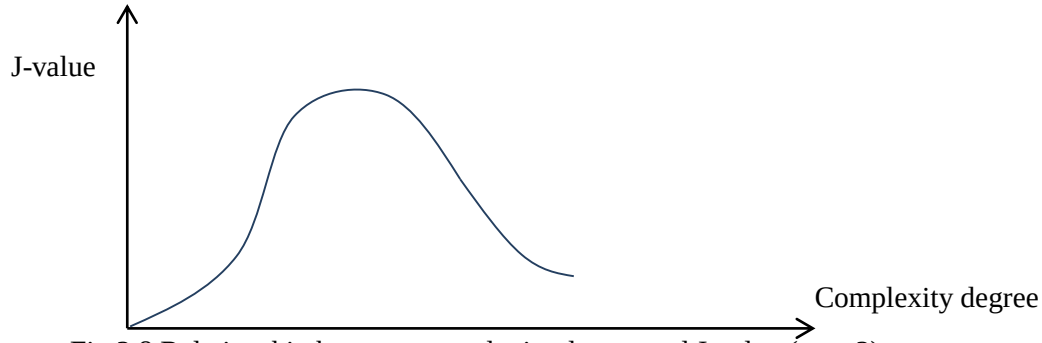

Fig.2.8 Relationship between complexity degree and J-value (case 2)

The basic idea of Jmax-pruning is illustrated in Algorithm 2.

```
Rule r = new Rule;
Boolean rule_Incomplete = true;
term_index = 0;
Do While (rule_Incomplete){
    Term t = generate new term;
    term_index++;
    append t to r;
    compute J_value of r;
```

```
    IF(J_value > best_J_Value){
       best_J_Value = J_Value;
      best_term_index = term_index;
    }
    IF(No more rule terms can be induced){
       cut r back to rule best_term_index;
       invoke clash handling for r;
       rule_Incomplete = false;
    }
}
```

Algorithm 2 Jmax-pruning for Prism algorithms

A series of experiments have shown that Jmax-pruning outperforms J-pruning in some cases (Stahl & Bramer, 2011; Stahl & Bramer, 2012) when there are more than one local maximum and the first one is not the global maximum as illustrated in Fig.2.5. However, it performs the same as J-pruning in other cases (Stahl & Bramer, 2011; Stahl & Bramer, 2012) when there is only one local maximum as illustrated in Fig.2.6 or the first one of local maxima is also the global maximum.

However, Jmax-pruning may be computationally more expensive as each rule generated by this method is post-pruned. The pruning action could be taken earlier during the rule generation and thus speed up the rule generation, especially when Big Data is used for training. This could be achieved by making use of the Jmax value as introduced above.

On the other hand, a special case may need to be taken into account when Prism is used as the classifier. This case is referred to as tie-breaking which is the case that there is more than one global maximum for the J-value during rule generation as illustrated in Fig.2.9.

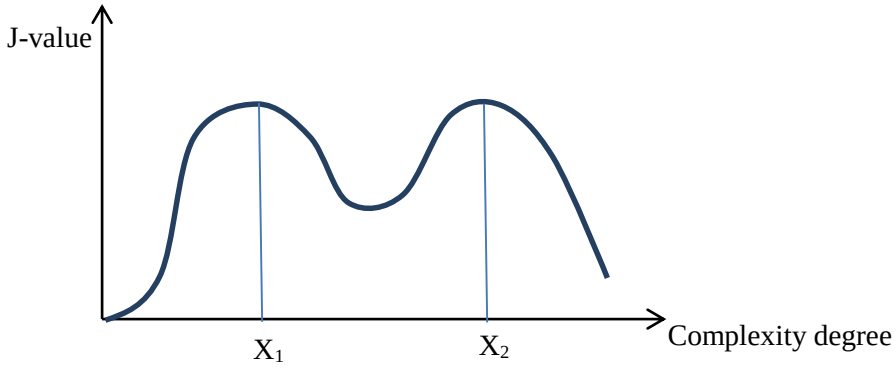

Fig.2.9 Relationship between complexity degree and J-value (case 3)

As mentioned earlier, Prism prefers to discard a rule rather than assign it to a majority class when a clash occurs. Therefore, it would even lead to underfitting of the generated rule set if a pruning method attempts to reduce the overfitting by pruning rules but unfortunately results in discarding rules. If this case is taken into account, it is worth to determine properly which one of the global maximum points to be chosen as the start point of pruning in order to avoid over-discarding rules. In other words, according to Fig.2.9, it needs to determine to choose either X_1 or X_2 as the start point for removing all rule terms afterward.

With regards to this issue, Jmax-pruning always chooses to take X_1 (the first global maximum point) as the start point of pruning and to remove all rule terms generated afterward. It may potentially lead to underfitting as it is possible that the rule is being discarded after handling a clash if X_1 is chosen but is being kept otherwise. In addition, another type of tie-breaking may arise with the case as illustrated below:

Let the current rule's last added rule term be denoted t_i, and the previously added rule term be denoted t_{i-1}. Then a tie break happens if J-value at t_i is less than that at t_{i-1} and Jmax-value at t_i equals J-value at t_{i-1}. It is also illustrated by an example (**Rule 1**) below.

Rule 1: If x=1 and y=1 and z=1 then class=1;

After adding first term:

If x= 1 then class= 1; (J= 0.33, Jmax= 0.55)

After adding second term:

If x=1 and y=1 then class=1; (J= 0.21; Jmax=0.33)

However, the two cases about tie-breaking mentioned above are not very likely to happen in practice but they are still worth to be taken into account. This is because serious underfitting is likely to result from the two cases in case they really happen.

On the basis of above descriptions about limitations of J-pruning and Jmax-pruning, it motivates the development of a new pruning algorithm to overcome the limitations of J-pruning and Jmax-pruning with respects to underfitting and computational efficiency. The new pruning algorithm is further introduced in Chapter 3.

2.2.3 Rule Representation Techniques

One of the biases for rule based methods defined in (Fürnkranz, 1999) is ‘search bias’, which refers to the strategy that the hypothesis is searched. The strategy in searching for rules that fire usually determines the computational efficiency in testing stage for predicting unseen instances. However, the search strategy also strongly depends on the representation of a set of rules. Existing rule representation techniques include decision tree and linear list. The rest of the subsection focuses the discussion on the limitations of the two existing rule representation techniques mentioned above.

As mentioned in Chapter 1, decision tree is an automatic representation for classification rules generated by ‘divide and conquer’ approach. However, the representation is criticized and identified as a major cause of overfitting in (Cendrowska, 1987) as illustrated in Fig.2.4. It is also pointed in (Cendrowska, 1987; Deng, 2012) that it is required to examine the whole tree in order to extract rules about a single classification in the worst case. This drawback on representation makes it difficult to manipulate for expert systems and thus seriously lowers the computational efficiency in predicting unseen instances. For the purpose of predictive modelling, computational efficiency in testing stage is significant especially when the expert systems to be constructed are time critical (Gegov, 2007). In addition, decision trees are often quite complex and difficult to understand (Fürnkranz, 1999). Even if decision trees are simplified by using pruning algorithms, it is still difficult to avoid that the decision trees become too cumbersome, complex and inscrutable to provide insight into a domain for knowledge usage (Quinlan, 1993; Fürnkranz, 1999). This undoubtedly lowers

interpretability of decision trees and is thus a serious drawback for the purpose of knowledge discovery. All of the limitations mentioned above motivate the direct use of 'if-then' rules represented by a linear list structure. However, predicting unseen instances in this representation is run in linear search with the time complexity O (n) while the total number of rule terms is used as the input size (n). This is because list representation works in linear search by going through rule by rule in an outer loop; and by going through term by term for each rule in an inner loop. It implies that it may have to go through the whole rule set to find the first rule firing in the worst case. This would lead to huge computational costs when the representation is used to represent a rule set generated by learning from large training data.

On the basis of above description about limitations of tree and list representation in terms of computational efficiency, it motivates the creation of a new representation of classification rules which performs a level of efficiency higher than linear time in time complexity. This new representation is further described in Chapter 3. On the other hand, with regards to the limitations of the two existing representations in terms of knowledge representation, it also motivates the creation of a new rule representation which performs a better interpretability. Higgins has introduced a representation called rule based network in (Higgins, 1993) as illustrated in Fig.2.10.

In this network, as explained in (Higgins, 1993), each node in the input layer represents an input attribute. Each node in the middle layer represents a rule. The connections between the nodes in the input layer and the nodes in the conjunctive layer indicate to which attributes a specific rule relates. In the output layer, each node represents a class label. The connections between the nodes in the conjunctive layer and the nodes in the output layer reflect the mapping relationships between rule antecedents and classifications (consequents). Each of the connections is also weighted as denoted by w_{mk}, where m is the index of the rule and k is the index of the class. The weight reflects the confidence of the rule for predicting the class given the antecedent of the rule. In this way, each class is assigned a weight, which is derived from the confidences of the rules having the class as consequents. The final classification is predicted by weighted majority voting, which is known as 'Winner-Take-All strategy' as illustrated in Fig.2.10 (Higgins, 1993).

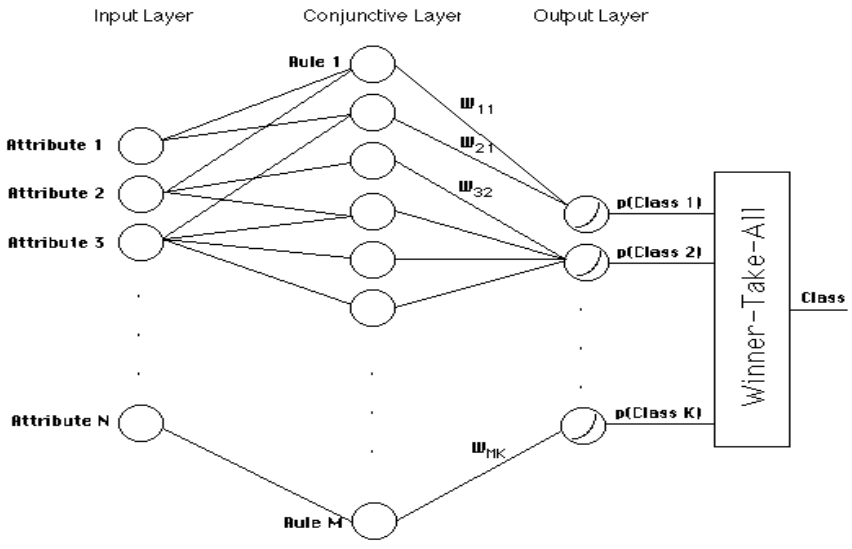

Fig.2.10 Higgins's non-deterministic rule based network for classification (Higgins, 1993)

The network topology illustrated in Fig.2.10 could be seen as a special type of rule based network representation. This is because of the possibility that there are two or more rules that fire with different classifications as rule consequences. This issue is known as conflict of classification as mentioned in Section 2.2.1 and introduced in (Bramer, 2007). Higgins's network topology actually takes into account this possible issue and deals with it by using the 'Winner-Take-All strategy'. Therefore, the network topology could be seen as a type of non-deterministic rule based network with certain inputs but uncertain outputs. However, the conflict of classification would never arise with the rule sets that are generated by adopting the divide and conquer approach. In this context, if the rule generation is based on deterministic logic, both inputs and outputs would be deterministic. As it is, the networked topology could be modified to become a deterministic rule based network which is introduced in Chapter 3.

On the other hand, a rule set may have some or all rules non-deterministic in terms of relationships between rule antecedents and consequents due to the presence of uncertainty in datasets. In this context, the rule set would be used to predict classes based on probabilistic or fuzzy logic. Therefore, a unified topology for rule based networks, which could fulfil being based on different type of logic such as deterministic, probabilistic and fuzzy logic, is created and introduced in Chapter 3.

2.3 Ensemble Rule Based Systems

As mentioned in Chapter 1, ensemble learning aims to achieve that multiple classifiers work together for making predictions in order to improve the overall predictive accuracy. In this context, multiple rule based systems that work together for predictions would act as an ensemble rule based systems. This section focuses on introduction of ensemble learning concepts and ensemble rule based methods.

2.3.1 Ensemble Learning Concepts

As introduced in (Kononenko & Kukar, 2007), ensemble learning can be done in parallel or sequentially. In the former way, there are no collaborations among different learning algorithms in training stage and only their predictions are combined together for final prediction making. In this context, the final prediction is typically made by majority voting in classification and by averaging in regression. In the latter way of ensemble learning, the first algorithm learns a model from data and then the second algorithm learns to correct the former one etc. In other words, the model built by the first algorithm is further corrected by the subsequent algorithms. Two commonly used methods are Bagging (Breiman, 1996) and Boosting (Freund & Schapire, 1996). The former is a type of parallel ensemble learning method and the latter is a type of sequential ensemble learning method. The rest of Section 2.3 focuses on the description and discussion of the two methods.

In parallel ensemble learning, a popular approach is to take sampling to a data set in order to get a number of samples such as Bagging. A classification algorithm is then used to train a classifier on each of these samples. The group of classifiers constructed will make predictions on test instances independently and final predictions on the test instances will be made based on majority voting. As mentioned in (Breiman, 1996), the term Bagging stands for bootstrap aggregating which is a method for sampling of data with replacement. In particular, the Bagging method is to take a sample with the size as same as that of the original data set and to randomly select an instance from the original data set to be put into the sample set. This means that some instances in the original set may appear more than

once in the sample set and some other instances may never appear in the sample set. According to the principle of statistics, each sample is expected to contain 63.2% of the original data instances (Breiman, 1996; Kononenko & Kukar, 2007; Tan, Steinbach, & Kumar, 2006). The Bagging method is useful especially when the base classifier is not stable with high variance (Kononenko & Kukar, 2007; Tan, Steinbach, & Kumar, 2006). This is because the method is robust and does not lead to overfitting as increasing the number of generated hypothesis (Kononenko & Kukar, 2007). Some unstable classifiers include neural networks, decision trees and some other rule based methods (Kononenko & Kukar, 2007; Tan, Steinbach, & Kumar, 2006).

As mentioned in (Kononenko & Kukar, 2007; Li & Wong, 2004), Boosting stands for Adaboost, which generates an ensemble learner in a sequential way. In other words, the generation of each single classifier depends on the experience gained from its former classifier (Li & Wong, 2004). Each single classifier is assigned a weight depending on its accuracy measured by using validation data. The stopping criteria are satisfied while the error is equal to 0 or greater than 0.5 as indicated in (Li & Wong, 2004). In testing stage, each single classifier makes an independent prediction as similar to Bagging but the final prediction is made based on weighted majority voting among these independent predictions.

In this thesis, empirical investigations on ensemble learning focus on Bagging based methods. This is because the Bagging method is highly suitable to increase the robustness of rule based classification methods as justified below. However, with regard to Boosting, the weighted majority voting is incorporated instead of equal majority voting applied to Bagging and introduced in more detail with respect to its use in Section 2.3.2.

2.3.2 Ensemble Learning Methods

Random forests is another popular method (Breiman, 2001) that is similar to Bagging but the difference is that the attribute selection at each node is random. In this sense, at each node, there is a subset of attributes chosen from the training set and the one which can provide the best split for the node is finally chosen (Li & Wong, 2004). As mentioned in Section 1, random forests has decision tree learning algorithms as the bases. In the training stage, the chosen algorithm of decision tree learning is used to generate classifiers independently on the samples of the training data. In the testing stage, the classifers make the independent predictions that are combined to make the final prediction based on equal voting. As concluded in (Kononenko & Kukar, 2007), the random forests algorithm is robust because of the reduction of the variance for decision tree learning algorithms. However, the random forests algorithm makes it difficult to interpret the combined predictions, especially when the number of decision trees generated is more than 100, and thus leads to the incomprehensability of the predictions made by the decision trees. The same problem also happens to other methods such as Bagging and Boosting.

Random Prism, an existing ensemble learning method (Stahl & Bramer, 2013; Stahl & Bramer, 2011), follows the parallel ensemble learning approach and uses Bagging for sampling as illustrated in Fig.2.11. It has been proven in (Stahl & Bramer, 2013; Stahl & Bramer, 2011) that Random Prism is a noise-tolerant method alternative to Random Forests. However, the Random Prism has two weak points in training and testing stages respectively.

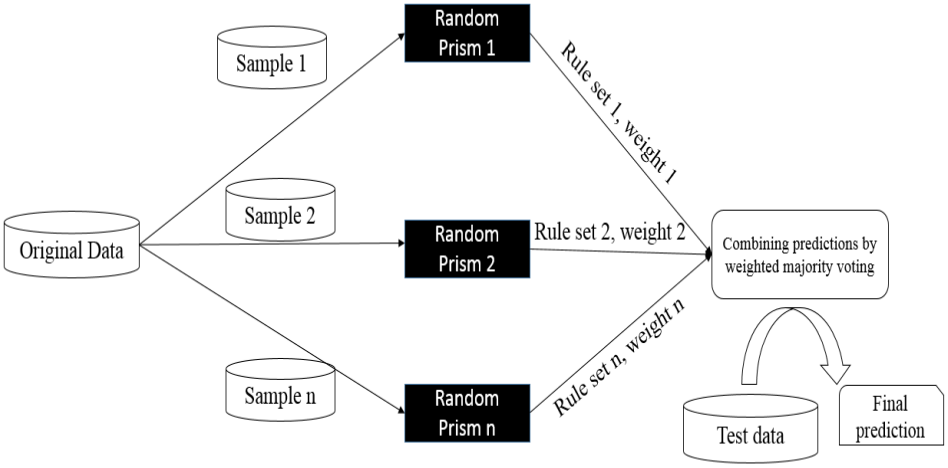

Fig.2.11 Random Prism with Bagging (Stahl & Bramer, 2013; Stahl & Bramer, 2011)

The first weak point is that there is only a single learning algorithm, PrismTCS, involved in training stage for Random Prism, which cannot always generate strong hypothesis (roust models). In fact, it is highly possible that a single algorithm performs well on some samples but poorly on the others. From this point of view, it is motivated to extend the ensemble learning framework by including multiple learning algorithms involved in training stage. This is in order to achieve that on each data sample the learner created is much stronger.

On the other hand, Random Prism uses weighted majority voting to determine the final prediction on test instances. In other words, each model is assigned a weight, which is equal to the overall accuracy checked by validation data as part of the sample. In prediction stage, each model is used to predict unseen instances and give an individual classification. The ensemble learning system then makes the final classification based on weighted majority voting instead of traditional majority voting. For example, there are three base classifiers: A, B and C. A makes the classification X with the weight 0.8 and both B and C make classification Y with the weights 0.55 and 0.2 respectively so the final classification is X if using weighted majority voting (weight for X: 0.8> 0.55+0.2=0.75) but is Y if using traditional majority voting (frequency for Y: 2>1). However, for the weighted majority voting mentioned above, the strategy in determining the weight is not reliable enough especially for unbalanced data sets. This is because it is highly possible that a classifier performs better on predicting positive instances but worse on negative instances if it is a two class classification task. The similar case can also happen in multi-class classification tasks. Therefore, it is more reasonable to use the individual accuracy for a single classification (e.g. true positive rate) as the weight.

The above two weak points are also mentioned with suggestions for further improvements in (Stahl & Bramer, 2013; Stahl & Bramer, 2011). Therefore, an advanced framework of ensemble learning is created in order to overcome the limitations and further introduced in Chapter 3.

2.4 Conclusion

This chapter reviews some existing methods and techniques in rule based classification and ensemble learning. Also, the strengths and limitations of the methods and techniques are highlighted. These thus motivate the development of more advanced methods and techniques which are further introduced in Chapter 3. In particular, a new method for rule generation is developed to overcome the limitations of Prism with respects to underfitting, clash and computational efficiency. Another method for rule simplification is developed to overcome the limitations of J-pruning and Jmax-pruning with respects to underfitting and computational efficiency. A new technique for rule representation is developed to overcome the limitations of decision tree and linear list with respects to computational complexity and knowledge representation. An advanced framework of ensemble learning is created to overcome the limitations of Random Prism with respects to use of base classifiers and confidence measure.

Chapter 3 Research Methodology

3.1 Introduction

Chapter 2 reviews two existing rule generation methods namely TDIDT and Prism. A list of limitations of the two methods are highlighted and lead to development of a new method for rule generation. Chapter 2 also reviews two existing pruning methods namely J-pruning and Jmax-pruning and highlights their limitations. Therefore, it is led to develop another pruning method. Besides, two existing techniques, decision trees and linear lists for rule representation and an existing ensemble learning method called Random Prism are also reviewed in Chapter 2. Some typical limitations are pointed out and lead to the development of novel methods and techniques. This chapter introduces a unified framework for design of rule based classification systems and some novel methods and techniques which are developed in the PhD research and can be successfully integrated into the framework.

3.2 Framework for Designing Rule Based Classification Systems

As mentioned in Chapter 1, most rule generation methods suffer from overfitting of training data. The overfitting can be successfully reduced by simplifying rules using suitable pruning methods so that loss of accuracy and efficiency is avoided. This indicates that rule simplification is a necessary operation for design of rule based classification systems. On the other hand, if a rule set consists of a large number of complex rules, efficiency in predicting unseen instances would be seriously affected. However, the efficiency in prediction stage is also subject to representation of the rule set in addition to complexity of the rule set, which means that the change to rule representation can also successfully improve the efficiency in prediction. In addition, for the same rule set, different rule representation also usually leads to different levels of interpretability for knowledge extraction. Therefore, the two points of view above indicate that rule representation is also a necessary operation for design of rule based classification systems.

On the basis of the above descriptions, a unified framework for design of rule based classification systems is created in the PhD research. The framework consists of three operations, namely rule generation, rule simplification and rule representation. Three novel methods/techniques that are developed in the PhD research and used for the three operations respectively are further introduced in the following subsections.

3.2.1 Information Entropy Based Rule Generation

Information Entropy Based Rule Generation (IEBRG) is a method of classification rules generation following the ‘separate and conquer’ approach. This method manages to avoid underfitting and redundant computational efforts.

This method is attribute-value-oriented like Prism but it uses the ‘from cause to effect’ approach. In other words, it does not have a target class pre-assigned to the rule being generated. The main difference from Prism is that IEBRG focuses mainly on minimising the uncertainty for each rule being generated no matter what the target class is. A popular technique used to measure the uncertainty is information entropy introduced in (Shanno, 1948). The basic idea of IEBRG is illustrated in Fig.3.1.

Input: a training set *T*, a subset $T' \subseteq T$, an instance $t \in T$, dimensionality *d*, an attribute a_x (*x* is the index of *a*), entropy *E*, number of classes *n*, class C_i (*I* is the index of *C*)
Output: a rule set *RS*, a result set of instances *T''* covered by a rule $R \in RS$

Initialize:
$T'= T$, $T''= \emptyset$, $E=-\sum P(C_i) \log_2 P(C_i)$, $i=0,1,2....n-1$
While $T' \neq \emptyset$ **Do**
  While $E \neq 0$ **Do**
    $x=0$;
    While $x < d$ **Do**
      For each value *v* of a_x **Do**
        Calculate $E(C| a_x= v)$;
      End For
      $x++$;
    End While
    assign $a_x= v$ to *R* as a rule term, while $E(C| a_x= v)$ is min;
    $\forall t: T'' \cap \{t\}$, if *t* comprise $a_x= v$;
  End While
  $RS= RS \cap \{R\}$;
  $T'= T'- T''$;
  update *E*;
End While

Fig.3.1 IEBRG Algorithm

As mentioned in Chapter 2, all versions of Prism need to have a target class pre-assigned to the rule being generated. In addition, an attribute might not be relevant to some particular classifications and sometimes only one value of an attribute is relevant (Deng, 2012; Cendrowska, 1987). Therefore, the Prism method chooses to pay more attention to the relationship between an attribute-value pair and a particular class. However, the class to which the attribute-value pair is highly relevant is probably unknown, as can be seen from the example in Table 3.1 below with reference to the *contact lenses* dataset illustrated in Chapter 2. This dataset shows that P (class= no lenses |tears= reduced) =1 illustrated by the frequency table for attribute *tears*. The best rule generated first would be "if tears= reduced then class= no lenses".

Table 3.1 Lens 24 dataset example

Class Label	Tears= reduced	Tears= normal
Class = hard lenses	0	6
Class = soft lenses	0	6
Class = no lenses	12	0
Total	12	12

This indicates that the attribute-value "tears= reduced" is only relevant to class *no lenses*. However, this is actually not known before the rule generation. According to PrismTCS strategy, the first rule being generated would select "class = hard lenses" as target class as it is the minority class (Frequency=6). Original Prism may select class *hard lenses* as well because it is in a smaller index. As described in (Bramer, 2007), the first rule generated by Original Prism is "if astig= yes and tears= normal and age=young then class= hard lenses". It indicates that the computational efficiency is slightly worse than expected and the resulting rule is more complex. When Big Data is used for training, the Prism method may be even likely to generate an incomplete rule covering a clash set as mentioned in Section

2.1 if the target class assigned is not a good fit to some of those attribute-value pairs in the current training set. Then the whole rule may be discarded resulting in underfitting and redundant computational effort.

In order to find a better strategy for reducing the computational cost, the IEBRG method is developed. In this technique, the first iteration of the rule generation process for the *contact lenses* dataset can make the entropy for the resulting subset reach 0. Thus the first rule generation is complete and its rule is represented as "if tears= reduced then class= no lenses".

In comparison with the Prism family, this algorithm may reduce significantly the computational cost, especially when Big Data is being dealt with. In addition, in contrast to Prism, the IEBRG method deals with clashes (introduced later) by assigning a majority class in the clash set to the current rule. This has the potential reducing the underfiting of a rule set thus reducing the number of unclassified instances although it may increase the number of misclassified instances. On the other hand, the IEBRG also has the potential to better avoid clashes occurring compared with Prism.

In practice, there are some typical problems that need to be taken into account and dealt with effectively. These include the ways of dealing with clashes and tie-breaking on conditional entropy as well as conflict of classification. The rest of the Section 3.3 focuses on description in these aspects.

With regard to clashes, there are two principal ways to deal with this kind of problem as mentioned in (Bramer, 2007) as follows:

1) Majority voting: to assign the most common classification of the instances in the clash set to the current rule.
2) Discarding: to discard the whole rule currently being generated

In this thesis, 'majority voting' is chosen as the strategy of dealing with this problem as the objective is mainly to validate this method and to find its potential in improving accuracy and computation efficiency as much as possible.

With regard to tie-breaking on conditional entropy, it is solved by deciding which attribute-value pair is to be selected to split the current subset when there are two or more attribute-value pairs that equally well match the selection condition. In the IEBRG method, this problem may occur when two or more attribute-value pairs have the equally smallest entropy value. The strategy is the same as the one applied to Prism by taking the one with the highest total frequency as introduced in (Bramer, 2007).

As mentioned in Chapter 2, the classification conflict problem may occur to modular classification rule generators such as Prism. Similarly, the IEBRG may also face this problem. In this thesis, the author chooses the 'take the first rule that fires' strategy which is already mentioned in Chapter 2 because this method usually generates the most important rules first. Consider the example below:

Rule 1: if x=1 and y=1 then class= 1;
Rule 2: if x=1 then class=2;

This seems as if there is a conflict problem but the two rules can be ordered as rule 1 is more important. In other words, the second rule can be represented in the following way:

Rule 2: if x=1 and y≠1 then class=2;

This indicates that after the first rule has been generated, all instances covered by the rule have been deleted from the training set; then the two conditions ‘x=1’ and ‘y=1’ cannot be met simultaneously any more. Thus the first rule is more important than the second one.

3.2.2 Jmid-pruning Based Rule Simplification

As mentioned in Chapter 2, neither J-pruning nor Jmax-pruning exploit the J-measure to its full potential and they may lead to underfitting. In addition, Jmax-pruning is computationally more expensive. Therefore, Jmid-pruning is developed, which avoids underfitting and unnecessary rule term inductions while at the same time rules are being pruned for reducing overfitting.

The Jmid-pruning is a modified version of the J-measure based pruning algorithm Jmax-pruning. It not only monitors and records the highest J-value observed so far but also measures the potentially highest J-value that may be achieved eventually by making use of the Jmax value as highlighted in Chapter 2 in comparison to Jmax-pruning. The basic concept of this algorithm is illustrated in Algorithm 3.

```
Rule r = new Rule;

Boolean rule_Incomplete = true;

term_index = 0;

Do While (rule_Incomplete){
    Term t = generate new term;
    term_index++;
    append t to r;
    compute J_value of r;
    IF(J_value > best_J_Value){
       best_J_Value = J_Value;
      best_term_index = term_index;
      record current_marjority_class;
    }
      compute Jmax_value of r;
     IF(best_J_value> Jmax_value){
       do not append t to r;
       cut r back to rule best_term_index;
       invoke clash handling for r;
       rule_Incomplete = false;

      }

      ELSE{
        append t to r;
      }

    IF(No more rule terms can be induced){
       cut r back to rule best_term_index;
       invoke clash handling for r;
```

```
        rule_Incomplete = false;
    }
}
```

Algorithm 3 Jmid-pruning for Prism algorithms

The Jmid-pruning aims to avoid underfitting and unnecessary computational effort especially when Big Data is used for training. In fact, J-pruning and Jmax-pruning do not actually make use of Jmax value to measure the potential search space of gaining benefits.

Let us consider an example from (Bramer, 2002) using the contact lenses dataset. There is a rule generated as follows according to (Bramer, 2002):

If tears= normal and astig=no and age= presbyopic and specRx = myope then class= no lenses;

After adding the four terms subsequently, the corresponding J and Jmax values change in the trend as follows according to (Bramer, 2002):

If tears= normal then class= no lenses; (J=0.210, Jmax=0.531)
If tears= normal and astig= no then class= no lenses; (J=0.161, Jmax=0.295)
If tears= normal and astig= no and age= presbyopic then class= no lenses; (J=0.004, Jmax=0.059)
If tears= normal and astig=no and age= presbyopic and specRx = myope then class= no lenses; (J=0.028, Jmax=0.028)

In this example, all of the three algorithms would provide the same simplified rule that is: if tears=normal then class= no lenses; this is because the highest J-value has been given after adding the first term (tears= normal). However, the computational efficiency would be different in the three methods. J-pruning would decide to stop the generation after the second term (astig=no) is added as the J-value goes down after the second term (astig=no) is added. In contrast, Jmax-pruning would stop when the rule is complete. In other words, the generation would be stopped after the fourth (last) term is added and then the terms (astig=no, age= presbyopic and specRx= myope) will be removed. In addition, Jmid-pruning would decide to stop the generation after the third term is added as the value of Jmax (0.295) is still higher than the J-value (0.210) given after the first term (tears=normal) is added although its corresponding J-value (0.161) decreases; however, the generation should be stopped after the third term (age= presbyopic) is added as both J (0.004) and Jmax (0.059) values are lower than the J-value (0.161) computed after the second term (astig=no) is added although the J-value could still increase up to 0.059.

On the basis of the description above, J-pruning would be the most efficient and Jmid-pruning is more efficient than Jmax-pruning. However, it seems that J-pruning may prune rules too early when the training data is in large scalability as mentioned in Chapter 2. For example, one of the rules is generated from the Soybean dataset (Lichman, 2013) and shows the change trend of J-value and Jmax as follows according to (Stahl & Bramer, 2011; Stahl & Bramer, 2012):

If temp= norm and same-lst-sev-yrs= whole-field and crop-hist= same-lst-two-yrs then class=frog-eye-leaf-spot;

First term:
If temp= norm then class=frog-eye-leaf-spot; (J= 0.00113, Jmax=0.02315)
Second term:
If temp= norm and same-lst-sev-yrs= whole-field then class=frog-eye-leaf-spot; (J=0.00032, Jmax=0.01157)
Third term:
If temp= norm and same-lst-sev-yrs= whole-field and crop-hist= same-lst-two-yrs then class=frog-eye-leaf-spot; (J=0.00578, Jmax=0.00578)
In this case, both Jmax-pruning and Jmid-pruning would normally stop the generation when the rule is complete and take the complete rule: If temp= norm and same-lst-sev-yrs= whole-field and crop-hist= same-lst-two-yrs then class=frog-eye-leaf-spot; as the final rule with the highest J-value (0.00578). In contrast, J-pruning would stop the generation after the second term (same-lst-sev-yrs= whole-field) is added and take the rule: If temp= norm then class=frog-eye-leaf-spot; as the final rule with a lower J-value (0.00113 instead of 0.00578).

The other potential advantage of Jmid-pruning in comparison with Jmax-pruning is that Jmid-pruning may successfully get more rules not being discarded eventually, when tie-breaking on J-value happens as mentioned in Chapter 2. From this point of view, Jmid-pruning is better in avoiding underfitting of rule sets.

3.2.3 Network Based Rule Representation

As mentioned in Chapter 2, both tree and list representations have their individual limitations. A networked representation of classification rules is created, which is called rule based networks and provides a higher level of computational efficiency than tree and list representations for the same rule set in prediction stage.

The rule based network representation is illustrated in Fig.3.2 below.

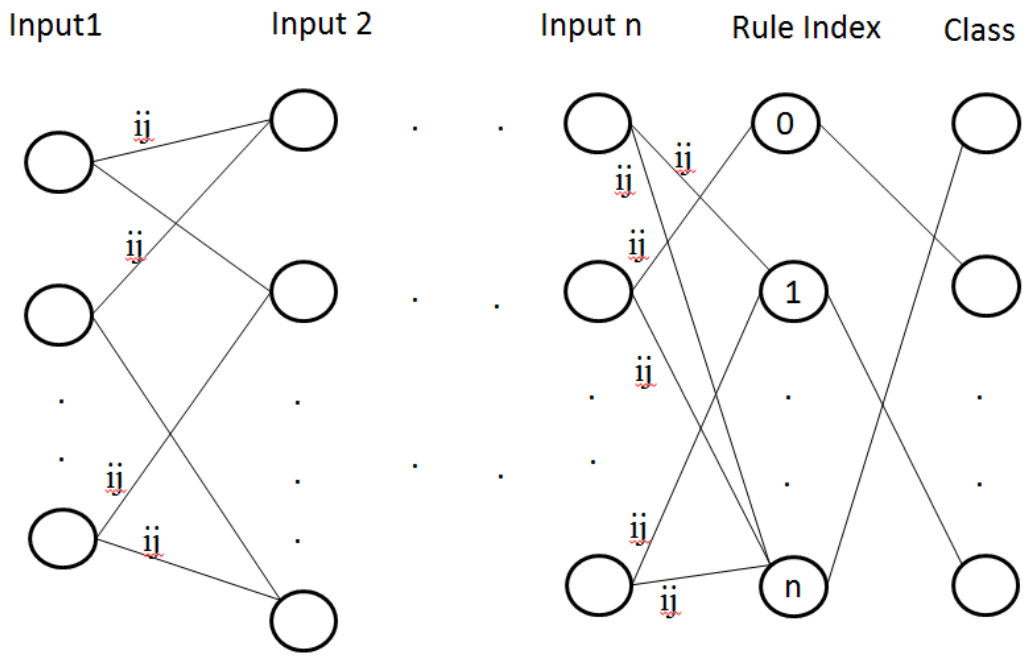

Fig.3.2 Rule Based Network version 1

In general, this is an n+2 layer network. The first n layers represent n input attributes. In each of the layers, each node represents a value of the corresponding attribute. The last second layer represents the rule index which is equal to the order of this rule minus one. For example, if the rule index is 0, it indicates that this is the first rule in the rule set. The last layer in the network represents the class output. There are also connections between different layers, which are to be explained further using specific examples. However, in general, the connections could be between two layers which are not adjacent each other. For example, the nodes in the first layer could have connections with other nodes in the third layer. This is very like a travel route which includes a number of cities. In this context, each city is like a rule term and each route is like a rule. It is possible that there are cities which are not adjacent each other but included in the same travel route. In addition, any two nodes may have not only one connection. This is because the same part of conjunction of rule terms may be in two or more rules as illustrated by the rules below:

If a=0 and b=0 and c=0 then class=0;
If a=0 and b=0 then class=0;

In the context of travel route as mentioned above, this is like that there could be common cities included in different routes.

On the other hand, rule representation is also significant to fulfil the requirement of the interpretability of a rule set as mentioned in Chapter 2. From this point of view, another version of rule based network is developed to fulfil the requirement of knowledge discovery as illustrated in Fig.3.3. This version is actually modified from Higgins's network topology (Higgins, 1993) as illustrated in Section 2.2.3.

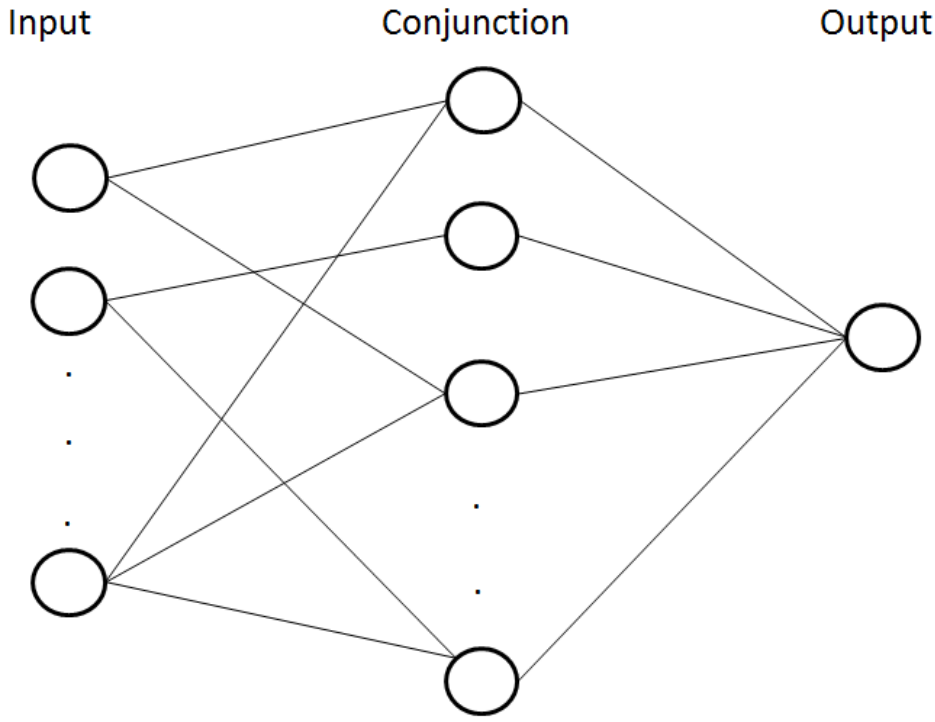

Fig.3.3 Rule Based Network version 2

In general, this is a three layer network. In the first layer, each node represents an input attribute and this layer is referred to as input layer. In the middle layer, each node represents a rule to make the conjunction among inputs and provide outputs for the node in the last layer and thus the middle layer is referred to as conjunction layer. The only node in the last layer represents the class output and thus this layer is referred to as output layer. In addition, the nodes in the input layer usually have connections to other nodes in the conjunction layer. Each of the connections represents a condition judgement which is explained further using specific examples. However, a node in the input layer may not necessarily have connections to other nodes in the conjunction layer. This is due to a special case that an attribute may be totally irrelevant to making a classification. In other words, this attribute is not involved in any rules in the form of rule terms. From this point of view, this version of rule based network representation can help identify the relevance of attributes for feature selection tasks, which is discussed further in this section.

In order to illustrate the two versions of rule based network introduced above in detail, let us see a set of rules based on Boolean logic below:

If x1=0 and x2=0 then class=0;
If x1=0 and x2=1 then class=0;
If x1=1 and x2=0 then class=0;
If x1=1 and x2=1 then class=1;

The corresponding networked representation is illustrated in Fig.3.4. In this representation, x1=1 and x2=1 are supposed to be the two inputs respectively for prediction. Thus both ‘x1’ and ‘x2’ layers get green node labelled 1 and red node labelled 0 because each node in the layer x1 represents a value of attribute x1 and so does each node in the layer x2. In addition, the two digits labelled to each of the connections between the nodes in the layers x1 and x2 represent the index of rule and rule term respectively. In other words, the two digits ‘11’ as illustrated in Fig.3.4 indicates that it is for the first rule and the first term of the rule. It can be seen from the list of rules above that the first term of the first rule is ‘x1=0’. However, the input value of x1 is 1 so the connection is coloured red as this condition is not satisfied. In contrast, the connections labelled ‘31’ and ‘41’ respectively are both coloured green as the condition ‘x1=1’ is satisfied. The same principle is also applied to the connections between the nodes in the layers ‘x2’ and ‘Rule Index’. As the two inputs are ‘x1=1 and ‘x2=1’, the connections ‘31’, ‘41’ and ‘42’ are coloured green and the node labelled 3 is green in the layer ‘Rule Index’ as well as the output is 1 in the layer ‘Class’.

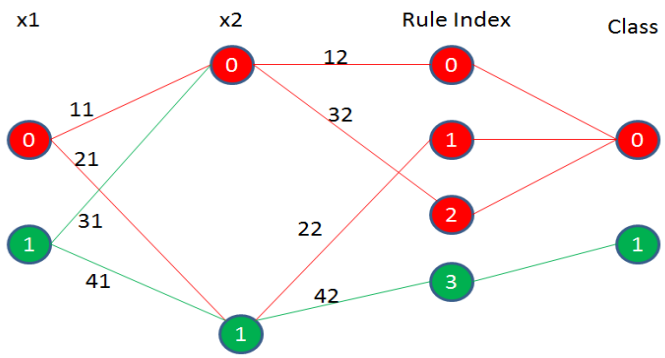

Fig. 3.4 Rule Based Network example (version 1)

For Rule Based Networks, the prediction process is run by going through rule terms in divide and conquer search (i.e. only going through those terms that fire). The total number of terms is used as the input size of data (n) as same as used in linear list representation and thus the efficiency is O (log (n)). As can be seen from Fig.10, it only takes three steps (going through connections '31', '41' and '42') to find the first rule that fires (the rule index is 3). This is because the input value of x1 is 1 and thus the connections '11' and '21' can be ignored. In the second layer, it is only concerned with the connection '42' as the input value of x2 is 1 and thus the connections '12' and '32' can be ignored. In addition, the connection '22' is ignored as well because the connection '21' is already discarded and thus it is not worth to go through the connection '22' any more. The above descriptions indicate that it is not necessary to examine the whole network in order to find the rules that fire and thus the efficiency of the rule based network is higher than that of the linear list, the latter of which is O (n) as mentioned in Chapter 2. In practice, it would significantly speed up the process of predictions when the corresponding rule set is generated by learning from Big Data.

The networked representation mentioned above is mainly used for machine learning purpose with respect to the improvement of efficiency in prediction making. However, in data mining tasks, the interpretability of a rule set is significant towards representation of knowledge as mentioned in (Stahl & Jordanov, 2012). From this point of view, another type of networked representation is developed for this purpose with respect to the improvement of knowledge representation. This type of networked representation is based on the relationship between attribute and class as illustrated in Fig.3.5 and both input values are supposed to be 1 (shown as green).

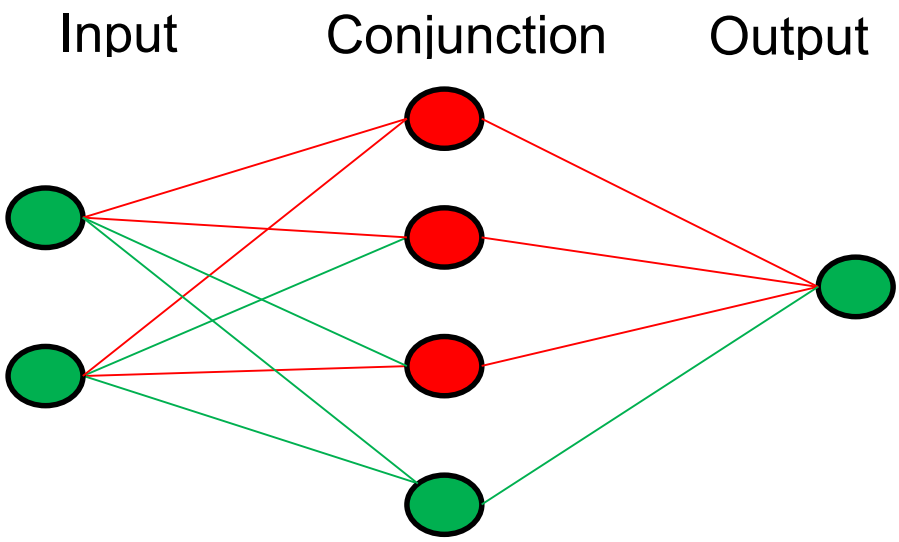

Fig.3.5 Rule Based Network example (version 2)

In this diagram, as mentioned earlier, each node in the input layer represents an input attribute; each node in the middle layer represents a rule and the layer is referred to as conjunction layer due to the fact that each rule actually reflects the mapping between inputs and outputs and that the output values strongly depend on the conjunction of input values; finally, the node in the output layer represents the class attribute. On the other hand, each of the connections between the input layer and the conjunction layer represents a condition judgement. If the condition is met, then the connection is coloured by green. Otherwise, it is coloured by red. In addition, each of the connections between the conjunction layer and the

output layer represents an output value from the corresponding rule. In other words, if all of the conditions in a rule are met, then the corresponding node in the conjunction layer becomes green. Otherwise, the corresponding node becomes red. The former case would result in that a node representing a rule becomes green and that the output value from the rule is assigned to the class attribute in the output layer. In the meantime, the connection between the node representing the rule and another node representing the class attribute becomes green, which means that the class attribute would be assigned the output value from the rule. In contrast, the latter case would result in that the node in the conjunction layer becomes red and that the output value from the corresponding rule cannot be assigned to the class attribute.

This type of networked rule representation shows the relationships between attributes and rules explicitly as shown connections between nodes in the input layer and nodes in the conjunction layer. In addition, the networked representation also introduces a ranking for both input attributes and rules according to their importance. The importance of an input attribute is measured by the weighted average of ranks for those rules that relate to the input attribute. For example, the attribute A relates to two rules namely rule 1 and rule 2. If the ranks for rule 1 and rule 2 are 4 and 8 respectively, then the average of ranks would be 6((4+8)/2). In real applications, this characteristic about ranking of attributes may significantly contribute to both knowledge discovery and feature selection with respect to feature importance. Besides, strength of the representation also lies in the strong interpretability on mapping relationships between inputs and outputs, which is significantly useful for knowledge discovery. On the basis of above descriptions, the rule based network illustrated in Fig.3.5 is thus a practically significant technique in data mining tasks.

However, as mentioned in Section 2.2.3, a rule set may be used to predict classes based on probabilistic or fuzzy logic due to the presence of uncertainty in some or all rules in the rule set. This motivates the generalization of the topology for rule based networks. This is in order to make the representation of rule based networks fit the computation based on different type of logics such as deterministic, probabilistic and fuzzy logic. The unified topology of rule based networks is illustrated in Fig 3.6 below.

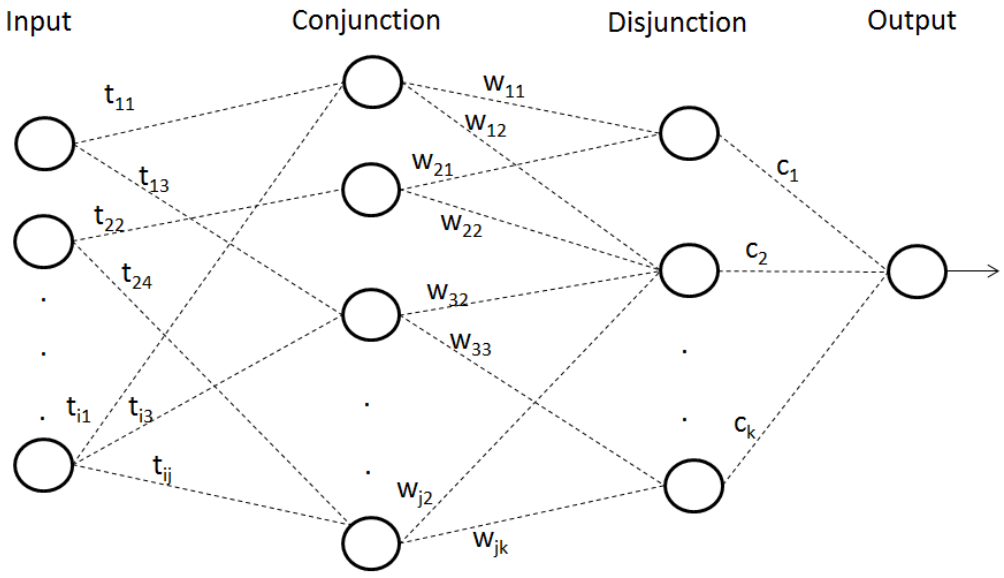

Fig.3.6 Unified rule based network

In this network topology, the modifications are made to the one illustrated in Fig.3.5 by adding a new layer called disjunction and assigning a weight to each of the connections between nodes. The disjunction layer is similar to the output layer in Higgins's network topology. In this layer, each node represents a class label. However, the final prediction is not necessarily made by choosing the most common class which has the highest posteriori probability in total. In addition, the topology also allows representing inconsistent rules, which means that the same rule antecedent could be mapped to different classes (consequents). For example, the first node in the conjunction layer is mapped to both the first and the second node in the disjunction layer as illustrated in Fig.3.6. With regard to the weights assigned to the connections between nodes, they would represent the truth values if the computation is based on deterministic or fuzzy logic. The truth value would be crisp (0 or 1) for deterministic logic whereas it would be continuous (between 0 and 1) for fuzzy logic. If the computation is based on probabilistic logic, the weights would represent the probabilities of the corresponding cases.

In the context of deterministic logic, each of the connections between the nodes in the input layer and the nodes in the conjunction layer would be labelled 1 for its weight, i.e. $t_{ij}= 1$, where i is index of attribute and j is the index of the rule, if the corresponding condition as part of the rule antecedent is met. A rule would have its antecedent satisfied if and only if all of the conditions are met. In this case, the rule is firing to indicate its consequent (as the class predicted) which is represented by a node in the disjunction layer. If the rule is consistent, the corresponding node should have a single connection to another node in the disjunction layer. The connection would be labelled 1 as its weight denoted by w_{jk}, where k is the index of the class. In this case, if there is only one rule firing or more rules firing without conflict of classification, then the output would be deterministic. This is because there is only one node in the disjunction layer providing a weight greater than or equal to 1 for its connection to the node in the output layer. For all other nodes, the weight provided for the corresponding connection would be equal to 0. However, as mentioned earlier, a rule may be inconsistent, which means that the same rule antecedent may be mapped to different classes as its consequent. In this case, the corresponding node would have multiple connections to different nodes in the disjunction layer. For each of the connections, the weight would be equal to a value between 0 and 1. Nevertheless, the sum of the weights for the connections would be equal to 1. With regard to each of the classes, it may be mapped from different rule antecedents. Therefore, each class would have a summative weight denoted by c_k, which is equal to the sum of the weights for the rule antecedents mapped to the class. Finally, the node in the output layer makes the weighted majority voting for the final prediction.

In the context of probabilistic logic, the t_{ij} would be equal to a value between 0 and 1 as a conditional probability. Similar to deterministic logic, a rule is firing if and only if all of the conditions are met. However, the rule antecedent would be assigned a firing probability computed in the corresponding node in the conjunction layer. The firing probability is simply equal to the product of the conditional probabilities for the rule terms (if corresponding attributes are independent) and also to the posterior probability of the rule consequent given the rule antecedent. If the rule is inconsistent, the sum of posterior probabilities for the possible classes (w_{jk}) would also be equal to the firing probability above. This is because the rule consequent is the disjunction of the output terms, each of which has a different class as the output value. In disjunction layer, each class is assigned a weight

which is equal to the sum of its posterior probabilities given different rule antecedents. The final prediction is made by weighted majority voting as same as based on deterministic logic.

In the context of fuzzy logic, in contrast to probabilistic logic, in the conjunction layer, the t_{ij} would be equal to a value between 0 and 1 as a fuzzy truth value for each corresponding condition. Similar to another two types of logic, a rule is firing if and only if all of the conditions are met. However, the rule antecedent would be assigned a firing strength computed in the corresponding node in the conjunction layer. The firing strength is simply computed by choosing the minimum among the fuzzy truth values of the conditions (that are assumed independent). The fuzzy truth value for the rule consequent is equal to the firing strength. If the rule is inconsistent, the fuzzy truth value (w_{jk}) for having each possible class as the consequent would be derived by getting the minimum between the firing strength and the original fuzzy truth value assigned to this class for this rule. In the disjunction layer, the weight for each class is computed by getting the maximum among the fuzzy truth values (w_{jk}) of the rules having the class as the consequents. The final prediction is made by weighted majority voting as same as based on above two types of logic.

3.3 Collaborative and Competitive Random Decision Rules

As mentioned in Chapter 2, Random Prism is a noise tolerant ensemble learning algorithm alternative to Random Forests (Breiman, 2001). However, it has two weak points in training and testing stages respectively. This section introduces an advanced ensemble learning framework extended from Random Prism with the aim to overcome the two weak points which are mentioned above and described in Chapter 2.

The advanced ensemble learning framework introduced in the PhD thesis is referred to as Collaborative and Competitive Random Decision Rules (CCRDR) and illustrated in Fig.3.7, which indicates that the ensemble learning framework includes both collaborative learning and competitive learning involved.

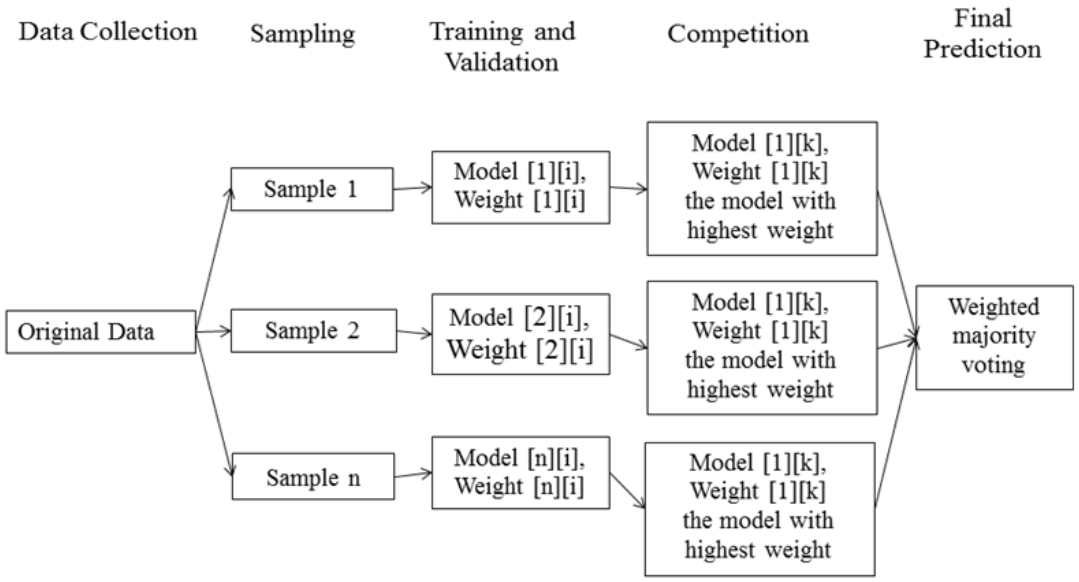

Fig.3.7 Procedures of Proposed Ensemble Learning

The first weak point of Random Prism is that there is only a single learning algorithm involved in training stage, which cannot always generate robust models as mentioned in

Chapter 2. In order to overcome the limitation, the ensemble learning framework is modified in the way that the framework can include multiple learning algorithms for training. Due to this modification, there is thus competition involved among the classifiers built on the same sample of training data. In other words, there are multiple learning algorithms applied to each sample of training data, which implies that multiple classifiers are built on each sample. In this context, it becomes achievable to find better classifiers to be involved in testing stage and worse classifiers to be absent through competition among the classifiers. The competition is based upon the weight (confidence) of each of the classifiers by means of overall accuracy measured by validation data. In the extended framework, only the classifier with the highest weight (confidence) is eligible to be involved in testing stage. The modification with regard to the first weak point is also reflected from the second part of the name of the method namely 'Competitive Random Decision Rules'. The name of decision rules indicates that any rule based classification methods are eligible for being involved in training stage as base algorithms. This modification theoretically contributes to that on each sample of data the learners constructed become much stronger.

The second weak point is about the way of determining the weight of a classifier for weighted majority voting as mentioned in Chapter 2. In order to overcome the limitation, the use of confusion matrix is recommended in (Stahl & Bramer, 2013; Stahl & Bramer, 2011) and also identified by the author of the thesis. However, the individual accuracy for a single classification reflected from confusion matrix is not effective in some special cases. In contrast, precision for a particular classification would be more reliable in determining the weight of a classifier. For example, there are 5 positive instances out of 20 in a test set and a classifier correctly predicts the 5 instances as positive but incorrectly predicts other 5 instances as positive as well. In this case, the recall/true positive rate is 100% as all of the five positive instances are correctly classified. However, the precision on positive class is only 50%. This is because the classifier predicts 10 instances as positive and only five of them are correct. This case indicates the possibility that high recall could result from coincidence due to low frequency of a particular classification. Therefore, precision is sometimes more reliable in determining the weight of a classifier on a particular prediction from this point of view. Overall, both precision and recall would usually be more reliable than overall accuracy in determining weight of a classifier especially for unbalanced data sets but it is important to determine which one of the two metrics to be used in resolving special issues.

The modifications to Random Prism with regard to its two weak points generally aim to improve the robustness of models built in training stage and to more accurately measure the confidence of each single model in making each of particular predictions.

3.4 Collaborative Rule Generation Framework

The ensemble learning concepts introduced earlier focus on parallel learning, which means that the building of each model is totally parallel to the others without collaborations in training stage and only their predictions in testing stage are combined for final decision making. However, the ensemble learning could also be done with collaborations in training stage. In this way, it could potentially help to improve the quality of each model generated in training stage. Therefore, the author of this thesis creates another framework of ensemble learning to have collaborations among different rule based methods involved in training stage. The collaboration strategy is illustrated by Fig.3.8 in the following:

The essence of this algorithm above is based on the procedure of 'separate and conquer' rule generation. In particular, there would be a rule generated in each of the iterations. The algorithm introduced above has all chosen rule based methods involved in the iteration to generate a single rule; each of the rule based methods may also be assisted by some pruning methods depending on the setup of experiments; and then all of the rules are compared with respect to their confidences; finally only the rule with the highest confidence is selected and added into the rule set. This process is repeated until all of instances have been deleted from the training set as specified in the 'separate and conquer' approach. This way of rule generation would make it achieved that in each of the iterations the rule generated is of as higher quality as possible. This is because of the possibility that some of rules are of higher quality but the others are of lower quality if there is only one rule based method involved in training stage. The main difference to the CCRDR introduced in Section 3.3 is with respect to competition between rule based methods. CCRDR involves a competition between rule based methods per rule set. In other words, the competition is made after each of chosen methods has generated a rule set in order to compare the quality of a whole rule set. In contrast, the newly proposed method involves such a competition per rule generated. In other words, the competition is made once each of the methods has generated a rule in order to compare the quality of a single rule.

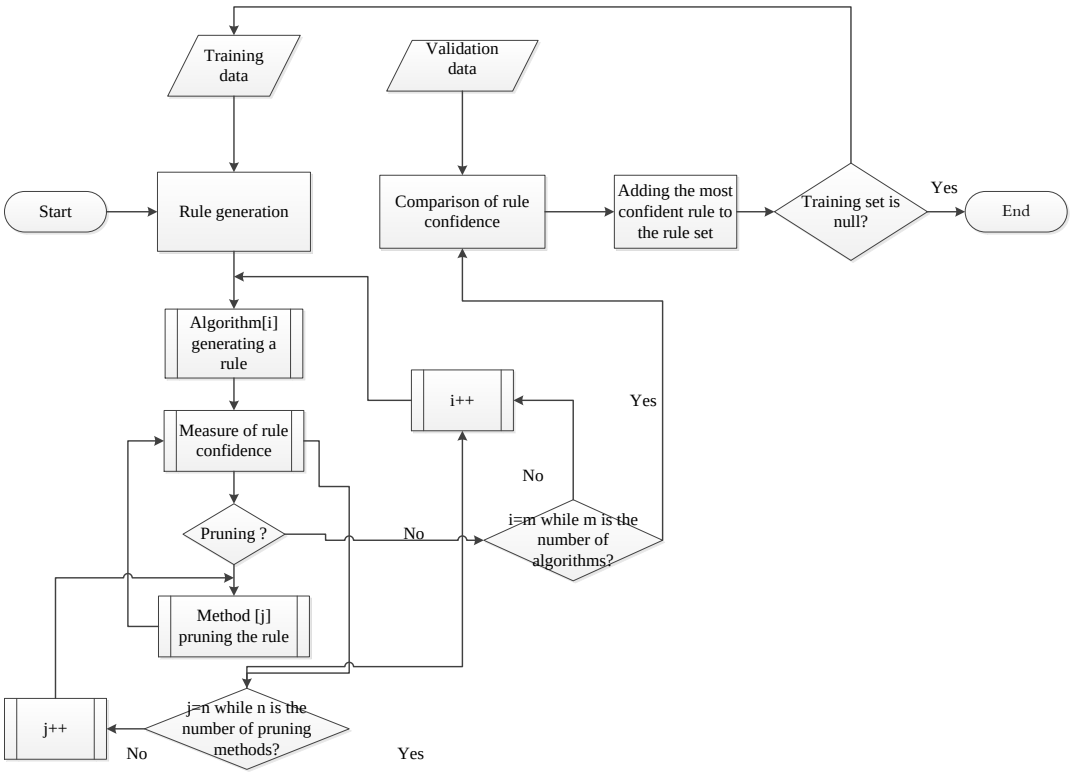

Fig.3.8 Collaborative rule generation framework

As mentioned in Chapter 2, Bagging, Boosting and Random Forests all have the disadvantage of incomprehensibility of the predictions made by different models. The same disadvantage also arises with the CCRDR approach introduced in Section 3.3. This is a serious drawback that arises with most existing ensemble learning approaches for data mining tasks. As mentioned in Chapter 1, data mining is aimed at knowledge discovery. Therefore, it is necessary for the models to allow explicit interpretation of the way in which

the prediction is made. The CRG approach would be able to fill the gap to some extent as it only generates a single rule set that is used for prediction. In addition, rule based models are highly interpretable as mentioned in Section 1; consequently, the CRG approach would fit well the purpose of knowledge discovery especially on interpretability.

With regard to accuracy, Bagging, Boosting and Random Forests all aim to improve it through scaling down data. However, there is nothing done by scaling up algorithms for improving accuracy. As mentioned in Chapter 1, it is necessary to deal with the issues on both algorithms and data sides in order to comprehensively improve the accuracy. The CCRDR can fulfil the need to a large extent. As justified in Section 3.3, each algorithm may have a different level of suitability to different data sets. On the same data set, different algorithms may also demonstrate different levels of performance. From this point of view, the CCRDR approach is designed in the way that after the training data is scaled down by drawing different samples, a group of learning algorithms are combined to generate a model on each of the samples. In this context, the CCRDR approach does not only scale down the data but also scale up the algorithms. However, as mentioned in Section 3.3, this approach does not involve any collaborations among different algorithms in the training stage. For rule based learning algorithms, it is very likely to generate a rule set that has some rules of high quality but also others of low quality. In other words, it is difficult to guarantee that each single rule generated by a particular algorithm is of high quality. In this sense, the CCRDR approach is only able to select the rule sets, each of which is generated on a particular sample set and has the highest quality on average compared with the others generated on the same sample set. In the testing stage, a rule set usually makes a prediction using a single rule that fires. If the single rule is of low quality, it is very likely to make incorrect predictions although most of the other rules are of high quality. On the other hand, for data mining tasks, each of the rules is used to provide knowledge insight for domain experts. Therefore, the reliability of each single rule is particularly significant. On the basis of the above description, the CRG approach would be useful and effective to help the CCRDR fill the gap relating to the quality of each single rule and thus also complements the other three popular methods mentioned in this section.

3.5 Conclusion

This chapter introduces novel methods and techniques in relation to rule based classification and ensemble learning. These methods and techniques are discussed by comparing them against existing methods and techniques reviewed in Chapter 2 in theoretical contexts. This chapter also highlights typical advantages of the novel methods and techniques from theoretical points of view. All of these descriptions in theoretical contexts are validated empirically in experimental studies in Chapter 4 except for rule representation which is validated in theoretical analysis.

Chapter 4 Quantitative Validation

4.1 Introduction

As mentioned in Chapter 3, the novel methods and techniques, namely IEBRG, Jmid-pruning and rule based network, are developed in order to overcome the limitations of the existing methods and techniques reviewed in Chapter 2, namely Prism, J-pruning, Jmax-pruning, decision tree representation and linear list representation. In addition, two novel ensemble learning frameworks, namely Collaborative and Competitive Random Decision Rules (CCRDR) and Collaborative Rule Generation (CRG), are created in order to overcome the limitations of existing approaches such as Random Forests and Random Prisms. The novel approaches are also compared against the existing ones in theoretical contexts in Chapter 3. Therefore, this chapter validates the advantages of the novel methods and techniques empirically in experimental studies, except for rule based network which is validated theoretically. The results are also presented and discussed in Chapter 4.

4.2 Data Sets

In order to empirically validate the methods, a number of data sets are chosen for conducting experiments. All of the data sets are retrieved from UCI repository (Lichman, 2013), Statlib repository (Vlachos, 2005) and Kent Ridge Bio-medical repository (Li & Liu, 2003).

With regard to the validation of IEBRG, the chosen data sets include: *vote, weather, contact-lense, breast-cancer, lung-cancer, nurse, tic-tac-toe, anneal, balance-scale, credit-g, credit-a, diabetes, heart-statlog, ionosphere, iris, kr-vs-kp, lymph, segment, zoo, wine, car, breast-w, mushroom, page-blocks, Monks-problems-3, dorothea, ALL-AML, colonTumor, DLBCLOutcome, DLBCLTumor, DLBCL-Stanford, LungCancer-Harvard2, lung-Michigan, lungcancer-ontario, MLL_Leukemia, NervousSystem, prostate_tumorVSNormal, BCR-ABL, E2A-PBX1, Hyperdip50, MLL, T-ALL, TEL-AML1, pos_neg_100*.

With regard to the validation of Jmid-pruning, the chosen data sets include: *vote, weather, contact-lenses, breast-cancer, car, lung-cancer, iris, segment, ionosphere, cmc, kr-vs-kp, ecoli, anneal.ORIG, audiology, optdigits, glass, lymph, yeast, shuttle, analcatdata_asbestos, analcatdata_happiness, and breast-cancer*.

With regard to the validation of CCRDR, the chosen data sets include: *anneal, balance-scale, diabetes, heart-statlog, ionosphere, lymph, car, breast-cancer, tic-tac-toe, breast-w, hepatitis, heart-c, lung-cancer, vote, page-blocks*.

With regard to the validation of CRG, the chosen data sets include: *anneal, credit-g, diabetes, heart-stalog, ionosphere, iris, kr-vs-kp, lymph, segment, zoo, wine, breast-cancer, car, breast-w, credit-a, heart-c, heart-h, hepatitis, mushroom, vote*.

For most of the chosen data sets as mentioned above, the dimensionality lies in the range of 5-100 and the number of instances in the range of 100-10000. In addition, all of the data sets are particularly for classification, which means each data set typically fulfils the following characteristics:

- Multiple input attributes and a single output attribute.
- Input attributes could be discrete or continuous.
- Output attribute must be discrete.

For some of the data sets, there are also missing values present in input attributes or class attributes. The strategies in dealing with missing values are further specified in Section 4.3. Besides, the more detailed description with respect to the characteristic of the chosen data sets are available in (Lichman, 2013; Vlachos, 2005 ; Li & Liu, 2003) and the appendix VII of the thesis.

4.3 Experimental Setup

The experimental studies are undertaken on the data sets mentioned in Section 4.2 for the validation of IEBRG, Jmid-pruning, collaborative and competitive random decision rules and collaborative rule generation.

As mentioned in Section 4.2, some of the chosen data sets contain missing values in input attributes or class attributes. This is usually a far larger issue that needs to be dealt with effectively as it would result in infinite loops for rule based methods in training stage. There are typically two ways of dealing with missing values as follows (Bramer, 2007):

1) Replace all missing values with the most frequent occurring values for discrete attributes or average values for continuous attributes.
2) Discard all instances with missing values.

In this experimental study, the first way is adopted because all of the chosen data sets are relatively small. It indicates that if the second way is adopted both training and test sets would be too small to be representative samples. Under this kind of situation, the model generated is likely to introduce biased patterns with low confidence especially if the model overfits the training data. However, the first way of dealing with missing values also potentially introduces noises to the data sets. Thus such an experimental setup would also provide the validation with respect to the noise tolerance of an algorithm in the meantime. On the other hand, if missing values are with class attribute, the best approach would be by adopting the second way mentioned above. This is because the first way mentioned above is likely to introduce noises to the data sets and thus incorrect patterns and predictive accuracies would be introduced. This is also mentioned in (Bramer, 2007) that the first way is unlikely to prove successful in most cases and thus the second way would be the best approach. In practice, the two ways of dealing with missing values can easily be achieved by using the implementations in some popular machine learning software such as Weka (Hall, et al., 2009), some of which can be located at the Journal of Machine Learning Research Repository (Machine Learning Open Source Software, 2000).

This experimental study is divided into three parts namely, unit testing, integrated testing and system testing, following software testing approaches. The following subsections describe the setup of each part of the experimental studies in more detail.

4.3.1 Unit Testing

This part of experimental study includes validation of IEBRG and Jmid-pruning. For the validation of both methods, the accuracy performed by random guess which depends on the number of classifications and distribution of these classifications is estimated for comparison with the other chosen methods. For example, if the objective function is a two class classification problem and the distribution is 50:50, then the accuracy performed by random guess would be 50%. Otherwise, the accuracy must be higher than 50% in all other cases.

With regard to IEBRG, the validation is against Prism in terms of classification accuracy and computational efficiency. With respect to accuracy, the chosen method is cross-validation (Bramer, 2007). The reasons are that the chosen data sets are all relatively small in terms of the number of instances as mentioned in the Section 4.2 and for most data sets there are no supplied test sets provided. In such cases, a single split of a data set into a training set and a test set is usually not able to provide convincing results. In other words, the results are likely to introduce bias with respect to the performance of a particular algorithm. This is because according to the standard rate in the split of a data set there are 70% of the instances used as the training set and the rest of them as the test set. This indicates that the test set would be quite small after the split of a small data set into a training set and a test set. The results obtained under this kind of situation is likely to be either extremely good or extremely poor, especially when the test instances are all highly similar to each other. This is a kind of bias that arises in testing stage. On the other hand, there is also another kind of bias that arises in training stage. This is because a model is likely to cover biased patterns discovered from training data if the training set is very small although there is a supplied large test set. On the basis of above considerations, cross validation is thus chosen for measuring classification accuracy for the chosen data sets. In addition, for each data set, the accuracy achieved by random guess is also calculated precisely to compare with that performed by IEBRG and Prism. This is because the accuracy must be at least higher than that achieved by random guess if the algorithm really works effectively.

With respect to efficiency, it is mainly measured by using runtime. In addition, there are also some other measures, namely number of rules, average number of rule terms, overall numbers of generated rule terms and discarded rule terms, used to show the approximate correction between actual runtime and computational complexity reflected from the measures mentioned above. Runtime is a popular measure particularly used for empirical validation. However, runtime is actually subject to many practical factors. For example, it is platform dependent, which means that the difference in hardware performance and operating systems would lead to different levels of runtime. In addition, the runtime is also subject to programming languages. In general, those complier based languages such as C are faster than those interpreter based languages such as Java. Even if the chosen programming language is the same, the runtime is still affected by the quality of implementations. For academic purpose, it is worth to use theoretical measures such as number of rules or rule terms to measure approximately the computational efficiency in a unified way. The overall number of rule terms actually indicates the approximate number of iterations with respect to computational complexity. As mentioned in Chapter 2, Prism prefers to discard a rule if a clash occurs. Therefore, the experimental study also takes into account the clash rate. The clash rate reflects the effectiveness of computation. This is because the fact that the algorithm takes time to generate a rule which is eventually discarded is equivalent to doing nothing and thus useless computation. On the basis of above considerations, runtime is thus chosen to validate with respect to efficiency in empirical context as well as those measures in relation to computational complexity in theoretical context.

Besides, the experimental studies for the comparison between IEBRG and Prism are also undertaken on noise data sets. The noise is artificially introduced to both training and test sets in different levels of percentages. This is in order to investigate empirically the robustness of an algorithm to noise. In particular, the investigation is to observe the change

trend in terms of accuracy and the numbers of rules and rule terms as the increase of the percentage of noise in both training and test sets.

With regard to Jmid-pruning, the validation is against J-pruning and Jmax-pruning in terms of classification accuracy and computational efficiency. With respect to accuracy, the chosen method is also cross validation as same as used in the validation of IEBRG. The reason is the same as described earlier with regard to validation of IEBRG. With respect to efficiency, in addition to those measures used in the validation of IEBRG as mentioned earlier, another measure is taken into use and referred to as number of backward steps. The number of backward steps indicates the number of rule terms which are generated but eventually discarded during rule generation. This measure is used to reflect necessity of computation in data mining context and effectiveness of learning in machine learning context. If the number of backward steps is very large, this means that the algorithm generates large number of rule terms which are eventually discarded and thus results in large amount of unnecessary computational costs due to ineffective learning.

4.3.2 Integrated Testing

This part of experimental study includes the validation of collaborative and competitive random decision rules and collaborative rule generation. For both approaches mentioned above, the validation involves the combination of different methods for rule generation and rule simplification such as Prism, IEBRG and Jmid-pruning. This is in order to show not only the performance of the general framework of ensemble learning but also to what extent the standard methods for rule generation and simplification contribute towards the improvement of overall accuracy of classification.

With regard to collaborative and competitive random decision rules, the validation is against Random Prism in terms of classification accuracy. The validation of accuracy is done by splitting a data set into a training set and a test set in the ratio of 80:20. For each data set, the experiment is repeated 10 times and the average of the accuracies is taken for comparative validation. The reason is that ensemble learning is usually computationally more expensive because the size of the data set dealt with by ensemble learning is as same as n times the size of the original data set. In other words, a data set should be pre-processed to get n samples, each of which has the same size of the original data set. In addition, the proposed ensemble learning method includes two or more learning algorithms in general (three learning algorithms in this experiment) used for each of the n samples. Therefore, in comparison with single learning tasks such as use of IEBRG or Prism, the computational efforts would be as same as 3*n times that conducted by a single learning task. In this situation, the experimental environment would be quite computationally constrained on a single computer if cross validation is used to measure the accuracy. On the other hand, instances in each sample are randomly selected with replacement from the original data set. Thus the classification results are not deterministic and the experiment is setup in the way mentioned above to make the results more convincing. Besides, the accuracy performed by random guess is also calculated and compared with that performed by each chosen algorithm. This is in order to check whether a chosen algorithm really works on a particular data set as mentioned earlier.

The validation of the CCRDR does not include measure of efficiency. This is because, on the basis of above descriptions, the computation conducted using the proposed method is theoretically much more complex than Random Prism if it is done on a single computer.

From this point of view, the comparison between the two ensemble learning approaches would be quite unfair and less implicative for practical applications. However, the efficiency can be easily improved in practice by adopting parallel data processing techniques and is thus not a critical issue. On the other hand, the CCRDR is a general framework which could employ any rule based methods involved in training stage in principle. From this point of view, the efficiency is approximately equal to the sum of efficiency for each single algorithm if the experiment is run on a single computer. Otherwise, it would be equal to the efficiency for the branch with the worst outcome if the experiment is done on a parallel computer. In other words, the efficiency does not directly depend on the framework but mainly on the performance of each single algorithm employed. In this experimental setup, all of the chosen rule based methods are validated individually as mentioned earlier in this chapter. Therefore, it is not necessary to undertake the redundant validation again.

With regard to collaborative rule generation, the validation aims to indicate that the combination of different rule based learning algorithms usually improves the overall accuracy compared with the use of each single learning algorithm. In particular, on algorithms side, two single learning algorithms namely, Prism and IEBRG, are chosen to be the base algorithms for the CRG framework. In general, this framework can employ any algorithms, which follow the separate and conquer rule learning approach, to be combined for the generation of a rule set. In this experimental study, there are only two algorithms chosen due to the consideration of computational constraints. The computational complexity of this kind of ensemble learning approaches is approximately *n* times the complexity of a single learning algorithm, where n is the number of base learning algoirthms, if no parallelisation is adopted. The reason why the Prism algorithm is chosen is due to the advantage that this algorithm can typlically overcome some limitations of decision tree learning algorithms to a large extent, such as the replicated subtree problem as mentioned in Chapter 2. The IEBRG algorithm is also chosen because it complements well the Prism algorithm with regard to some of its disadvantages. In fact, the aim of the CRG approach is to enable that combined algorithms complement each other. In other words, the disadvantages of one algorithm could be overcome by the advantages of another algorithm. Therefore, it would be appropriate to choose algorithms that have different advantages and disadvantages and that are complementary to each other.

On the other hand, as mentioned in Section 3, the CRG approach involves measuring the quality of each single rule generated. In this context, the approach needs to employ at least one of the measures of rule quality to judge which one of the generated rules is of the highest quality. In this experimental study, the four measures, namely confidence, J-measure, lift and leverage, are chosen due to their significance and popularity in real applications (Tan, Kumar, & Srivastava, 2004).

Under the above setup, for the measure of classification accuracy, the experiments are conducted by splitting a data set into a training set and a test set in the ratio of 70:30. For each data set, the experiment is done 10 times and the mean of the accuracies are calculated for comparative validation. As mentioned earlier, ensemble learning approaches are usually computationally more expensive. Therefore, cross validation is not adopted in this study. On the other hand, for the measure of rule quality, the whole training set is used to evaluate each single rule generated with regard to the quality of this rule for each fold of validation.

4.3.3 System Testing

This part of experimental study involves validating the combination of CCRDR and CRG as a hybrid ensemble learning approach. This is in order to show more comprehensive improvement of overall classification accuracy through collaborations in both training and testing stages. In particular, Bagging is used to scale down data as same as the setup in CCRDR. However, on each sample of training data drawn by the Bagging, the CRG framework is adopted to have all employed rule learning algorithms (Prism and IEBRG) collaborate to generate a single rule set. In addition, J-measure is used as the measure of rule quality. In testing stage, the independent predictions by these generated rule sets are combined to predict to which class an unseen instance belong. The voting for final prediction is based on precision as already proven in integrated testing that precision is more effective and confident in measuring the reliability of a classifier in predicting a particular class. The experimental setup is the same as that for the CCRDR framework as specified in Section 4.3.2.

4.4 Results and Discussion

As mentioned in Section 4.3, the validation of IEBRG against Prism is divided into two parts. One is in noise free domain and the other one is in noise domain. The results in classification accuracy are reflected from Table 4.1 as below.

Table 4.1 reflects that on all of the data sets both IEBRG and Prism perform better accuracies than random classifier which makes prediction by random guess. This indicates that both algorithms really work on the chosen data sets. With regard to the comparison between IEBRG and Prism, Table 4.1 reflects that IEBRG outperforms Prism in 16 out of the 20 cases. For the rest of the cases, IEBRG performs the same as Prism in two cases on *page-blocks* and *dorothea* data sets and worse on the other two cases on *ionosphere* and *mushroom*. Although IEBRG sticks out on the two data sets, the accuracy is still close to that performed by Prism.

Table 4.1 Accuracy for IEBRG vs Prism

Dataset	Prism	IEBRG	Random classifier
anneal	80%	**85%**	60%
balance-scale	37%	**65%**	43%
credit-g	62%	**67%**	58%
credit-a	59%	**77%**	50%
diabetes	64%	**72%**	54%
heart-statlog	66%	**68%**	50%
ionosphere	**89%**	83%	54%
iris	69%	**93%**	32%
kr-vs-kp	52%	**84%**	50%
lymph	69%	**79%**	47%
segment	53%	**74%**	14%
zoo	63%	**88%**	20%
wine	81%	**89%**	33%
car	70%	**72%**	33%
breast-w	90%	**93%**	55%
mushroom	**93%**	91%	50%
page-blocks	**92%**	**92%**	80%
Monks-problems-3	50%	**84%**	50%
dorothea	**93%**	**93%**	50%
dexter	78%	**83%**	50%

The results in computational efficiency are reflected from Table 4.2-4.4 in terms of computational complexity and runtime. Table 4.5 shows clash rate which helps explain why some rules are discarded.

Table 4.2 is used to reflect the complexity of the rule sets generated by IEBRG or Prism on the chosen data sets. The complexity of a rule set would be an impact factor for computational efficiency in prediction stage. As mentioned in Chapter 2, decision tree learning algorithm is criticised due to the generation of a large number of complex rules. It indicates that a highly complex rule set usually makes it difficult and computationally expensive to extract a classification assigned to an unseen instance in prediction stage. Therefore, the complexity of a rule set could be used to pre-measure the approximate level of efficiency in testing stage. It can be seen from Table 4.2 that IEBRG generates more general and fewer rules than Prism in 13 out of the 20 cases. This indicates that in most cases IEBRG not only needs cheaper computation than Prism in generation of a rule set but also provides a simpler rule set which makes it cheaper to predict further unseen instances in testing stage. In addition, although IEBRG generates more complex rule sets than Prism on three data sets, namely *balance-scale*, *car* and *dorothea*, Prism discard a large number of rules and thus a large number of rule terms in two out of the three cases on *balance-scale* and *car* data sets due to the way of dealing with clashes. This still indicates that Prism is computationally more expensive than IEBRG in generation of a rule set. This is because discarded rules and rule terms also need to conduct computation for their generation although they are eventually discarded. The action for discarding rules and rule terms would also potentially result in underfitting of training data and thus loss of accuracy as mentioned in Chapter 2. This implies another negative phenomenon of Prism.

Table 4.2 Number of rules and average number of terms for IEBRG vs Prism

Dataset	Prism		IEBRG	
	Count(rules)	Ave(terms)	Count (rules)	Ave(terms)
anneal	12	4.92	**8**	**1.0**
balance-scale	**13**	**2.54**	21	3.05
credit-g	15	2.07	**8**	**1.0**
credit-a	8	1.0	**7**	**1.0**
diabetes	8	1.125	**8**	**1.0**
heart-statlog	8	1.0	**6**	**1.0**
ionosphere	**2**	**1.0**	**2**	**1.0**
iris	5	1.0	**4**	**1.0**
kr-vs-kp	17	1.71	**9**	**1.0**
lymph	**6**	**1.0**	**6**	**1.0**
segment	12	1.25	**9**	**1.0**
zoo	7	1.0	**5**	**1.0**
wine	**5**	**1.0**	**5**	**1.0**
car	**3**	**1.0**	23	3.96
breast-w	8	1.125	**6**	**1.0**
mushroom	10	1.0	**9**	**1.0**
page-blocks	10	2.7	**8**	**1.0**
Monks-problems-3	9	5.78	**7**	**1.86**
dorothea	**5**	**1.0**	7	1.0
dexter	**7**	**1.0**	**7**	**1.0**

Table 4.3 Numbers of generated terms and discarded terms

Dataset	Prism		IEBRG	
	Generated terms	Discarded terms	Generated terms	Discarded terms
anneal	59	0	**8**	**0**
balance-scale	33	44	**64**	**0**
credit-g	31	0	**8**	**0**
credit-a	8	0	**7**	**0**
diabetes	9	0	**8**	**0**
heart-statlog	8	0	**6**	**0**
ionosphere	**2**	**0**	**2**	**0**
iris	5	0	**4**	**0**
kr-vs-kp	29	0	**9**	**0**
lymph	**6**	**0**	**6**	**0**
segment	13	0	**9**	**0**
zoo	7	0	**5**	**0**
wine	**5**	**0**	**5**	**0**
car	3	276	**91**	**0**
breast-w	9	0	**6**	**0**
mushroom	10	0	**9**	**0**
page-blocks	27	60	**8**	**0**
Monks-problems-3	52	90	**13**	**0**
dorothea	**5**	**0**	7	0
dexter	**7**	**0**	**7**	**0**

Table 4.3 is used to reflect the approximate number of iterations conducted during generation of a rule set. In particular, the number of generated rule terms plus that of discarded terms would be the approximate number of iterations conducted in training stage. This is because all discarded rule terms should be generated first and eventually discarded due to the discarding of the corresponding rules to which the terms belong. This table reflects that in 14 out of 20 cases IEBRG generates less number of rule terms than Prism and thus needs less number of iterations to generate a rule set. In three cases on *balance-scale*, *car* and *dorothea* respectively, IEBRG generates a larger number of rule terms than Prism but there is a far large number of terms discarded by Prism due to the way of dealing with clashes as mentioned in Chapter 2. This still indicates that Prism needs a larger number of iterations to generate a rule set and is thus computationally more expensive. In addition, there are other two cases on *page-blocks* and *Monks-problems-3* respectively that Prism discards a far large number of rule terms. On the other hand, Table 4.3 is also used to reflect the approximate number of iterations conducted for predicting an unseen instance in testing stage while a particular rule representation is determined. In detail, as mentioned in Chapters 2 and 3, the overall number of terms generated is used as the input size for measuring the time complexity using BigO notation. As mentioned above, there are only three cases that IEBRG generates a more complex rule set than Prism. This indicates that in most cases IEBRG generates a rule set that makes it faster to make a classification on an unseen instance in testing stage comparing with Prism if the rule representation is kept same. In other words, rule representation is another impact factor for efficiency in prediction stage as mentioned in Chapter 2. The comparison among different representations has been introduced in Chapter 3 with respect to theoretical validation of rule based networks against linear list.

Table 4.4 Runtime in milliseconds for IEBRG vs Prism

Dataset	Prism	IEBRG
anneal	1016	**187**
balance-scale	109	**78**
credit-g	906	**125**
credit-a	398	**64**
diabetes	343	**94**
heart-statlog	94	**15**
ionosphere	579	**141**
iris	47	**15**
kr-vs-kp	797	**578**
lymph	**15**	16
segment	4219	**2984**
zoo	31	**16**
wine	140	**31**
car	578	**125**
breast-w	47	**31**
mushroom	1641	**1515**
page-blocks	12985	**3188**
Monks-problems-3	125	**16**
dorothea	**761871**	964966
dexter	65691	**31670**

Table 4.4 shows empirical comparison between IEBRG and Prism in terms of runtime with regard to efficiency in training stage. It can be seen from this table that IEBRG is faster than Prism in generation of a rule set in 18 out of the 20 cases. The only case that IEBRG falls a bit far behind Prism is on *dorothea* data set. This is also the only case that IEBRG conducts a larger number of iterations than Prism in training stage through looking at Table 4.3. In addition, there is another case that IEBRG is marginally slower than Prism on *lymph* data set. This could be explained by the time complexity analysis of the two algorithms as illustrated below:

Time complexity analysis for IEBRG with regards to the time of rule term generation:

Suppose a data set has i instances and a attributes so the size is i× a as well as v attribute-value pairs and c classifications

Step 1: create a frequency table

Time complexity: i× v+ i× a× c

Step 2: calculate conditional entropy for attribute value

Time complexity: v× c

Step 3: rank the conditional entropy for all attribute values

Time complexity: v+ a

Step 4: split the dataset by deleting the instances that don't comprise the attribute-value pair

Time complexity: i

Therefore, the time complexity is: O (i× v+ i× a× c+ v× c+ v+ a + i), for the generation of each rule term while the input size (n) is the total number of rule terms.

Time complexity analysis for Prism with regards to the time of rule term generation:

Suppose a data set has i instances and a attributes so the size is m× n as well as v attribute-value pairs and c classifications

Step 1: create a frequency table

Time complexity: i× v+ i× a

Step 2: calculate posterior probability of a target class given an attribute value as condition

Time complexity: v

Step 3: rank the posterior probability for all attribute values

Time complexity: v+ a

Step 4: split the dataset by deleting the instances that don't comprise the attribute-value pair

Time complexity: i

Therefore, the time complexity is: O (i× v+ i× a+ v + v+ a + i), for the generation of each rule term while the input size (n) is the total number of rule terms.

The above analysis shows that IEBRG is computationally more complex than Prism with regard to the generation of each single rule term on the basis of similar size of training data. The time complexity for the entire training stage would approximately be the sum of the complexities for the generation of all rule terms respectively, which is referred to as summative time complexity. This is why on *lympth* data set IEBRG generates the same numbers of rules and rule terms as Prism but is a bit slower than the latter algorithm in runtime.

The above way of complexity analysis would usually help check the reasons while the runtime performed by a particular algorithm looks odd and unexpected. However, this way may not be very effective in measuring the computational complexity of an algorithm globally by means of analysing the entire procedure. This is because reuse engineering is increasingly popular and thus practitioners usually reuse the APIs, which are usually invisible, for their implementations. Therefore, this objective situation results in the difficulty for more precise analysis of computational complexity.

The results shown in Tables 4.2 and 4.3 prove the correlation that a large number of iterations conducted in training stage usually leads to a slow process of generating a rule set. There are also results on other noise free data sets with respects to classification accuracy and computational efficiency available to view in Appendix VI. The results also reflect the similar phenomenon that IEBRG outperforms Prism in most cases in terms of accuracy and efficiency. In addition, the results are also shown in terms of individual accuracy for each single classification in Appendix V.

Table 4.5 Clash rate for IEBRG vs Prism

Dataset	Prism	IEBRG
anneal	0.0	0.0
balance-scale	0.5	**0.43**
credit-g	0.0	0.0
credit-a	0.0	0.0
diabetes	0.0	0.0
heart-statlog	0.0	0.0
ionosphere	0.0	0.0
iris	0.0	0.0
kr-vs-kp	0.0	0.0
lymph	0.0	0.0
segment	0.0	0.0
zoo	0.0	0.0
wine	0.0	0.0
car	0.94	**0.39**
breast-w	0.0	0.0
mushroom	0.0	0.0
page-blocks	0.375	0.0
Monks-problems-3	0.67	**0.14**
dorothea	0.0	0.0
dexter	0.0	0.0

Besides, the comparison between the algorithms mentioned above is also done in noise domain. The chosen data sets include *breast cancer*, *kr-vs-kp*, *contact lenses*, *zoo* and *lympth*. The results are available to view in Appendix VI. These figures reflect that except for *contact lenses* data set IEBRG performs higher tolerance to noise than Prism in almost all of the cases. There is a phenomenon in some cases that the increase of noise levels does not lead to the decrease of accuracy levels. This could be partially explained on the basis of pattern consistency. The accuracy is actually dependent on the consistence between the pattern learned from the training set and the pattern that exists in the test set. If they are highly consistent, then the accuracy would be higher. Otherwise, it would be lower. Noise is actually introduced artificially to both training and test data. If the accuracy is higher, that means that the pattern learned from the training set is highly consistent with that exists in the test set. However, if the same percentage of noise is introduced to training and test sets, the patterns between the two data sets could become very inconsistent. This is because the training set is obviously larger than the test set in the experimental setup. In other words, the pattern in the training set may get a relatively small change but that in the test set may get a relatively large change. From this point of view, the accuracy may be obviously worse than that achieved in noise free data sets when a little noise is added. However, it may be increasing as the increase of noise levels in test sets as the pattern that exists in test sets would gradually get more consistent with that learned from training sets. In another case, if a large percentage of noise, say 50%, is introduced to the training set but only a small percentage, say 10%, to the test set, it may result in that the pattern learned from the training set gets a large change but that exists in the test set only gets a small change. Therefore, it could have the tendency that the accuracy would increase as the increase of noise levels in test sets.

With regard to the validation of Jmid-pruning against J-pruning and Jmax-pruning, the experiment is divided into two parts. The first part is to prove that the accuracy could be lost

if the tie-breaking on J-measure, as mentioned in Chapter 2, is not dealt with effectively and that Jmid-pruning usually generates fewer rule terms that are eventually discarded and thus needs a smaller number of iterations in order to find the global maximum of J-measure. Furthermore, the second part of experiment is undertaken to prove empirically that the reduction in number of iterations would usually speed up the process of rule generation. Therefore, in this part of experiment, each of the chosen data sets matches the characteristic that Jmax-pruning and Jmid-pruning generates the same set of rules and thus performs the exactly same classification accuracy. This is in order to make fair comparisons on the extent to which the computational efficiency is affected due to the increase/decrease of the number of backward steps. This setup of experiments also helps with the scientific proof with respect to the approximate correlation between number of backward steps and runtime.

Table 4.6 Accuracy for pruning methods

Dataset	J-pruning	Jmax-pruning	Jmid-pruning
Vote	**97%**	**97%**	**97%**
Weather	**83%**	**83%**	**83%**
Contact-lenses	80%	**85%**	**85%**
Lense24	67%	**75%**	**75%**
Breast-cancer	55%	**58%**	**58%**
Car	74%	74%	**78%**
Lung-cancer	**95%**	**95%**	**95%**
Iris	67%	77%	**82%**
Segment	53.1%	53.3%	**53.8**
ionosphere	**87%**	**87%**	**87%**

It can be seen from Table 4.6 that Jmid-pruning leads PrismTCS to perform a similar level of classification accuracy in comparison with J-pruning and Jmax-pruning in 7 out of 10 cases but outperforms the two algorithms in the other cases. With regards to efficiency, Table 4.7 shows that PrismTCS with Jmid-pruning generates a rule set with a similar level of rule complexity or even fewer but more general rules in comparison with J-pruning and Jmax-pruning. However, Table 4.8 shows that Jmid-pruning performs better compared with Jmax-pruning in terms of computational efficiency. It can be seen by looking at the number of backward steps that Jmid-pruning needs a smaller number of iterations than Jmax-pruning to make Prism stop generating rules. Therefore, Jmid-pruning is computationally more efficient from theoretical point of view.

Table 4.7 Number of rules and terms per rule for pruning methods

Dataset	J-pruning		Jmax-pruning		Jmid-pruning	
	Count(rules)	Ave(terms)	Count(rules)	Ave(terms)	Count(rules)	Ave(terms)
Vote	**2**	**2.5**	5	4.2	**2**	**2.5**
Weather	**3**	**1.67**	3	1.7	**3**	**1.67**
Contactlenses	3	1.67	3	1.67	3	1.67
Lense24	**4**	**1.5**	4	2.25	4	2.0
Breast-cancer	8	1.125	**7**	**1.0**	**7**	**1.0**
Car	**3**	**1.0**	**3**	**1.0**	**3**	**1.0**
Lung-cancer	**4**	**1.0**	**4**	**1.0**	**4**	**1.0**
Iris	**5**	**1.0**	**5**	**1.0**	**5**	**1.0**
Segment	11	1.09	13	1.69	**10**	**1.0**
ionosphere	**2**	**1.0**	**2**	**1.0**	**2**	**1.0**

Table 4.8 Number of discarded rules and backward steps for pruning methods

Dataset	J-pruning	Jmax-pruning		Jmid-pruning	
	Discarded rules	Discarded rules	Backward steps	Discarded rules	Backward steps
Vote	**4**	**4**	154	**4**	**5**
Weather	**1**	2	3	**1**	**1**
Contact-lenses	**1**	**1**	4	**1**	**2**
Lense24	2	**1**	5	2	**3**
Breast-cancer	**1**	2	**1**	2	**1**
Car	**12**	46	207	**12**	**10**
Lung-cancer	**0**	**0**	0	**0**	0
Iris	**0**	**0**	0	**0**	0
Segment	5	**3**	7	4	**6**
ionosphere	**0**	**0**	0	**0**	0

Table 4.9 Accuracy for Jmid-pruning vs Jmax-pruning

Dataset	Jmax-pruning	Jmid-pruning	Random classifier
cmc	55%	55%	35%
vote	97%	97%	52%
kr-vs-kp	55%	55%	50%
ecoli	62%	62%	27%
anneal.ORIG	78%	78%	60%
audiology	51%	51%	13%
car	74%	74%	33%
optdigits	47%	47%	10%
glass	53%	53%	24%
lymph	76%	76%	47%
yeast	55%	55%	21%
shuttle	92%	92%	65%
analcatdata_asbestos	73%	73%	43%
analcatdata_happiness	63%	63%	30%
breast-cancer	69%	69%	58%

The second part of experimental results is reflected from Table 4.9-4.11 with respects to classification accuracy and computational efficiency. Table 4.9 reflects that in all of the cases both Jmax-pruning and Jmid-pruning outperforms random classifier which makes classification by random guess in terms of classification accuracy. In addition, Jmid-pruning performs the same accuracy as Jmax-pruning in all of the cases. The reason is explained earlier in the section. These examples are particularly chosen in order to focus on the special validation that it is really scientifically possible that Jmid-pruning finds the global maximum of J-measure earlier than Jmax-pruning and thus needs a smaller number of backward steps in training stage if the two algorithms make PrismTCS generate the same rule set. In this case, the two algorithms perform the same in accuracy but Jmid-pruning makes PrismTCS complete rule generation earlier and is thus more efficient in comparison with Jmax-pruning.

Through looking at Table 4.10 and 4.11, the results show that Jmid-pruning outperforms Jmax-pruning in terms of both the number of backward steps and runtime in all of the cases except on *lymph* data set. The results also reflect the approximate correction between the two aspects mentioned above that the reduction in the number of backward steps would speed up the process of rule generation in training stage, especially when the difference in the number

of backward steps is significant. In addition, on *lymph* data set, Jmid-pruning takes the same number of backward steps as Jmax-pruning but is a little bit behind the latter algorithm in runtime. This is because Jmid-pruning needs to take time to calculate the value of Jmax to measure the maximum of J-measure, which may be achieved eventually during rule generation, whereas Jmax-pruning does not as mentioned in Chapter 3. The above description indicates that Jmid-pruning makes it scientifically achievable to help rule based classifiers improve efficiency without loss of accuracy and even to improve accuracy when tie-breaking on J-measure really arises in practice.

Table 4.10 Number of backward steps for Jmid-pruning vs Jmax-pruning

Dataset	Jmax-pruning	Jmid-pruning
cmc	329	**307**
vote	50	**46**
kr-vs-kp	595	**399**
ecoli	44	**25**
anneal.ORIG	874	**315**
audiology	263	**140**
car	117	**113**
optdigits	1461	**1278**
glass	9	**6**
lymph	**2**	**2**
yeast	43	**11**
shuttle	131	**113**
analcatdata_asbestos	2	**1**
analcatdata_happiness	1	**0**
breast-cancer	34	**25**

Table 4.11 Runtime in milliseconds for Jmid-pruning vs Jmax-pruning

Dataset	Jmax-pruning	Jmid-pruning
cmc	5000	**4625**
vote	812	**625**
kr-vs-kp	9078	**7500**
ecoli	1359	**1125**
anneal.ORIG	17595	**15891**
audiology	54549	**53580**
car	1062	**1047**
optdigits	400404	**364730**
glass	1829	**1562**
lymph	**60**	63
yeast	2141	**2125**
shuttle	95442	**95411**
analcatdata_asbestos	47	**46**
analcatdata_happiness	16	**15**
breast-cancer	47	**31**

With regard to rule based network, it is validated theoretically against linear list in term of time complexity using BigO notation (Cormen , Leiserson , Rivest , & Stein, 2001). As mentioned in Chapter 2, the network representation could achieve that prediction process is run in divide and conquer search and the efficiency is O (log (n)), where *n* is the overall

number of rule terms in a rule set. In contrast, list representation could only achieve a linear search process for the same purpose and the efficiency is O (n). The above comparison indicates that the network representation usually makes it faster to make a classification on an unseen instance comparing with list representation for the same rule set. Thus rule representation is another impact factor for computational efficiency in testing stage as mentioned earlier. In practice, for the purpose of predictive modelling, the network representation would contribute as many quicker decisions as possible in prediction stage in expert systems. The difference to the listed rule representation in the efficiency would be significant especially when Big Data is used to generate a rule set.

With regard to CCRDR, the validation is divided into two parts of comparison. The first part is to prove empirically that combination of multiple learning algorithms would usually outperforms a single algorithm as a base algorithm for ensemble learning with respect to accuracy. The second part is to prove that use of precision instead of overall accuracy or recall as the weight of a classifier would be more reliable in making final predictions. In Table 4.12, the CCRDR I represents that the weight of a classifier is determined by the overall accuracy of the classifier. In addition, the CCRDR II and III represent that the weight is determined by precision for the former and by recall for the latter.

Table 4.12 Ensemble learning results for CCRDR

Dataset	Random Prism	CCRDR I	CCRDR II	CCRDR III	Random classifier
anneal	71%	78%	79%	**80%**	60%
balance-scale	44%	56%	**68%**	64%	43%
diabetes	66%	68%	**73%**	68%	54%
heart-statlog	68%	71%	**74%**	63%	50%
ionosphere	65%	68%	**69%**	65%	54%
lympth	68%	60%	**89%**	65%	47%
car	69%	68%	**71%**	70%	33%
breast-cancer	70%	72%	**74%**	73%	58%
tic-tac-toe	63%	65%	66%	**67%**	55%
breast-w	**85%**	75%	81%	75%	55%
hepatitis	81%	84%	**87%**	82%	66%
heart-c	70%	74%	**83%**	65%	50%
lung-cancer	75%	79%	**88%**	75%	56%
vote	67%	82%	**95%**	80%	52%
page-blocks	**90%**	**90%**	**90%**	89%	80%

The results in Table 4.12 show that all of the chosen methods outperform the random classifier in classification accuracy. This indicates that all of the methods really work on the chosen data sets. In the comparison between Random Prism and CCRDR I, the results show that the latter method outperforms the former one in 12 out of 15 cases. This proves empirically that combination of multiple learning algorithms usually helps generate a stronger hypothesis in making classifications. This is because the combination of multiple algorithms could achieve both collaboration and competition. The competition among these classifiers, each of which is built by one of the chosen algorithms, would make it achievable that for each sample of training data the learner constructed is much stronger. All of the stronger learners then effectively collaborate on making classifications so that the predictions would be more accurate.

As mentioned earlier, the second part of comparison is to validate that precision would usually be a more reliable measure than overall accuracy and recall for the weight of a classifier. The results in Table 4.12 indicate that in 13 out of 15 cases CCRDR II outperforms CCRDR I and III. This is because in prediction stage each individual classifier would first make a classification independently and their predictions are then combined in making the final classification. For the final prediction, each individual classifier's prediction would be assigned a weight to server for the final weighted majority voting. The weight is actually used to reflect how reliable the individual classification is. The heuristic answer would be based on the historical record that how many times the classifier has recommended this classification and how correct it is. This could be effectively measured by precision. The weakness of overall accuracy is that this measure can only reflect the reliability of a classifier on average rather than in making a particular classification as mentioned in Chapter 2. Thus overall accuracy cannot satisfy this goal as mentioned above. In addition, although recall can effectively reflect the reliability of a classifier in making a particular classification, the reliability is affected by the frequency of a particular classification and thus cheats the final decision maker, especially when the frequency of the classification is quite low as mentioned in Chapter 3. Therefore, the results prove empirically that precision would be more reliable in determining the weight of a classifier for weighted majority voting.

The above description with regard to CCRDR validates that combination of multiple learning algorithms would be more effective for improving the overall accuracy of classification and that precision would be a more reliable measure in determining the weight of a classifier to successfully serve for weighted majority voting, especially on unbalanced data sets.

In statistical analysis, there are some popular measures with respect to errors such as mean absolute error (MAE) and squared error (SE), which are widely used in regression tasks. However, they are not adopted for results analysis in this thesis. This is because this thesis focuses on classification tasks and it is required to be able to calculate the distance between the predicted class and the actual class for this type of tasks. For two class classification tasks, it is always the case that the error is 0 if the prediction is correct and 1 otherwise, which means that it is impossible to identify extent to which the prediction is incorrect. For multi-class classification tasks, there is usually no way to rank the classes except for the case that the class attribute is an ordinary attribute such as 'very large', 'large', 'medium', 'small', 'very small'. If the classes cannot be ranked, it indicates that there is no way to calculate the difference between classes and thus it is unable to identify the extent to which the classification is incorrect. On the other hand, classification is an approach of decision making. Incorrect decisions are usually not expected in real applications. For example, when a student takes a multiple choice for an exam, there could only be one right choice and the rest of the choices have different distances to the right one. In this context, it may be highly possible that the student only makes a minor mistake and gets a wrong answer which is the closest to the right answer. However, the outcome would be that the student cannot gain any marks for this question. In general, both minor and major mistakes may result in a failure, which is not expected. From this point of view, classification tasks would always aim to have as many classes correctly predicted as possible. Therefore, overall accuracy is usually seen as the most important measure to reflect the performance of a classification method.

Table 4.13 Ensemble learning results for CRG

Dataset	Prism with Jmid-pruning	IEBRG with no pruning	CRG with confidence	CRG with J-measure	CRG with lift	CRG with leverage
anneal	68%	90%	**92%**	88%	90%	89%
credit-g	62%	67%	65%	69%	**72%**	70%
diabetes	56%	70%	72%	68%	69%	**75%**
heartstatlog	64%	66%	75%	74%	**77%**	76%
ionosphere	**89%**	81%	84%	84%	**89%**	84%
iris	72%	93%	92%	92%	94%	**97%**
kr-vs-kp	61%	83%	**93%**	92%	92%	**93%**
lymph	73%	70%	71%	75%	73%	**83%**
segment	51%	68%	73%	**81%**	77%	78%
zoo	59%	79%	**87%**	85%	82%	85%
wine	83%	91%	89%	**92%**	91%	**92%**
breastcancer	66%	69%	71%	**72%**	71%	**72%**
car	69%	76%	**77%**	75%	**77%**	76%
breast-w	94%	**95%**	93%	92%	94%	92%
credit-a	62%	69%	78%	80%	**81%**	78%
heart-c	62%	69%	71%	**77%**	74%	74%
heart-h	69%	74%	77%	78%	**81%**	78%
hepatitis	84%	82%	83%	80%	**86%**	82%
mushroom	**98%**	**98%**	96%	96%	**98%**	**98%**
vote	92%	90%	**95%**	94%	**95%**	94%

Results in Table 4.13 show the comparison among CRG with different measures of rule quality, Prism and IEBRG in terms of classification accuracy. This part of validation is with regard to the performance of the CRG.

With regard to classification accuracy, Table 4.13 shows that the CRG approach, which has different variants, outperforms both of Prism and IEBRG in 17 out of 20 cases. This indicates that the combination of different rule learning algorithms usually improves the overall accuracy of classification as expected. In some cases (on *heart-statlog*, *ks-vs-kp*, *segment* and *credit-a* data sets), the CRG approach even outperforms both of the two base algorithms to a large extent. This phenomenon can support the argument that two algorithms can be complementary to each other, especially on the basis that they have different advantages and disadvantages and that they are combined in an effective way. On the other hand, the results show that this approach has a bias on the chosen measure of rule quality. It can be seen from Table 4.13 on the data sets, *anneal*, *ionosphere*, *iris*, *lymph*, *wine*, *car*, *breast-w*, *hepatitis* and *mushroom*, that at least one of the measures of rule quality fails to help outperform both of the two base learning algorithms namely, Prism and IEBRG. This phenomenon is also due partially to the variance on data side but it is still critical to appropriately choose the measure of rule quality to reduce the bias on the algorithms side.

Overall, the empirical results shown in Tables 4.13 indicate that the CRG approach is useful for improving the quality of each single rule generated on average and thus improving the overall accuracy. In machine learning tasks, the main concern of a rule based learning algorithm is typically about using a rule set as a whole to accurately predict on unseen instances. In this context, some rules that are of low quality may be rarely or even never used to predict. In this case, although the accuracy may not be seriously affected, the improvement for the quality of each single rule is still necessary towards the improvement of overall accuracy, especially when a large set of test instances are used. On the other hand, the rules generated in data mining tasks aim for knowledge usuage. From this point of view,

the main concern would be about the reliability of each single rule when the rule is used to provide insights for a knowledge domain. This even makes it necessary to a larger extent to improve the quality of each single rule. Besides, for separate and conquer rule learning, the generation of each single rule would affects that of all subsequent rules. In other words, the quality of each single rule generated would lead to a chained impact on the generation of all subsequent rules. Therefore, it is important to ensure that each single rule is generated to have a quality as high as possible. On the basis of above description, the CRG approach introduced in Chapter 3 is worth to be developed further especially on reduction of bias originating from algorithms, such as choice of rule quality measures.

Table 4.14 Ensemble learning results for hybrid approach

Dataset	Random Forests	CCRDR	Hybrid
credit-a	85%	70%	**87%**
credit-g	72%	71%	**74%**
vote	97%	93%	**98%**
hepatitis	85%	84%	**92%**
lung-cancer	70%	86%	**93%**
lymph	86%	70%	**90%**
breast-cancer	65%	78%	**81%**
breast-w	**97%**	85%	91%
labor	88%	**90%**	88%
heart-h	83%	79%	**85%**

Table 4.14 shows that the hybrid ensemble rule based classification framework outperforms random forests and CCRDR in 8 out of 10 cases. On *breast-w* and *labor* data sets, the hybrid ensemble learning framework performs a bit worse than random forests and CCRDR.

The results indicate that it is necessary to take both scaling up algorithms and scaling down data in order to comprehensively improve classification accuracy like the hybrid ensemble rule based classification framework. In this way, accuracy can be improved through reduction of both bias and variance. In contrast, random forests only involves scaling down data and nothing on scaling up algorithms. Therefore, random forests only enables the reduction of variance on data side but is biased on the decision tree learning algorithm chosen. CCRDR enables the reduction of both bias and variance. However, on algorithms side, the chosen algorithms do not collaborate with each other and thus the reduction of bias is not sufficient. This could be explained by the assumption that each algorithm may generate a rule set that has some rules of high quality but the others of low quality. In other words, it cannot ensure that each single rule is generated to have a high quality and thus may result in incorrect classifications by low quality rules.

On the basis of above discussion, the hybrid ensemble rule based classification framework is strongly motivated due to its flexibility in employing rule learning algorithms and rule quality measures, as well as its involvement that different rule learning algorithms collaborate to complement each other.

4.5 Conclusion

This chapter shows empirical results for the validations of IEBRG against Prism, Jmid-pruning against J-pruning and Jmax-pruning and CCRDR against Random Prism. This chapter also shows theoretical results for the validation of rule based networks against linear list. The results show that IEBRG outperforms Prism in both accuracy and efficiency in

noise free domain and is also equally comparative to Prism in noise tolerance level. With regards to Jmid-pruning, the results show Jmid-pruning leads PrismTCS to perform a similar level of classification accuracy in comparison with J-pruning and Jmax-pruning in most cases but outperforms the two algorithms in other cases. With regards to efficiency, Jmid-pruning makes PrismTCS generate a rule set with a similar level of rule complexity or even fewer but more general rules in comparison with J-pruning and Jmax-pruning. However, Jmid-pruning usually performs better compared with Jmax-pruning in terms of computational efficiency. It is proven in both theoretical analysis and empirical validation that Jmid-pruning makes Prism conduct a smaller number of iterations and faster in generating a rule set. Therefore, Jmid-pruning seems likely to be computationally more efficient especially when training data is very large. In addition, rule based network is proven to be likely to make faster predictions than linear list in testing stage. The validation of CCRDR also indicates that it usually helps improve the overall classification accuracy to combine multiple learning algorithms with collaborations and competitions and to measure the weight of a classifier using precision or recall instead of overall accuracy. The validation of CRG also indicates that collaboration and competition involved per each rule generated would usually help improve the overall quality of a rule set and thus improve the overall accuracy of classification. The hybrid ensemble rule based classification framework, which involves the combination of CCRDR and CRG, is also validated empirically by comparing its performance with Random Forests and CCRDR. The results indicate that the hybrid approach is helpful to improve overall classification accuracy through reduction of both bias and variance. The implication of these results is further discussed in qualitative analysis against the research objectives in Chapter 5.

Chapter 5 Qualitative Evaluation

5.1 Introduction

Chapter 4 describes the empirical validation of IEBRG, Jmid-pruning, CCRDR and CRG, and theoretical validation of rule based network representation as well as discusses the results in quantitative context. The results indicate that these approaches usually outperform the existing ones introduced in Chapter 2. This chapter evaluates the completed work against the research objectives listed in Chapter 1 in qualitative context and to clarify the extent to which the completed work is significant in scientific context.

5.2 Critical Discussion against Objectives

This thesis introduces a unified framework for design of rule based classification systems in Chapter 3, which includes three operations, namely rule generation, rule simplification and rule representation, in order to fulfil the aim of the research as mentioned in Chapter 1. The significance of the three operations is justified in theoretical context. The comparative validation introduced in Chapter 4 also proves empirically that all of the three operations are significant for design of rule based classification systems in both theoretical and practical aspects. This is because the results show that effective rule simplification really helps improve both predictive accuracy and computational efficiency in training stage and that effective rule representation really helps improve computational efficiency in testing stage. As mentioned in Chapter 1, there are three objectives that need to be achieved towards the fulfilment of the research aim. The rest of this section introduces the extent to which each of the three objectives is achieved.

With regard to objective 1, this thesis introduces novel methods and techniques with respects to the three operations mentioned above. The comparative validation proves that the development of the novel methods and techniques brings in methodological impact with respects to the improvement of both predictive accuracy and computational efficiency. In particular, IEBRG is introduced in Chapter 3 with respect to its essence. The motivation of its development is also justified in theoretical analysis using specific examples to show the advantages of the IEBRG method in comparison with Prism. Empirical results also prove that these theoretical advantages usually make IEBRG outperform Prism in the aspects of predictive accuracy and computational efficiency. On the other hand, Chapter 3 also introduces Jmid-pruning as a novel method of rule simplification with respect to its essence. The advantages of the method are theoretically analysed and thus motivate the development of the method towards the improvement of accuracy and efficiency in comparison with J-pruning and Jmax-pruning. The empirical results also prove that these advantages usually make Jmid-pruning outperform the other two existing pruning methods. Furthermore, rule based network is introduced in Chapter 3 as a novel technique of rule representation. The importance of this representation is justified by highlighting its advantages in terms of graphical complexity and theoretically validated in terms of time complexity. The theoretical justification and validation indicate that rule based network performs a higher level of efficiency than decision tree and linear list. Therefore, the above descriptions indicate that the first objective of this research is successfully achieved.

With regard to objective 2, this thesis introduces two advanced frameworks of ensemble learning for classification in Chapter 3. The first one of the frameworks is referred to as Collaborative and Competitive Random Decision Rules (CCRDR) as mentioned in Chapter

3. The motivation of creating the advanced framework is soundly justified by highlighting its advantages that are capable of filling the gaps that exist in existing approaches such as Random Forests and Random Prisms. The empirical validation also proves the relevance and importance of the CCRDR with respect to the compliments made to Random Prism. In other words, the advantages of CCRDR, which includes the incorporation of multiple learning algorithms and competitive learning and the way of determining the weight of a classifier for weighted majority voting, usually make the ensemble learning approach generate hypothesis with high robustness and reliability towards the improvement of overall classification accuracy. In particular, Chapter 3 introduces the way to combine different learning algorithms for construction of strong learners on training data. The empirical results shown in Chapter 4 also proves that the combination of multiple learning algorithms usually make the constructed leaners stronger in comparison with use of a single algorithm. On the other hand, a new way of determining the weight of a single classifier for weighted majority voting is introduced in Chapter 3. The empirical results shown in Chapter 4 also proves that the new way that precision is used to measure the weight usually improves the reliability of a single classifier in making a particular classification in comparison with use of overall accuracy or recall which is used or suggested in (Stahl & Bramer, 2013; Stahl & Bramer, 2011). On the other hand, this thesis also introduces another novel framework of ensemble learning on the basis of the former framework. For the latter framework, the motivation is also soundly justified by highlighting its advantages in comparison with the collaboration strategy applied to CCRDR. The empirical validation also proves that collaboration and competition involved per rule generated would achieves a higher quality of model than that involved per rule set generated. Therefore, the above descriptions indicate that the second objective of this research is successfully achieved.

With regard to objective 3, this thesis introduces the comparative validation including description of data sets, experimental setup and results in Chapter 4. Firstly, all of the chosen data sets are described with respect to the characteristics in general in Chapter 4 and in detail in Appendix VII. The reason why the data sets are chosen is also justified. Secondly, the experimental setup is described in detail and the ways to measure the accuracy and efficiency and to make comparisons are also soundly justified in order to make the results convincing enough. Thirdly, the results are also discussed in quantitative context. The reasons why the novel approaches perform better or worse than the existing ones are also explained by highlighting their advantages and disadvantages. As mentioned earlier, the first two objectives are successfully achieved, which means the results indicate that the research methodology developed in the thesis is scientifically significant. In particular, the research methodology shows theoretical significance, practical importance and methodological impact in scientific context, which are further specified in Chapter 6. Therefore, the above descriptions indicate that the third objective of this research is successfully achieved.

5.3 Theoretical Analysis of Interpretability

As described in Chapter 1, the interpretability of rule based systems is significant due to the presence of some problems from machine learning methods, size of data, model representation and human expertise and preferences. This section discusses how these factors have an influence on interpretability and lists several criteria for evaluation on the interpretability.

5.3.1 Learning Strategy

As mentioned earlier, different machine learning algorithms usually involve different strategies of learning. This would usually result in differences in two aspects namely, transparency and model complexity.

In terms of transparency, a rule based method aims to generate a rule set typically in the form of either a decision tree or if-then rules. As mentioned in Chapter 1, rule based knowledge representation is able to explain the reason explicitly with regard to providing an output and, thus, it is well transparent. This is a significant advantage compared with some other popular machine learning methods such as neural networks and k nearest neighbour. Neural network learning aims to construct a network topology that consists of a number of layers and that has a number of nodes, each of which represents a perceptron. As a neural network is working in a black box manner with regard to providing an output, the transparency is poor, i.e. people cannot see in an explicit way the reason why the output is given. On the basis of the above description, neural networks have been judged poorly interpretable in (Stahl & Jordanov, 2012). K nearest neighbour involves lazy learning. In other words, the learning algorithm does not aim to learn in depth to gain some pattern from data but just to make as many correct predictions as possible. In the training stage, there is no actual learning but just some data loaded into computer memory. In this sense, there is no model built in the training stage so there is nothing to be visible for people to gain some useful patterns.

In terms of model complexity, as mentioned in Chapter 1, rule based methods can be divided into two categories namely, 'divide and conquer' and 'separate and conquer', due to the difference in their strategies of rule generation. As mentioned in (Fürnkranz, 1999), the latter approach usually generates fewer and more general rules than the former approach. The above phenomenon is due mainly to the strategy of rule learning. As mentioned in Chapter 2, the rule set generated by TDIDT needs to have at least one common attribute to be in the form of decision trees. The same also applies to each of the subtrees of a decision tree, which requires to have at least one common attribute represented as the root of the subtree. Due to this requirement, TDIDT is likely to generate a large number of complex rules with many redundant terms such as the replicated subtree problem illustrated in Chapter 2 and thus results in a model of high complexity. On the other hand, as mentioned above, k nearest neighbour does not build a model in the training stage. From this point of view, the model complexity is 0 as there is no model built.

On the basis of the above description relating to transparency and complexity, the strategies of learning involved in learning algorithms are an impact factor that affects interpretability.

5.3.2 Data Size

As mentioned in Section 5.3.1, different learning algorithms involve different strategies of learning and thus generate models with different levels of complexity. In this sense, when the same data set is used, different learning algorithms would usually lead to different model complexity. However, for the same algorithm, data in different size would also usually result in the generation of models with different levels of complexity. The rest of this subsection justifies the potential correlation between data size and model complexity using rule based methods as examples.

As mentioned earlier, rule based methods involve the generation of rule sets. The complexity of a rule set is determined by the total number of rule terms, which is dependent upon the number of rules and the average number of terms per rule. However, the total number of rule terms is also affected by the data size in terms of both dimensionality (number of attributes) and sample size (number of instances). For example, a data set has n attributes, each of which has t values, and its sample contains m instances and covers all possible values for each of the attributes. In this example, the model complexity would be equal to $\Sigma\, t^{i}$, while $i=0, 1, 2 \ldots n$, but no greater than $m \times n$ in the worst case. This indicates that a rule set consists of a default rule, which is also referred to as 'else' rule, and t^{i} rules, each of which has i terms, for $i=0, 1, 2 \ldots n$ respectively. However, each rule usually covers more than one instance and the rule set is expected to cover all instances. Therefore, the number of rules from a rule set is usually less than the number of instances from a data set. As also justified above, each rule would have up to n (the number of attributes) terms due to the requirement that each attribute can only appear once comprising one of its possible values in any of the rules.

On the basis of above description, the complexity of a rule set is up to the product of dimensionality and sample size of a data set.

5.3.3 Model Representation

As mentioned in Chapter 2, different types of machine learning algorithms may generate models represented in different forms. For example, the 'divide and conquer' approach generates a rule set in the form of a decision tree as illustrated in Fig.1 whereas the 'separate and conquer' approach would generate if-then rules represented in a linear list. In addition, a neural network learning algorithm would generate a multi-layer network with a number of interconnected nodes, each of which represents a perceptron. As described in Section 5.3.1, models generated by rule based learning methods are in white box and thus well transparent whereas models constructed by neural network learning methods are in black box and thus poorly transparent. As justified in Section 5.3.1, the level of transparency can affect the level of interpretability. However, models that demonstrate the same level of transparency may also have different levels of interpretability due to their differences in terms of representation. The rest of this subsection justifies why and how the nature of model representation can affect the level of interpretability of rule based models.

As argued in Chapter 2, decision trees suffer from the replicated subtree problem and thus are often difficult for people to read and understand to gain knowledge. In contrast to decision trees, linear lists do not have the constraint that all rules must have common attributes and thus reduces the presence of redundant terms in a rule set. However, redundancy may still arise with this representation. This is because the same attribute may repetitively appear in different rules as illustrated by the example below:

Rule 1: If x=0 and y=0 Then class= 0;
Rule 2: If x=0 and y=1 Then class= 1;
Rule 3: If x=1 and y=0 Then class= 1;
Rule 2: If x=1 and y=1 Then class= 0;

When a training set is large, there would be a large number of complex rules generated. In this case, the presence of redundancy would make the rule set (represented in a linear list) become very cumbersome and difficult to interpret for knowledge usage. In other words, a large number of complex rules represented in this way is quite like a large number of long

paragraphs in an article that would be very difficult for people to read and understand. Instead, people prefer to look at diagrams to gain information. In this sense, graphical representation of rules would be expected to improve the interpretability of a model. More details about the improvement are presented in Chapter 6.

5.3.4 Level of Expertise and Preferences

As mentioned in Chapter 1, different people may have different levels of expertise and preferences and thus different levels of cognitive capability to understand the knowledge extracted from a particular rule based system. The rest of this subsection justifies why and how human expertise and characteristics may affect the interpretability of rule based systems.

In terms of expertise, due to the fact that an expert system is used to act as a domain expert to extract knowledge or make predictions, people need to have the relevant expertise in order to be able to understand the context. From this point of view, the exactly same model may demonstrate different levels of interpretability for different people due to their different levels of expertise in this domain.

In terms of preferences, due to the fact that different people may have different preferences with respect to the way of reading, the exactly same model may also demonstrate different levels of interpretability for different people due to their different preferences. From this point of view, human characteristics is also an impact factor that may affect the interpretability of model.

As mentioned in Section 5.3.3, model representation can affect the interpretability with respect to level of redundancy. In other words, the same model can have different levels of interpretability depending on its representation. However, due to the difference in expertise and preferences, a particular representation may be understandable to some people but not to others. For example, some people in nature science/engineering would prefer to read diagrams/ mathematical formulas whereas others may dislike them. From this point of view, model representation, human expertise and characteristics may jointly determine the cognitive capability for people to read and understand the knowledge extracted from a model.

5.3.5 Criteria for Evaluation of Interpretability

On the basis of above description in this section, the list of the identified impact factors would have the causal relationship to the interpretability as illustrated in Fig 5.1.

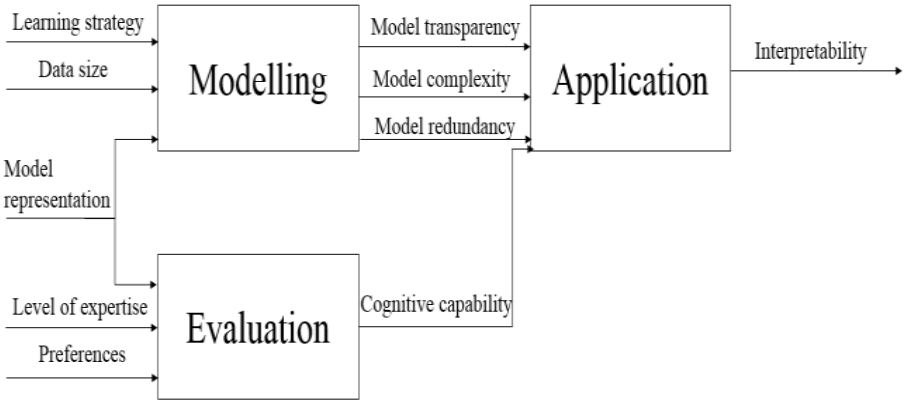

Fig.5.1 Causal relationship between impact factors and interpretability

Fig.5.1 indicates that the evaluation on interpretability could be based several criteria, namely model transparency, model complexity, model redundancy and cognitive capability, due to their direct relationships to interpretability.

In terms of model transparency, as mentioned in Section 5.3.1, the evaluation is based on information visualization. In other words, in what percentage the information is visible or hidden to people. For example, a neural network learning method generates a model in black box, which means that the information in the model is mostly hidden to people and thus poorly transparent. In contrast, a rule based learning method generates a model in white box, which means the information in the model is totally visible to people and thus well transparent.

In terms of model complexity, the evaluation is subject to the type of learning algorithms to some extent. In particular, with regard to rule based methods, the model complexity could be measured by checking the total number of rule terms in a rule set, which is referred to as rule set complexity. For the rule set given below, the complexity would be 8.

Rule 1: If x=0 and y=0 Then class= 1;
Rule 2: If x=0 and y=1 Then class= 0;
Rule 3: If x=1 and y=0 Then class= 0;
Rule 2: If x=1 and y=1 Then class= 1;

In terms of model redundancy, the evaluation could be based on the extent of information duplication. In particular, a rule set may be represented in different forms namely, decision tree, linear list and rule based network. As mentioned in Section 5.3.3, the first two representations both may include duplicated information. For decision tree, the replicated subtree problem is a typical example to indicate that redundancy is a principal problem that arises with the representation. As can be seen from Fig.2.2 in Chapter 2, there are four subtrees identical. For a linear list, as can be seen from the rule set given earlier in this section, all of the four rules have two common attributes, namely ‘x’ and ‘y’, which are repeated. The authors have developed two types of network topologies in order to reduce the redundancy as introduced in Chapter 3. In particular, one is attribute-value-oriented and the other one is attribute oriented. More details on the improvements have been described in Chapter 3.

In terms of cognitive capability, the evaluation would be based on empirical analysis following machine learning approaches. This is in order to analyse the extent to which the model representation is understandable to particular people. In particular, this could be designed as a classification task to predict the cognitive capability in qualitative aspects or as a regression task to predict in quantitative aspects. Briefly speaking, the analysis could be done by collecting the data records on expertise and preferences from previous people who have high similarities to the current people and then taking the majority voting (if designed as a classification task) or averaging (if designed as a regression task) with respect to the cognitive capability. The above task can be done effectively using k nearest neighbour algorithm.

5.4 Research Contributions

This thesis makes main contributions in scientific aspects. In particular, this thesis introduces methods and techniques in rule based classification, namely Information Entropy Based Rule Generation, Jmid-pruning and rule based network representation, as well as two

ensemble learning frameworks known as Collaborative and Competitive Random Decision Rules and Collaborative Rule Generation respectively. With respect to rule based network representation, there is also a unified network topology created to generalize the type of representation. This is in order to make the network topology fulfil being based on any type of logic such as deterministic, probabilistic and fuzzy logic. In addition, the topology is even generalized to fit any other type of computational networks such as neural networks, Bayesian network and digital circuit as introduced in Chapter 6. The contributions mentioned above belong to methodological novelty. This thesis also achieves conceptual novelty, which includes the creation of a unified framework for design of rule based systems and the division of rule based systems into single rule based systems and ensemble rule based systems. Both parts of the conceptual novelty are in the context of system theory, which is about the relationship between systems and components or between super-systems and sub-systems. This would also be seen as a novel application of system theory. Similarly, this thesis also involves some other novel applications. For example, the development of Information Entropy Based Rule Generation method involves using information entropy, which is a popular technique in information theory. In addition, the development of rule based network representation involves using graph theory and the validation of the representation using BigO notation.

On the other hand, this thesis also has some contributions in philosophical aspects. In particular, this thesis introduces novel understanding of data mining and machine learning as well as the difference between the two subjects from cognitive point of view. This thesis also introduces novel understanding of ensemble learning in the context of learning theory and ensemble rule based systems in the context of system theory. In addition, novel applications of UML class and instance diagrams for modelling of knowledge systems are specified and an example that illustrates the research framework of this PhD using the UML diagrams is also specified in the appendix I. This thesis also involves novel applications of mathematical theory and object oriented programming concepts in rule based systems with respects to rules, rule bases and rule sets. Finally, this thesis introduces how the research methodology introduced in Chapter 3 strongly relates to three main theories namely information theory, system theory and control theory. More details on the contributions highlighted above are presented in Chapter 6.

5.5 Conclusion

This chapter evaluates the completed work against the research objectives listed in Chapter 1 in qualitative context. It clarifies the extent to which the objectives are achieved and justifies why the research methodology is successfully developed and significant in scientific aspects. The contributions of the thesis are also summarised in scientific and philosophical aspects in this chapter. The advantages of the research work are further specified in Chapter 6 with respects to theoretical significance, practical importance, methodological impact and philosophical aspects. However, there is still space for further improvement in the specified research area in spite of that the research methodology introduced in the thesis shows significant advantages. Therefore, further directions of this research area are also specified further in Chapter 6 towards improvement of research methodologies in the area.

Chapter 6 Conclusion

6.1 Theoretical Significance

The aim of the thesis is to introduce a theoretical unified framework for design of rule based classification systems as mentioned in Chapter 1. The framework includes three main operations namely, rule generation, rule simplification and rule representation as mentioned in Chapter 3. The experimental results shown in Chapter 4 prove that the incorporation of rule simplification is relevant and can generally make rule generation more accurate and efficient. In addition, the theoretical results on rule representation has shown that different structures of a rule set can lead to different levels of efficiency in prediction stage. This indicates that it is very necessary to represent a rule set in a suitable structure.

This thesis also introduces an advanced framework of ensemble learning for design of ensemble rule based classification systems in Chapter 3. In this framework, competitive learning is incorporated in training stage in order to find better classifiers and ignore worse classifiers for prediction stage. From theoretical point of view, this incorporation can improve the flexibility of the design framework. This is because different learning algorithms may have different levels of fitness to a particular data set due to its characteristics. In addition, a new strategy in determining the weight for weighted majority voting, which is based on precision for each individual classification, has been introduced in the thesis to increase the reliability of a classifier for making a particular classification. This is because of the fact that the overall accuracy cannot well represent the capability of a classifier in making predictions on instances of a particular classification, especially on extremely unbalanced data sets. The empirical results shown in Chapter 4 indicate that the two above modifications to the ensemble learning framework can improve the overall classification accuracy.

The thesis also brings in a novel application of graph theory (Biggs, Lloyd, & Wilson, 1986; Bondy & Murty, 2008) in Chapter 3 for the development of networked rule representation as well as a novel application of BigO notation (Cormen , Leiserson , Rivest , & Stein, 2001) in Chapter 4 for the theoretical validation of the representation with respect to computational efficiency in prediction stage. Both concepts are widely used in discrete mathematics (Johnsonbaugh, 2008) and complexity theory (Gegov, 2007).

The theoretical framework for designing rule based classification systems introduced in Chapter 3 can also be illustrated in the context of system theory in addition to machine learning. In other words, in the past research, a rule based system is conceptually referred to as a special type of expert system but the design of such systems mainly follows traditional engineering approaches rather than machine learning approaches. Although there is a special type of classification referred to as rule based classification in machine learning, it is usually in the context of rules or rule sets rather than system theory. In this thesis, the conceptual connection is made between rule based systems and machine learning as follows. In machine learning context, the objective of rule based classification is the induction of rule sets. Therefore, a rule based classifier is usually referred to as a rule set. In system theory context, the classifier can also be regarded as a rule based classification system. On the other hand, if the generation of classification rules is done by an ensemble learning approach, there would be multiple classifiers constructed as an ensemble learner. In system theory context, this can be seen as design of an ensemble rule based classification system which has each single

classifier as a subsystem of its. Therefore, the above descriptions bring new concepts to system engineering (Schlager, 1956; Sage, 1992) with respect to methodological differences in comparison with existing methodologies (Hall A. D., 1962; Goode & Machol, 1957).

6.2 Practical Importance

The theoretical framework introduced in the thesis is generally domain independent in real applications because almost all domains usually follow similar approaches in problem solving in machine learning context. This is a significant difference to expert based approaches, which are generally domain dependent and need to have expert knowledge acquired as the main task in the process of designing a rule based system (Aksoy, 2008; Liu & White, 1991; Hart, 1989; Mrozek, 1992). The framework can contribute to the development of expert systems for the purpose of knowledge discovery and predictive modelling such as medical applications as reported in (Quinlan, 1987; Quinlan, 1988; Michalski, Mozetic, Hong, & Lavrac, 1986).

In the aspect of knowledge discovery, rule based systems can be used by domain experts to find interesting, useful and previously unknown patterns such as causal relationships. This can help the experts further identify new research directions as well as make necessary validations on their hypothesis by using real data. In classification, as mentioned in Chapter 3, one purpose of rule representation is to present the knowledge in different ways in accordance with specific commercial requirements. From this point of view, the networked representation can effectively reflect the importance of input attributes and provide a ranking of the attributes according to their importance. In practice, each input attribute can be seen as an impact factor to which a decision outcome is subject. And the ranking of attributes can help domain experts identify which ones are more important, less important or irrelevant to the decision outcome.

In the aspect of predictive modelling, rule based systems can be used to help with prediction such as recommendation, decision making and stock price prediction. In classification tasks, as mentioned in Chapter 3, it can help make categorical predictions in qualitative aspects on a single variable such as weather prediction, degree classification and faults classification. In this context, each classifier constructed in training stage actually acts as a decision maker to make decisions/predictions. If the ensemble learning approach is adopted, it means that multiple classifiers constructed in training stage act as a group of decision makers to achieve collaborative decision making. In addition, as part of the design framework for rule based classification systems, rule simplification can help speed up the process of modelling and rule representation can help make quicker decisions/predictions.

Besides, as mentioned in Chapter 2, ensemble learning could be done in parallel, which means that each single machine learner is constructed independently and that only their predictions are combined for final decision making. This indicates that the ensemble learning could be done by a parallel computer to improve the computational efficiency in both training and testing stages. In addition, each company or organization may have branches in different cities or countries so the databases for the companies or organizations are actually distributed over the world. As the existence of high performance cloud and mobile computing technologies, the ensemble learning framework can be easily transplant into distributed or mobile computing environments such as multi-agent systems (Wooldridge, 2002).

The rule based systems described in this thesis are based on Boolean logic. In other words, this kind of systems is usually referred to as deterministic rule based systems which make decisions under certainty in practice. However, rule based systems can also be designed based on probabilistic logic or fuzzy logic in real applications for decision making under probabilistic or non-probabilistic uncertainty. In this context, the theoretical framework introduced in the thesis can also be extended for design of probabilistic and fuzzy rule based systems when the two kinds of rule based systems are required in practice. In addition, the thesis also focuses on rule based systems for classification such as qualitative diagnostics. However, this kind of systems can also be used for regression and association in machine learning context so the theoretical framework can also be extended for design of rule based regression/ association systems for other practical purposes.

6.3 Methodological Impact

This thesis introduces some novel methods and techniques in rule based classification and ensemble learning in Chapter 3 namely information entropy based rule generation (IEBRG), Jmid-pruning, rule based networks, collaborative and competitive decision rules and collaborative rule generation.

IEBRG is developed by making modifications to Prism algorithm in order to overcome the limitations of Prism in the aspects of clashes, underfitting and computational efficiency as mentioned in Chapter 3. IEBRG represents the so-called 'from causes to effects approach' for rule generation which has been proven empirically more effective and efficient than the so-called 'from effects to causes approach' represented by Prism in Chapter 4. For example, in comparison with the latter approach, the former approach can significantly reduce the number of discarded rules due to clash handling and thus reduce the unnecessary computational costs as well as make the machine learning tasks more effective and efficient. In most cases, IEBRG outperforms Prism in both classification accuracy and computational efficiency on noise-free data sets but also show equally competitive comparing with Prism on noise data set.

Jmid-pruning is an extension of Jmax-pruning for rule simplification as mentioned in Chapter 3. In this context, the major modifications made to Jmax-pruning are with regard to improvement of computational efficiency. As mentioned in Chapter 3, for both rules and rule terms, if they are discarded after they have been generated, this means that the computation is irrelevant in data mining context and that the learning is not effective in machine learning context. The empirical results in Chapter 4 show that Jmid-pruning usually generates fewer discarded rules or rule terms and thus needs a smaller number of iterations for rule generation. The results indicate that Jmid-pruning achieved the improvement in the context of both data mining and machine learning. In addition, the Jmid-pruning development also involves modifications to Jmax-pruning with regard to improvement of classification accuracy for a special case as mentioned in Chapter 3. In the special case, Jmid-pruning is developed towards reducing the number of discarded rules and thus avoiding underfitting. It is proven empirically in Chapter 4 that Jmid-pruning usually keeps more rules and thus avoids loss of accuracy in the special case. It indicates that Jmid-pruning also makes rule sets generated more robust.

Rule based network is a networked representation of rules or rule sets, which means a rule based system is represented in a networked structure. In this context, a special type of rule based systems can be referred to as rule based networks if the rule based systems are in the

form of networks. In addition, in the context of complexity theory, it is proven theoretically in Chapter 4 that the networked rule representation is computationally more efficient than decision tree and linear list representations in term of time complexity for predicting unseen instances. In the context of knowledge representation, rule based network can also provide a better interpretability to reflect the relationships between inputs and outputs as well as the importance of input attributes as justified in Chapter 3.

Collaborative and competitive random classification rules is an advanced framework of ensemble learning, which involves modifications made to Bagging based approaches such as Random Prism as mentioned in Chapter 3. The incorporation of competitive learning can bring flexibilities into training stage in order to find as better fitness as possible between chosen learning algorithms and data samples. In other words, each algorithm may have different levels of fitness to different samples of data. Also, different algorithms may perform different levels of accuracy on the same sample. In this context, competitive learning is useful to find the best potential candidate on each particular sample. In addition, modification in weighted majority voting is relevant to measure more accurately the reliability of a classifier in making a particular decision/prediction. The empirical results shown in Chapter 4 prove that the above modifications usually improve the overall classification accuracy. In addition, the creation of another framework of ensemble learning called Collaborative Rule Generation also helps improve the quality of each rule generated by learning from training set.

In comparison with other popular machine learning methods such as neural networks, support vector machine and k nearest neighbour, rule based methods have significant advantages in terms of model transparency and depth of learning.

In terms of model transparency, neural network is in black box and thus poorly transparent. This would usually make general audiences difficult to understand the principles of making predictions. In data mining tasks, general audiences would only know the mappings between problems (inputs) and solutions (outputs) but not be aware of the reasons. In data mining tasks, it is more important to find the reasons between problems and solutions for knowledge discovery. Support vector machine is not in black box but still less transparent to general audiences. This is because the model built by using this algorithm is function lines as decision boundaries in geometric form or a piecewise function in algebraic form. This type of model representation would usually be less interpretable to non-technical audience who doesn't know mathematics well. K nearest neighbour does not aim to build a model but just to memorize all data instances in the training stage. Therefore, it is less transparent for what it has learned due to the absence of transformation from data to information/knowledge. In other words, audience would usually not be interested in pure data but the pattern that is hidden inside the data. In contrast to the three methods above, rule based methods are in white box as the models built by using this type of methods are in the form of rules. Most audiences would easily understand the logical relationships between causes (inputs) and effects (outputs). Therefore, such models are highly interpretable so that it could be clearly known by general audiences that in what way the predictions are made. This advantage would usually make rule based methods more popular than other machine learning methods for data mining tasks to extract the knowledge discovered from data.

In terms of depth of learning, neural network does not involve a sound theory in learning strategy. In contrast, it practically starts from random design of the network with regards to

the neurons, connections and weights for inputs and then makes corrections to the network topology through experience. It indicates that neural network does not have a clear learning outcome. Support vector machine involves a sound theory in learning strategy but the learning by this method is not in depth. This is because this method does not go through all data instances but just take a look at a few instances that are selected as support vectors. K nearest neighbour just involves a simple memorized learning in training stage as mentioned earlier. Therefore, the depth of learning using this method is quite insufficient. On the basis of above descriptions on support vector machine and k nearest neighbour, both methods are seen as lazy learning. The type of learning would just aim to identify the way to solve a specific problem rather than to pay attention to the essence of the solution in depth. In contrast to the three above methods, rule based methods would aim to discover the causal relationships between problems and solutions by going through the whole data sets and to represent the causal relationships in the form of rules. In this way, the method does not only find the way to solve a problem but also gets aware of the essence of the solution in depth.

Besides, rule based methods are capable of dealing with both discrete and continuous attributes. In contrast, neural network and support vector machine are less effective in dealing with discrete attributes. K nearest neighbour is capable of dealing with ordinary attributes according to the rank of values such as ‘very large’, ‘large’, ‘medium’, ‘small’ and ‘very small’. However, it is still less effective to deal with other types of discrete attributes for this method. Overall, the above description indicates that rule based methods would be useful and popular in both data mining and machine learning tasks due to their good transparency and depth of learning.

6.4 Philosophical Aspects

This thesis mainly focuses on scientific concepts. However, the concepts also have some philosophical aspects.

One of the aspects is about the understanding of data mining and machine learning as well as the difference between them from conceptual point of view.

There are continuously some criticises that a machine is neither able to learn nor to get beyond people scientifically. The argument is that machines are invented by people and the performance and actions of the machines are totally dependent on the design and implementation by engineers and programmers. It is true that machine is controlled by program in executing instructions. However, what if the program is about the implementation of a learning method? The answer would be obviously that the machine executes the program to learn something. On the other hand, if a machine is thought to be never superior to people, it would be equivalent to imply in human learning that a student would never be superior to his/her teacher. It is not really true especially if the student has the strong capability to learn independently without being taught by teachers. Therefore, it would also be valid in machine learning if the machine holds a good learning method. On the other hand, machine learning needs to be given a definition. This is because there is still not a unified definition of machine learning till today.

Langley (1995) defined that "*Machine learning is a science of the artificial. The field's main objects of study are artifacts, specifically algorithms that improve their performance with experience*."

Tom Mitchell (1997) defined that "*Machine Learning is the study of computer algorithms that improve automatically through experience*."

Alpaydin (2004) defined that "*Machine learning is programming computers to optimize a performance criterion using example data or past experience*."

All of the three definitions would not be sufficient. The first one points out the field and objects of the study. The second one mentions the expected outcome of the research in the context of computer algorithms. The third one specifies the way to improve the performance of computer programs. However, none of them makes a strong relationship to 'learning theory'. Generally speaking, machine learning would be inspired by human learning in order to simulate the process of learning in computer software. In other words, the name of machine learning would tell people that machine is capable of learning. Therefore, the definition of machine learning would be in relation to learning methods, learning outcome and depth of learning.

In connection to data mining, there is also some incorrect cognition. Firstly, people think that data mining is an application of machine learning. It is not really true. Machine learning methods are usually just involved in the key stage for knowledge discovery in data mining tasks. In other words, it is required to do data collection and pre-processing prior to knowledge discovery in data mining tasks. For knowledge discovery, it does not have to adopt machine learning methods. In principle, it could be done by experts manually walking through data or other statistical methods without actual learning. However, data mining is defined as a branch of artificial intelligence. Therefore, machine learning would be obviously one of the most popular approaches. On the other hand, data mining also involves some other tasks that are not done by machine learning techniques. For example, data mining needs to pre-process data such as feature selection/extraction and sampling by using statistical methods. In the past research, these types of methods are misclassified to machine learning methods. Strictly speaking, it is not really true because there is no learning done when these methods are used. For example, Principle Component Analysis (PCA) is just a statistical method to calculate the eigenvalues for a feature set and to make judgement on principle components and their rankings according to their corresponding eigenvalues. These methods can be defined as tools to support learning methods in machine learning tasks. In other words, machine learning also needs methods relating to data pre-processing to improve the performance of learning algorithms. On the basis of above description, machine learning strongly overlaps with data mining in scientific research.

In scientific aspects, data mining and machine learning incorporate almost the same theoretical concepts and most methods, such as decision trees, Naive Bayes and k nearest neighbour, belong to both areas. Therefore, it seems that there is no obvious difference between data mining and machine learning from this point of view. However, the two subjects actually have very different practical purposes. Data mining is aimed at knowledge discovery, which means it is working in white box so that the pattern discovered can be visible to people and will be further used for knowledge or information. In contrast, machine learning is aimed at predictive modelling, which means it is working in black box and the model actually represents the knowledge learned from data but is only used further to help make predictions. From this point of view, people are not interested in the model contents but only in the model outputs. This is very similar to that students do not pay attention to principles of knowledge in depth but just want to know how to apply the knowledge they

learned. On the basis of above descriptions, data mining and machine learning thus have different significance in validation. In particular, data mining is generally processing large volume of data in order to discover as correct pattern as possible. From this point, the efficiency in training stage is critical as it can reflect if the chosen algorithm is computationally feasible in practice. The efficiency in testing stage is not critical as the testing aims to check the reliability of the model for further use as knowledge or information. However, machine learning may generally process relatively small data in real applications. Therefore, the efficiency is not critical in training stage but is in testing stage. This is because training could be done offline and aims to learn knowledge from data but testing aims not only to check how accurately a prediction is made by the model but also how quickly the prediction is made due to the fact that prediction needs to be not only right but also quick. On the other hand, with regard to accuracy, data mining aims to measure the extent to which the model can be trusted if it is further used as knowledge. In contrast, machine learning aims to measure how accurately the model can make predictions.

From another point of view, the difference between data mining and machine learning is also like the difference between human research and learning. As mentioned above, the purpose of data mining is for knowledge discovery. In other words, data mining acts as a researcher in order to discover something new which is previously unknown. Therefore, it is like research tasks and thus significant for researchers to guarantee the correctness of pattern discovered from data. Machine learning is obviously like human learning tasks. In other words, machine learning acts as a learner/student to learn something new which is previously known. To what extent the learning outcomes are achieved is typically measured by assessments such as examination or coursework. From this point of view, it is more important to achieve a high level of predictive accuracy on unseen instances in comparison with the correctness of the model built. This is because predictive accuracy is like marks awarded from assessments whereas model correctness is like the correctness of knowledge learned. In fact, it is possible that students do not understand principle of knowledge in depth but can correctly answer exam questions to gain marks. Similarly, the model built by a machine learning algorithm may have bias and defects but can correctly make predictions. For example, some strategies in machine learning, such as conflict resolutions and assigning a default classification mentioned in Chapter 2, are like some examination skills in human learning. From this point of view, it indicates that not all of machine learning methods could be well used in data mining tasks. This is just like that not all of the learning methods could be evolved to a research method. In human learning, learning methods could be classified according to education levels such as fundamental education, higher education and training education. Generally speaking, probably only the learning methods applied in higher education are more likely to be evolved to research methods. This is because this type of methods could usually better help learners develop the understanding of principles in depth. For example, a learning method may help students gain skills for practical applications but not develop understanding of principles in depth. This would usually result in the case that student can well apply what they learned in depth but cannot make other people understand what they learned. It is equivalent to that a model built by using a machine learning method has a good predictive accuracy but is poorly interpretable. Therefore, some machine learning methods that do not aim to learn in depth would not become good data mining methods.

The second philosophical aspect is on the understanding of ensemble learning in the context of learning theory. As mentioned in Chapter 2, ensemble learning can be done in parallel or

sequentially. In the former way, there are no collaborations among different learning algorithms in the training stage and only their predictions are combined in the testing stage. In academic learning theory, this is like team working, which means students learn knowledge independently and only work on group works together using their knowledge. Their ways of making collaborations in the works are just like the strategies in making final predictions in ensemble learning. In another way of ensemble learning, there are collaborations involved in training stage in the way that the first algorithm aims to learn models and then the latter one learns to correct the models etc. In academic learning theory, this is like group learning with interactions among students in order to improve the learning skills and to gain knowledge more effectively.

The third philosophical aspect is on the understanding of ensemble rule based systems in the context of system theory (Stichweh, 2011). As mentioned earlier, an ensemble rule based system consists of a group of single rule based systems in general, each of which is a subsystem of the ensemble system. In other words, it is a system of systems like a set of sets in set theory (Jech, 2003). In addition, an ensemble rule based system can also be a subsystem of another ensemble system in theory. In other words, a super ensemble rule based system contains a number of clusters, each of which represents a subsystem that consists of a group of single rule based systems.

The fourth philosophical aspect is on the understanding of rule based systems in the context of discrete mathematics such as mathematical logic and relations. With respect to mathematical logic, rule based systems theory has connections to conjunction, disjunction and implication. In machine learning, each rule in a rule set has disjunctive connections to the others. In addition, each rule consists of a number of rule terms, each of which typically has conjunctive connections to the others. In rule based systems, each rule is typically in the form of if-then. In this context, it can represent an implication from the left hand side (if part) to the right hand side (then part) for logical reasoning. With respect to relations, rule based systems can reflect a functional mapping relationship between input space and output space. In other words, the if-then rules in the system must not reflect any one-to-many relationships, which means the same inputs must not be mapped to different outputs as same as the restriction in functions. In rule based systems, this is usually referred to as consistency.

The fifth philosophical aspect is on the novel applications of UML class diagrams (Avison & Fitzgerald, 2006) for modelling of knowledge frameworks as illustrated in Appendix I. The basic idea is very similar to modelling of information systems. In general, a UML class diagram supports to represent four types of relationship between different classes, namely association, inheritance, aggregation and composition (Avison & Fitzgerald, 2006). In a knowledge framework, there are some crossed areas such as the connection between rule based systems and machine learning, which can be represented by association to show that rule based system relates to machine learning. In addition, as mentioned earlier, the framework for design of rule based systems consists of three operations, namely rule generation, rule simplification and rule representation. This could be represented using aggregation/ composition to indicate that the three operations mentioned above are defined as three subclasses of the framework which acts as the superclass. Furthermore, both single rule based systems and ensemble rule based systems could be generalised to be referred to as rule based systems. In a UML class diagram, this could be represented by inheritance to show that the rule based system, which is defined as the superclass, has two subclasses named above. On the other hand, UML instance diagrams (Avison & Fitzgerald, 2006) are

also useful with regard to modelling of the knowledge framework. For example, Prism and IEBRG are the methods of rule generation, which can be represented by two instance diagrams to indicate that the two methods are defined as two instances of rule generation. Similarly, it is also suitable for rule simplification and representation to show that J-pruning, Jmax-pruning and Jmid-pruning are three instances of the former and that decision tree, linear list and rule based network are instances of the latter. Therefore, the basic idea of modelling information systems by UML class and instance diagrams is also applicable to modelling knowledge frameworks from this philosophical point of view.

The sixth philosophical aspect is on the novel application of mathematical theory and object oriented programming concepts. As introduced in Chapter 1, rule base is used to manage rules that have common attributes for both inputs and outputs. Rule base can be seen as a component of a rule set.

In connection to functions as part of mathematical theory, a rule base can be seen as an abstract function, denoted as $f(x_1, x_2, \ldots, x_n)$, without a specific expression. In this context, a rule can be seen as a specific function with a specific expression and domain constrained for its inputs such as the notation below:

$$f(x_1, x_2) = \begin{cases} 1, if \mathrm{x1} = 1 \ \wedge \mathrm{x2} = 1 \\ 0, if x1 = 0 \ \vee x2 = 0 \end{cases}$$

In the notation above, there are two rules: if x_1=1 and x_2=1 then $f(x_1, x_2) = 1$; if x_1=0 and x_2=0 then $f(x_1, x_2) = 0$. In other words, each of rules in a rule base is corresponding to a branch of the function that is corresponded from the rule base.

In connection to object oriented programming concepts, a rule set can be seen as a subclass of abstract rule based systems. This is because a rule based system consists of a set of rules as mentioned in Chapter 1. In this context, a rule based system can be defined as a class and a rule set as an object of the system in the concept of object oriented programming. As it is unknown with respect to what rules a rule set consists of, the class that is defined to represent a rule based system would be abstract, which relates to abstraction as a part of object oriented techniques. As mentioned above, a rule base can be seen as an abstract function. It is actually corresponding to an abstract method in object oriented programming. A rule set consists of a number of rule bases which would have different input and output attributes. It is corresponding to another object oriented technique known as polymorphism. This is because it is achievable that different functions (methods) have the same name but different input parameters and types of return values. Therefore, rule bases in a rule set could be seen as abstract (functions) methods, which have the same name but different input parameters and types of return values, in a class defined for a type of rule based systems to which the rule set belongs. In addition, each of rules in a rule base is corresponding to a branch of if-then-else statement.

In practice, when a training set is given, an abstract class is defined for rule based systems with a number of rule bases. This is because all of possible rule bases could be derived from attributes information of the dataset. Each of the rule bases is defined by an abstract function (method). For each abstract function, its input parameters and type of the return value are specified according to the input and output attributes related to the corresponding rule base. Once a particular rule based learning algorithm is chosen, a subclass of the abstract class, which is corresponding to a rule set generated using this algorithm, is created. All abstract

functions, each of which represents a rule base, are override and overloaded in the subclass. This indicates that each of rule bases is filled by rules if the rules belong to the rule base or defined as null if none of the generated rules fits the rule base. In programming, it is equivalent to implement a function which is originally abstract by providing a specific program statement or leaving the body of the function blank. Once a test instance is given, an object of the subclass is specified to call the functions, each of which is corresponding to a rule base, in order to make predictions.

The last philosophical aspect is on the relationship of the research methodology to three main theories namely information theory, system theory and control theory. From philosophical point of view, the three main theories mentioned above could be understood by the following context:

Information theory generally means passing information from one property to another one. In the process of information passing, it actually happens to have interactions between the two properties. This could be seen as a relationship to system theory. In other words, the two properties are supposed to be two components of a system. However, it is necessary to ensure that the information passing is effective and efficient with high quality. This is because in the process of information passing there may be noise that is present and interferes the transmission. In addition, there may be some information that is confidential to any third parties. In this case, the information usually needs to be encrypted on senders' side and then decrypted on receivers' side. The above description would relate to control theory.

In many other subject areas, the three main theories are also highly significant. A typical example would be in humanities and social science. This world consists of humans, animals, plants and all other non-biological individuals/systems. From this point of view, no one is living alone in the world. Therefore, everyone needs to have interactions with others. This indicates the involvement of system theory to identify the way to interact among individuals/groups. However, the way to achieve interactions would typically be through information passing. The way of passing information could be in many different forms such as oral, written and body languages and some other actions. This brings in control theory in order to effectively control the way of information passing. This is because inappropriate way may result in serious accidents due to misunderstanding of information or unaccepted actions on receivers' side. Therefore, the three main theories would composite an organized entirety in real applications for most types of problem solving.

In this thesis, the research methodology is introduced along all of the three main theories. In particular, the research methodology includes a unified framework for design of single rule based systems. This framework is illustrated by a UML class diagram in Appendix I. As introduced in Chapter 3, this framework consists of three modules namely rule generation, rule simplification and rule representation. This could be seen as an application of system theory. In rule generation, a novel method referred to as IEBRG is based on entropy which is a technique of information theory. In rule simplification, a novel pruning algorithm called Jmid-pruning is based on J-measure which is also an information theoretical technique. On the other hand, both rule simplification and rule representation are incorporated into the framework in order to control machine learning tasks in training stage and testing stage respectively. In particular, rule simplification aims to effectively control the generation of rules towards reduction of overfitting and rule complexity as well as efficiency in training

stage. Rule representation aims to effectively control the process of prediction towards improvement of efficiency in testing stage.

This thesis also has two frameworks of ensemble learning introduced. As introduced in Chapter 2, ensemble learning generally aims to combine different models that are generated by a single or multiple algorithm(s) in order to achieve collaborative predictions. As introduced in Chapter 3, in the two frameworks, there are both collaborations and competitions involved. Multiple algorithms make up an ensemble learning systems and multiple generated rule sets composite an ensemble rule based classifier. Therefore, the creation of the two frameworks involves the application of system theory. However, competitions among classifiers aim to choose the ones of higher quality. The way to measure the quality of each classifier is significant and critical and thus control theory needs to be involved. In addition, in prediction stage, each individual classifier would provide the final prediction maker with a prediction as well as its confidence. It indicates that there is information passing between individuals and thus the application of information theory is also involved in this environment. A unified framework for control of machine learning tasks is proposed as part of future work and introduced along the three main theories in Section 6.5.

6.5 Future Work

As mentioned in Chapter 3, several theoretical frameworks are introduced in the thesis for design of rule based classification systems and ensemble learning. These frameworks can be combined for design of ensemble rule based systems. In this context, the combined framework will further be transformed into another framework referred to as networked rule bases (Gegov, 2010; Gegov, Petrov, & Vatchova, 2010; Gegov, Petrov, Vatchova, & Sanders, 2011). A networked rule base consists of a number of single rule bases as illustrated in Fig.6.1 below.

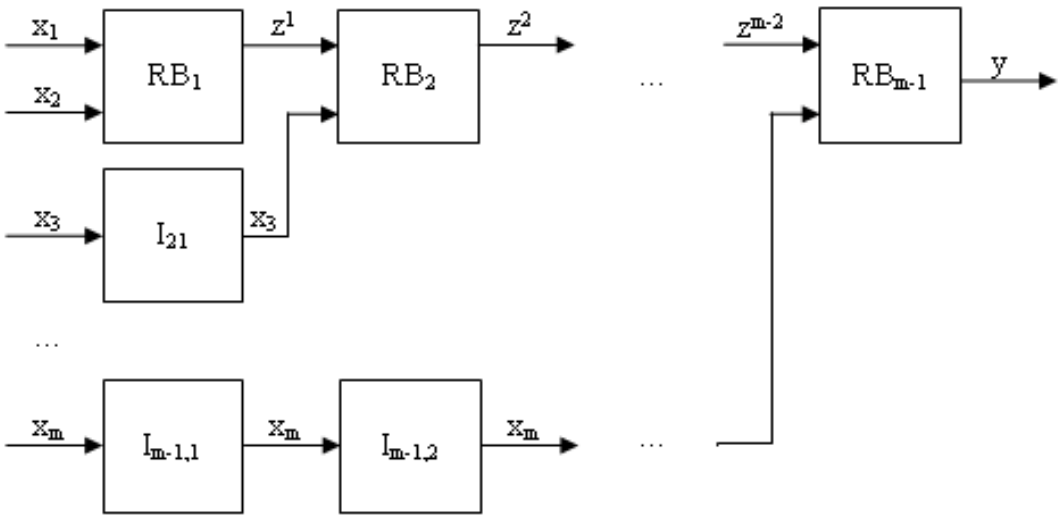

Fig.6.1 Rule Based Network (modular rule bases) (Gegov, 2010)

In this network, each node represents a rule base. The nodes can be connected sequentially or in parallel. In particular, each of the variables labelled x_{m-1}, where *m* represents the number of layer in which the node locates, represents an input and y represents the output. In

addition, each of these labels labelled z^{m-2} represents an intermediate variable, which means this kind of variables are used as outputs for a former rule base and then again as inputs for a latter rule base as illustrated in Fig.6.1. On the other hand, there are two kinds of nodes representing rule bases as illustrated in Fig.6.1, one of which is a type of standard rule bases and labelled RB_{m-1}. This kind of nodes are used to transform the input(s) to output(s). The other type of nodes, in addition to the standard type, represent identities. It can be seen from Fig.6.1 that this type of nodes do not make changes between inputs and outputs. This indicates that the functionality of an identity is just like an email transmission, which means that the inputs are exactly the same as the outputs.

In practice, a complex problem could be subdivided into a number of smaller sub-problems. The sub-problems may need to be solved sequentially in some cases. They can also be solved in parallel in other cases. In connection to machine learning context, each sub-problem could be solved by using a machine learning approach. In other words, the solver to each particular sub-problem could be a single machine learner or an ensemble learner of which a single rule base can make up.

On the other hand, a unified rule based network topology is introduced in Chapter 3. However, this topology can be generalized to fit any type of networks which are used to do computation such as neural networks, Bayesian networks and digital circuits. The topology is illustrated in Fig.6.2 below.

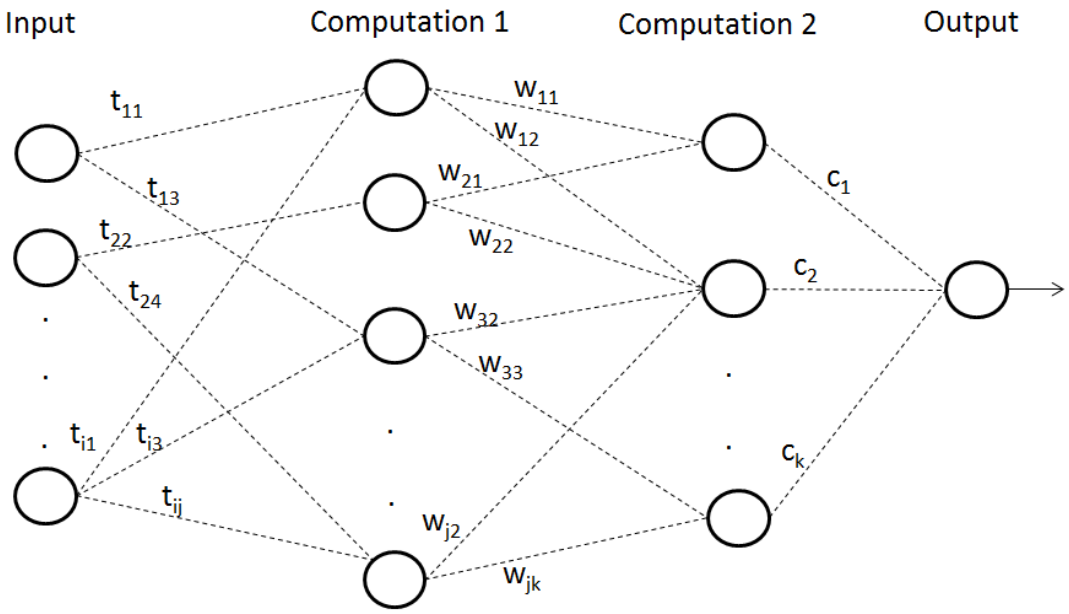

Fig.6.2 Generic Computational Network

In this network, the middle layers represent computation layers, which means that each nodes in this kind of layers represents a special type of computation such as conjunction, disjunction, weighted majority voting, weighted averaging and logical AND, OR and NOT. These operations can also be used in the same network representing a hybrid computational network topology. In such a type of network, there can be either a single computation layer or multiple computation layers as illustrated in Fig.6.2. This is very similar to the neural network topology which could be of either single layer perception or multi-layer perception. Similar to the rule based network topology introduced in Chapter 3 as well as neural network,

each input is assigned a weight when its corresponding value is used for computation. An output from a node in each computation layer is used again as an input with a weight to another node in the latter computation layer if applicable. In practice, this network topology can potentially fulfil the requirement that multiple types of computation must be combined to solve a particular problem.

So far, ensemble learning concepts introduced in machine learning literatures mostly lie in single learning tasks. In other words, all algorithms involved in ensemble learning need to achieve the same learning outcomes in different strategies. This is defined as local learning by the author in the thesis. In this context, the further direction would be definitely to extend the ensemble learning framework to achieve global learning by means of different learning outcomes. The different learning outcomes are actually not independent of each other but have interconnections. For example, the first learning outcome is a prerequisite for achieving the second learning outcome. This direction of extension is towards evolving machine learning approaches in a universal vision. To fulfil this objective, the networked rule bases can actually provide this kind of environments for discovering and resolving problems in a global way. In military process modelling and simulation, each networked rule base can be seen as a chain of commands (chained rule bases) with radio transmissions (identities). In a large scale raid, there may be more than one chain of commands. From this point of view, the networked topology should have more than one networked rule bases parallel to each other. All these networked rule bases should finally connect to a single rule base which represents the centre of command.

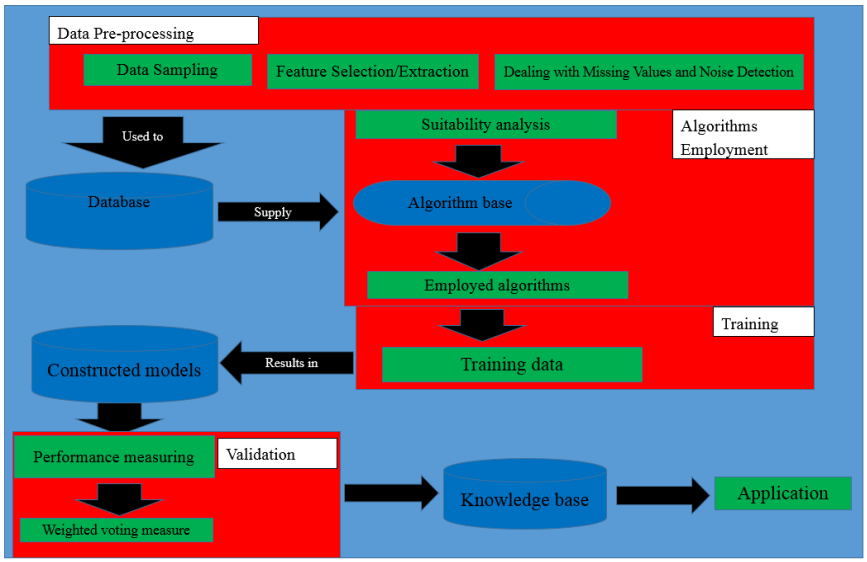

Fig.6.3 Unified Framework for Control of Machine Learning Tasks

Besides, a unified framework for control of machine learning tasks is proposed as illustrated in Fig.6.3. This is in order to effectively control the pre-processing of data and to empirically employ learning algorithms and models generated. As mentioned in Chapter 1, it is also relevant to scale down data in addition to scaling up algorithms for improvement of classification performance. In fact, database is daily updated in real applications, which results in the gradual increase of data size and the changes to pattern existing in the database. In order to avoid the decrease of computational efficiency, the size of sample needs to be determined in an optimal way. As mentioned in (Stahl & Bramer, 2013), the difficulty in

sampling is to determine the size of sample. In addition, it is also required to avoid the loss of accuracy. From this point of view, the sampling is critical not only in the size of sample but also in the representativeness of the sample. Feature selection/extraction is another critical task with regard to pre-processing of data. As mentioned in Chapter 1, high dimensional data would usually results in high computational costs. In addition, it is also very likely to contain irrelevant attributes which result in noise and coincidental pattern. In some cases, it is also necessary to effectively detect noise if the noise is introduced artificially. For example, noise may be introduced in a dataset due to typing errors or illegal modifications from hackers. A potential solution would be using association rules to detect that the value of an attribute is incorrect while the attribute-value pair appears in a data instance together with other attribute-value pairs. Appropriate employment of learning algorithms and models are highly relevant because of the fact that there are many machine learning algorithms existing but no effective ways to determine which ones are suitable to work on a particular data set. Traditionally, the decision is made by experts based on their knowledge and experience. However, it is very difficult to judge the correctness of the decision prior to empirical validation. In real applications, it is not realistic to frequently change decisions after confirming that the chosen algorithms are not suitable.

The above description motivates the creation of the framework for control of machine learning tasks. In other words, this framework aims to use machine learning techniques to control machine learning tasks. In this framework, the employment of both algorithms and models follows machine learning approaches. The suitability of an algorithm and the reliability of a model are measured by statistical analysis on the basis of historical records. In particular, each algorithm in the algorithm base as illustrated in Fig.6.3 is assigned a weight which is based on its performance in previous machine learning tasks. The weight of an algorithm is very similar to the impact factor of a journal which is based on its overall citation rate. In addition, each model generated is also assigned a weight which is based on its performance on latest version of validation data in a database. After the two iterations of employment, a knowledge base is finalised and deployed for applications as illustrated in Fig.6.3.

This unified framework actually includes three main theories involved namely, information theory, system theory and control theory as introduced in Section 6.4. In this framework, there are four modules namely, data pre-processing, algorithms employment, training and validation, and four bases namely, database, algorithm base, model base and knowledge base. The four bases are used to store and manage information in different forms which is in relation to information theory. The four modules are created to control machine learning tasks with respect to decision making in data sampling, use of algorithms and build and validation of models, which relates to control theory. There are also interactions between modules such as passing of chosen data, algorithms and models. What is passing between modules would be a special form of information, which could be seen as a kind of communication and thus relates to information theory. In addition, the interactions between modules would be seen as behaviour of coordination between systems, which relates to system theory.

The unified framework illustrated in Fig.6.3 would provide a Marco vision for research in data mining and machine learning. This would satisfy with real applications of machine learning. This is because in reality machine learning tasks are usually undertaken in complex environments unlike in laboratories. In the latter environment, research is usually undertaken

in a Micro vision and in a pre-processed environment which ignores or eliminates all other impact factors with regard to performance of machine learning tasks. In the future work, the research methodology introduced in Chapter 3 together with other existing approaches would be integrated into the framework for simulation of the control process.

As mentioned in Chapter 1, the main focus of the thesis is on rule based systems for classification. However, rule based systems can also be used for regression (Freedman, 2005; Armstrong, 2012; Okafor , 2005) and association (Dietrich, 1991; Aitken, 1957). Therefore, all of the completed and future work mentioned in the thesis can also be extended to regression and association subject areas for design of rule based systems in the future. On the other hand, the research methodology introduced in Chapter 3 is mainly based on deterministic. In the future, the methodology can also be extended to be based on probabilistic and fuzzy logic in practical applications.

Chapter 6 lists some impact factors for interpretability of rule based systems as well as some criteria for evaluation of the interpretability. In general, it applies to any types of expert systems. Therefore, in order to improve the interpretability of expert systems, it is necessary to address the four aspects namely, scaling up algorithms, scaling down data, selection of model representation and assessment of cognitive capability, in accordance with the criteria for evaluation of the interpretability.

Scaling up algorithms can improve the transparency in terms of depth of learning. For example, rule based methods usually generate models with good transparency because this type of learning is in a great depth and on an inductive basis. On the other hand, the performance of a learning algorithm would also affect the model complexity as mentioned in Chapter 5. In this case, the model complexity could be reduced by scaling up algorithms. In the context of rule based models, complexity could be reduced through proper selection of rule generation approaches. As mentioned in Chapter 2, the separate and conquer approach is usually likely to generate less complex rule sets than the divide and conquer approach. In addition, it is also helpful to employ pruning algorithms to simplify rule sets as introduced in Chapter 2. In this way, some redundant or irrelevant information is removed and thus the interpretability is improved.

Scaling down data usually results in the reduction of model complexity. This is because model complexity is usually affected by the size of data. In other words, if a data set has a large number of attributes with various values and instances, the generated model is very likely to be complex. As introduced in Chapter 1, the dimensionality issue can be resolved by using feature selection techniques, such as entropy (Shanno, 1948) and information gain, both of which are based on information theory pre-measuring uncertainty present in the data. In other words, the aim is to remove those irrelevant attributes and thus make a model simpler. In addition, the issue can also be resolved through feature extraction methods, such as Principal Component Analysis (PCA) (Jolliffe, 2002) and Linear Discriminant Analysis (LDA) (Yu & Yang, 2001). On the other hand, when a dataset contains a large number of instances, it is usually required to take advantage of sampling methods to choose the most representative instances. Some popular methods comprise simple random sampling (Yates, Moore, & Starnes, 2008), probabilistic sampling (Deming, 1975) and cluster sampling (Kerry & Bland, 1998). Besides, it is also necessary to remove attribute values due to the presence of irrelevant attribute values. For example, in a rule based method, an attribute-value pair may be never involved in any rules as a rule term. In this case, the value of this

attribute can be judged irrelevant and thus removed. In some cases, it is also necessary to merge some values for an attribute in order to reduce the attribute complexity especially when the attribute is continuous with a large interval. There are some ways to deal with continuous attributes such as ChiMerge (Kerber, 1992) and use of fuzzy linguistic terms (Ross, 2004).

As introduced in Chapters 2 and 3, a change of model representation would usually result in the change of model interpretability. As also introduced, rule based models could be represented in different forms such as decision tree and linear list. These two representations usually have redundancy present. For example, a decision tree may have the replicated subtree problem and a linear list may have the same attribute appear in different rules on a repetitive basis. This kind of problem could be resolved by converting to a rule based network representation as argued in Chapter 3. For other types of machine learning algorithms, it is also applicable to change the model representation in order to improve the interpretability. For example, mathematical formulas could be transformed into graphs or networks to make it easier to read and understand.

However, due to the difference in levels of expertise and personal preferences from different people, the same model representation may show different levels of comprehensibility for different people. For example, people who do not have a good background in mathematics may not like to read information in mathematical notations. In addition, people in social sciences may not understand technical diagrams used in engineering fields. On the basis of the above description, cognitive capability needs to be assessed to make the knowledge extracted from expert systems more interpretable to people in different domains. This can be resolved by using expert knowledge in cognitive psychology and human-machine engineering, or by following machine learning approaches to predict the capability as mentioned in Chapter 5.

The above discussion recommends that the four ways, namely, scaling up algorithms, scaling down data, selection of model representation and assessment of cognitive capability, can be adopted towards potential improvement of interpretability of expert systems in the future.

References

Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, (pp. 207-216). Washington D.C.

Aho, A. V., Hopcraft, J. E., & Ullman, J. D. (1983). *Data Structures and Algorithms.* Amsterdam: Addison-Wesley.

Aitken, A. (1957). *Statistical Mathematics* (8th ed.). Oliver & Boyd.

Aksoy, M. S. (2008). A Review of Rules Families of Algorithms. *Mathematical and Computational Applications, 1*(13), 51-60.

Alpaydın, E. (2004). *Introduction to Machine Learning (Adaptive Computation and Machine Learning).* MIT Press.

Altman, N. S. (1992). An introduction to kernel and nearest-neighbour nonparametric regression. *The American Statistician, 46*(3), 175–185.

Armstrong, J. S. (2012). Illusions in Regression Analysis. *International Journal of Forecasting, 28*(3), 689.

Avison, D., & Fitzgerald, G. (2006). *Information Systems Development: Methodologies, Techniques and Tools.* London: McGraw-Hill Education.

Azhagusundari, B., & Thanamani, A. S. (2013). Feature Selection based on Information Gain. *International Journal of Innovative Technology and Exploring Engineering, 2*(2), 18-21.

Biggs, N. L., Lloyd, E. K., & Wilson, R. J. (1986). *Graph Theory 1736-1936.* Oxford University Press.

Blachman, N. M. (1968). The Amount of Information That y Gives About X. *IEEE Transactions on Information Theory, 14*(1), 27-31.

Bondy, A., & Murty, U. (2008). *Graph Theory.* Berlin: Springer.

Brain, D. (2003). *Learning from Large Data, Bias, Variance, Sampling and Learning Curves.* PhD Thesis, Deakin University, Geelong, Victoria.

Bramer, M. (2000). Automatic induction of classification rules from examples using N-Prism. *Research and Development in Intelligent Systems. XVI*, pp. 99–121. Cambridge: Springer.

Bramer, M. (2002). Using J-Pruning to Reduce Overfitting in Classification Trees. *Knowledge Based Systems, 15*, 301-308.

Bramer, M. (2002). Using J-Pruning to Reduce Overfitting of Classification Rules in Noisy Domains. *Proceedings of 13th International Conference on Database and Expert Systems Applications— DEXA 2002.* Aix-en-Provence, France.

Bramer, M. (2005). Inducer: a public domain workbench for data mining. *International Journal of Systems Science, 36*(14), 909–919.

Bramer, M. (2007). *Principles of Data Mining.* London: Springer.

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*(2), 123–140.

Breiman, L. (2001). Random Forests. *Machine Learning, 45*(1), 5-32.

Brin, S., Motwani, R., Ullman, J. D., & Tsur, S. (1997). Dynamic itemset counting and implication rules for market basket data. *Proceedings ACM SIGMOD International Conference on Management of Data*, (pp. 255-264). Tucson, Arizona, USA.

Cendrowska, J. (1987). PRISM: an algorithm for inducing modular rules. *International Journal of Man-Machine Studies, 27,* 349–370.

Cormen , T. H., Leiserson , C. E., Rivest , R. L., & Stein, C. (2001). *Introduction to Algorithms* (2nd ed.). The MIT Press.

Deming, W. E. (1975). On probability as a basis for action. *The American Statistician, 29*(4), 146-152 .

Deng, X. (2012). *A Covering-based Algorithm for Classification: PRISM.* Lecture Note, University of Regina, Department of Computer Science, Regina.

Dietrich, C. F. (1991). *Uncertainty, Calibration and Probability: The Statistics of Scientific and Industrial Measurement* (2nd ed.). CRC Press .

Ekeberg, O., & Lansner, A. (1988). Automatic generation of internal representations in a probabilistic artificial neural network. *Proceedings of the First European Conference on Neural Networks.*

Freedman, D. A. (2005). *Statistical Models: Theory and Practice.* Cambridge University Press.

Freund , Y., & Schapire, R. E. (1996). Experiments with a New Boosting Algorithm. *Machine Learning: Proceedings of the Thirteenth International Conference*, (pp. 148-156). Bari.

Fürnkranz, J. (1999). Separate-and-Conquer Rule Learning. *Artificial Intelligence Review, 13*, 3-54.

Gegov, A. (2007). *Complexity Management in Fuzzy Systems.* Berlin: Springer.

Gegov, A. (2010). *Fuzzy Networks for Complex Systems.* Berlin: Springer.

Gegov, A. (2013). *Advanced Computation Models.* Lecture Notes, University of Portsmouth, School of Computing, Porstmouh.

Gegov, A., Petrov, N., & Vatchova, B. (2010). Advanced Modelling of Complex Processes by Rule Based Networks. *5th IEEE International Conference on Intelligent Systems* (pp. 197-202). London: IEEE.

Gegov, A., Petrov, N., Vatchova, B., & Sanders, D. (2011). Advanced Modelling of Complex Processes by Fuzzy Networks. *WSEAS Transactions on Circuits and Systems, 10*(10), 319-330.

Geng , L., & Hamilton, H. J. (2006). Interestingness measures for data mining: A survey. *ACM Computing Surveys, 38*(3).

Goode, H. H., & Machol, R. E. (1957). *System Engineering: An Introduction to the Design of Large-scale Systems.* McGraw-Hill.

Hahsler, M. (2015, February 13). *A Probabilistic Comparison of Commonly Used Interest Measures for Association Rules*. Retrieved February 20, 2015, from http://michael.hahsler.net/research/association_rules/measures.html

Hall, A. D. (1962). *A Methodology for Systems Engineering.* Van Nostrand.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explorations, 11*(1), 10-18.

Han, J., & Kamber, M. (2006). *Data Mining: Concept and Techniques* (2nd ed.). San Francisco: Morgan Kaufmann Publishers.

Hart, A. (1989). *Knowledge acquisition for expert systems.* London: Chapman and Hall.

Higgins, C. M. (1993). *Classification and Approximation with Rule Based Networks.* PhD Thesis, California Institute of Technology, Department of Electrical Engineering, Pasadena, California.

Jech, T. (2003). *Set Theory* (The Third Millennium Edition ed.). Berlin, New York: Springer.

Johnsonbaugh, R. (2008). *Discrete Mathematics* (7th ed.). Prentice Hall.

Jolliffe, I. T. (2002). *Principal Component Analysis.* New York: Springer.

Kerber, R. (1992). ChiMerge: Discretization of Numeric Attributes. *Proceedings of the 10th National Conference on Artificial Intelligence* (pp. 123-128). California: AAAI Press.

Kerry, S. M., & Bland, J. M. (1998). The intracluster correlation coefficient in cluster randomisation. *British Medical Journal, 316*(7142), 1455–1460.

Kononenko, I. (1989). Bayesain Neual Networks. *Biologi-cal Cybernetics, 61*, 361-370.

Kononenko, I., & Kukar, M. (2007). *Machine Learning and Data Mining: Introduction to Principles and Algorithms.* Chichester, West Sussex: Horwood Publishing Limmited.

Langley, P. (1995). *Elements of machine learning.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Li, J., & Liu, H. (2003). *Kent Ridge Bio-medical Dataset*. (I2R Data Mining Department) Retrieved May 18, 2015, from http://datam.i2r.a-star.edu.sg/datasets/krbd/

Li, J., & Wong, L. (2004). Rule-Based Data Mining Methods for Classification Problems in Biomedical Domains. *A tutorial note forthe 15th European Conference on Machine Learning (ECML) and the 8th European Conference on Principles and Practice for of Knowledge Discovery in Databases (PKDD).* Pisa.

Lichman, M. (2013). *UCI Machine Learning Repository*. (Irvine, CA: University of California, School of Information and Computer Science) Retrieved from http://archive.ics.uci.edu/ml

Liu, W. Z., & White, A. P. (1991). A review of inductive learning. *Research and Development in Expert Systems. VIII*, pp. 112-126. Cambridge: Springer.

Machine Learning Open Source Software. (2000). Retrieved June 15, 2014, from JMLR: http://jmlr.org/mloss/

Michalski, R. S. (1969). On the quasi-minimal solution of the general covering problem. *Proceedings of the Fifth International Symposium on Information Processing*, (pp. 125–128). Bled, Yugoslavia.

Michalski, R. S., Mozetic, I., Hong, J., & Lavrac, N. (1986). The multi-purpose incremental learning system AQ15 and its testing application to three medical domains. *Proceedings of the 5th National Conference on Artificial Intelligence* (pp. 1041-1044). Philadelphia, PA: AAAI Press.

Michiel, H. e. (2001). Bayes formula. In *Encyclopedia of Mathematics.* Springer.

Mitchell, T. M. (1997). *Machine Learning.* McGraw Hill.

Mrozek, A. (1992). A new method for discovering rules from examples in expert systems. *International Journal of Man-Machine Studies, 36*, 127-143.

Murdoch, J., & Barnes, J. A. (1973). *Statistics: Problems and Solutions.* London and Basingstoke: The Macmillan Press Ltd.

Okafor , A. (2005). *Entropy Based Techniques with Applications in Data Mining.* PhD Thesis, University of Florida, Florida.

Partridge, D., & Hussain, K. M. (1994). *Knowledge Based Information Systems.* Mc-Graw Hill.

Piatetsky-Shapiro, G. (1991). Discovery, analysis, and presentation of strong rules. In G. Piatetsky-Shapiro, & W. Frawley, *Knowledge Discovery in Databases.* Cambridge: AAAI/MIT Press.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2007). Section 16.5. Support Vector Machines. In *Numerical Recipes: The Art of Scientific Computing.* New York: Cambridge University Press.

Quinlan, R. (1987). Inductive knowledge acquisition: a case study. In R. Quinlan (Ed.), *Applications of Expert Systems* (pp. 157-173). Turing Institute Press.

Quinlan, R. (1988). Induction, Knowledge and expert systems. In J. S. Gero, & R. Stanton (Eds.), *Artifical Intelligence Developments and Applications* (pp. 253-271). Amsterdam: Elsevier Science.

Quinlan, R. (1993). *C4.5: Programs for Machine Learning.* Morgan Kaufman.

Rosenblatt, F. (1962). *Principles of Neurodynamics: Perceptron and the Theory of Brain Mechanisms.* Washington, DC: Spartan Books.

Ross, T. J. (2004). *Fuzzy Logic with Engineering Applications.* West Sussex: John Wiley & Sons Ltd.

Sage, A. P. (1992). *Systems Engineering.* Wiley IEEE.

Schaffer, C. (1993). Overfitting Avoidance as Bias. *Machine Learning, 10,* 153–178.

Schlager, K. J. (1956). Systems engineering: key to modern development. *IRE Transactions EM, 3*(3), 64–66.

Schneider, S. (2001). *The B-Method: An Introduction.* Basingstoke& New York: Palgrave Macmillan.

Shanno, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal, 27*(3), 379-423.

Simpson, S. G. (2013). *Mathematical Logic.* PA.

Smyth, P., & Goodman, R. M. (1991). Rule Induction Using Information Theory. In G. Piatetsky-Shapiro, & W. J. Frawley (Eds.), *Knowledge Discovery in Databases* (pp. 159-176). AAAI Press.

Smyth, P., & Goodman, R. M. (1992). An Information Theoretic Approach to Rule Induction from Databases. *IEEE Transactions on Knowledge and Data Engineering, 4*(4), 301-316.

Stahl, F., & Bramer, M. (2011). Induction of modular classification rules: using Jmax-pruning. *In: In Thirtieth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence.* Cambridge: Springer.

Stahl, F., & Bramer, M. (2011). Random Prism: an alternative to Random Forests. *Research and Development in Intelligent Systems. XXVIII*, pp. 5-18. Cambridge: Springer.

Stahl, F., & Bramer, M. (2012). Jmax-pruning: A facility for the information theoretic pruning of modular classification rules. *Knowledge-Based Systems, 29,* 12-19.

Stahl, F., & Bramer, M. (2013). Random Prism: a noise-tolerant alternative to Random Forests. *Expert Systems, 31*(5), 411-420.

Stahl, F., & Bramer, M. (2013). Scaling Up Classification Rule Induction through Parallel Processing. *The Knowledge Engineering Review, 28*(4), 451-478.

Stahl, F., & Jordanov, I. (2012). An overview of use of neural networks for data mining tasks. *WIREs: Data Mining and Knowledge Discovery*, 193-208.

Stahl, F., Gaber, M. M., Aldridge, P., May, D., Liu, H., Bramer, M., & Yu, P. S. (2012). Heterogeneous Distributed Classification for Pocket Data Mining. (A. Hameurlain, J. Küng, & R. Wagner, Eds.) *Transactions on large-scale data and knowledge-centered systems, V*, 183-205.

Stahl, F., Gaber, M. M., Liu, H., Bramer, M., & Yu, P. S. (2011). Distributed classification for pocket data mining. *Proceedings of the 19th International Symposium on Methodologies for Intelligent Systems* (pp. 336-345). Warsaw: Springer.

Stichweh, R. (2011). Systems Theory. In B. Badie (Ed.), *International Encyclopaedia of Political Science.* New York: Sage.

Tan, P.-N., Kumar, V., & Srivastava, J. (2004). Selecting the right objective measure for association analysis. *Information Systems, 29*, 293–313.

Tan, P.-N., Steinbach, M., & Kumar, V. (2006). *Introduction to Data Mining.* New Jersey: Pearson Education.

Uttley, A. M. (1959). The Design of Conditional Probability Computers. *Information and control, 2*, 1-24.

Vlachos, P. (2005 , July 19). *StatLib---Datasets Archive*. (Carnegie Mellon University) Retrieved May 18, 2015, from http://lib.stat.cmu.edu/datasets/

Wolpert, D. H. (1993). *On Overfitting Avoidance as Bias.* Technical Report, SFI TR.

Wooldridge, M. (2002). *An Introduction to Multi-Agent Systems.* John Wiley & Sons.

Yates, D. S., Moore, D. S., & Starnes, D. S. (2008). *The Practice of Statistics* (3rd ed.). Freeman.

Yu, H., & Yang, J. (2001). A direct LDA algorithm for high-dimensional data — with application to face recognition. *Pattern Recognition, 34*(10), 2067–2069.

Appendix I UML Diagrams

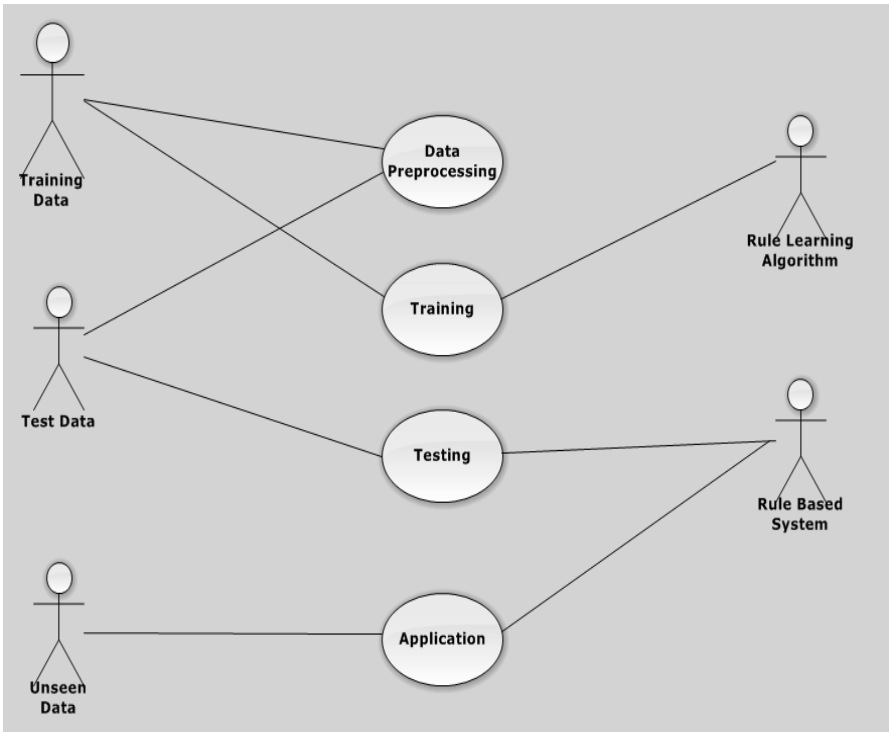

Fig.A.1 UML Use Case Diagram for machine learning scenarios

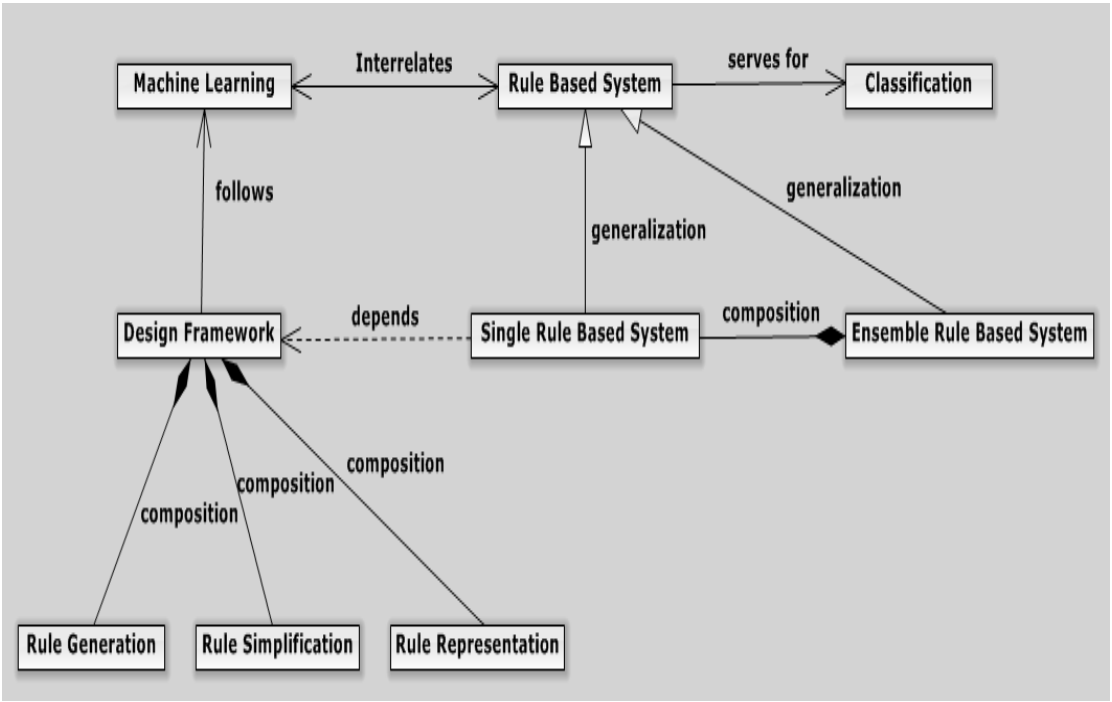

Fig.A.2 UML Class Diagram for Research Framework

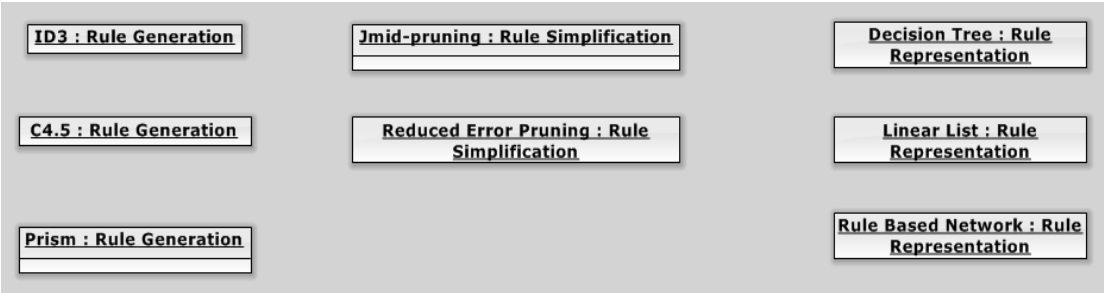

Fig.A.3 UML Instance Diagram for generation, simplification and representation of rules

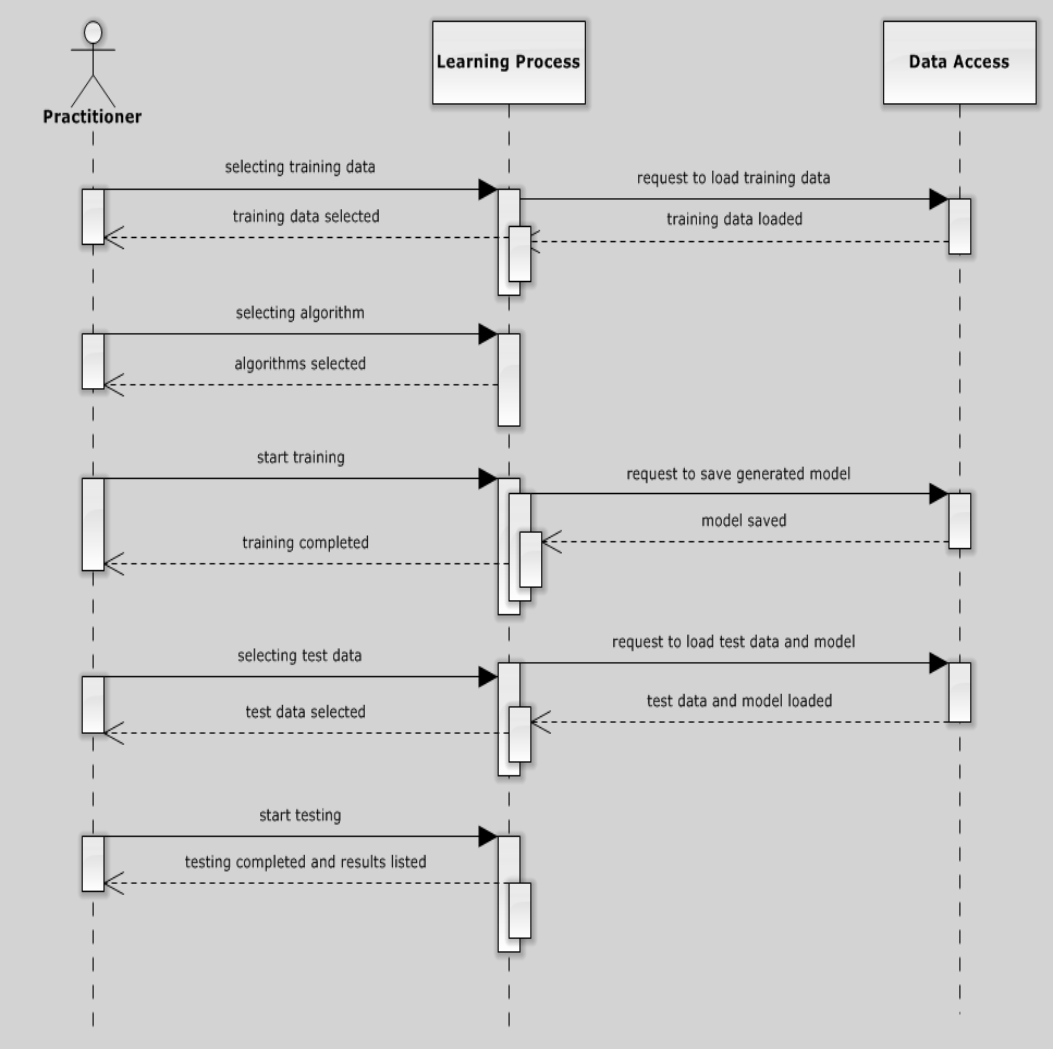

Fig.A.4 UML Sequence Diagram for machine learning systems

Appendix II Data Flow Diagram

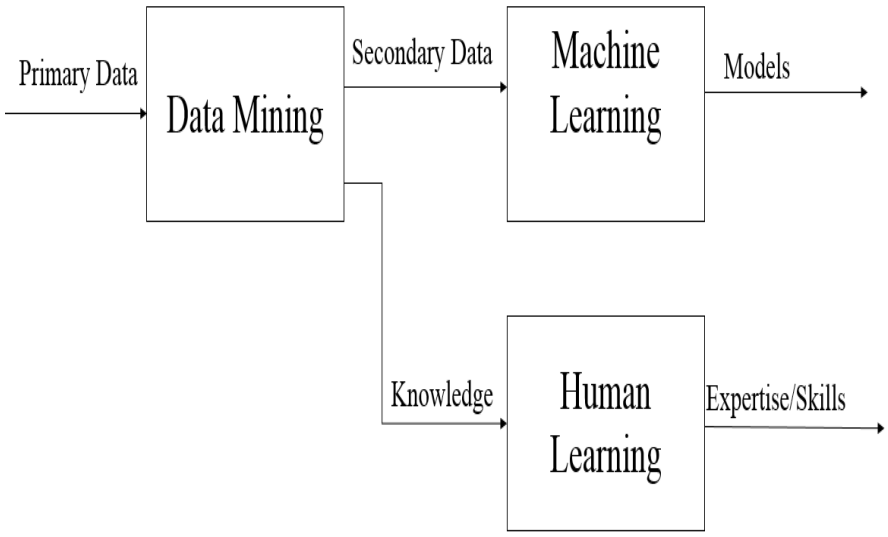

Fig.A.5 Chained relationship between data mining and machine learning

Appendix III Glossary

Terms in Machine Learning	Terms in other related areas
attribute, feature	variable, field, column
instance	record, data point, tuple, row
training, learning	modelling, building, construction
testing, prediction	verification, validation, checking
classifier, learner	model, expert system, hypothesis
inconsistency	overlapping
missing value	unknown value
dimensionality	number of attributes/variables
data size	number of instances/data points
classification	categorical prediction, decision
regression	numerical prediction
association	correlation
clustering	grouping
noise	incorrect record
classification/regression/association rules	if-then rules, decision rules
classification/regression trees	decision trees
efficiency in training stage	modelling efficiency
efficiency in testing stage	prediction efficiency
computational complexity	time complexity
rule based classifier	rule set
rule based learner	rule based model, rule based system
rule based ensemble learner	ensemble rule based system
class, label	output
attribute value	input/output

Appendix IV Empirical Results on Medical Data

Table A.1 accuracy for IEBRG vs Prism

Dataset	Prism	IEBRG	Random Classifier
ALL-AML	**91%**	91%	47%
colonTumor	**80%**	73%	52%
DLBCLOutcome	**56%**	48%	48%
DLBCLTumor	76%	**79%**	61%
DLBCL-Stanford	80%	**92%**	49%
LungCancer-Harvard2	80%	**99%**	50%
lung-Michigan	**94%**	**94%**	80%
lungcancer-ontario	**60%**	57%	49%
MLL_Leukemia	**80%**	60%	33%
NervousSystem	42%	**50%**	53%
prostate_tumorVSNormal	26%	**74%**	59%
BCR-ABL	96%	**98%**	89%
E2A-PBX1	**100%**	98%	84%
Hyperdip50	93%	**96%**	68%
MLL	92%	**99%**	89%
T-ALL	**100%**	**100%**	77%
TEL-AML1	**95%**	88%	63%
pos_neg_100	61%	**73%**	50%

Table A.2 number of rules and average number of terms

Dataset	Prism		IEBRG	
	Count(rules)	Ave(terms)	Count (rules)	Ave(terms)
ALL-AML	**2**	1.0	**2**	1.0
colonTumor	**3**	1.0	4	1.0
DLBCLOutcome	4	1.0	**3**	1.0
DLBCLTumor	4	1.0	**3**	1.0
DLBCL-Stanford	**3**	1.0	**3**	1.0
LungCancer-Harvard2	**2**	1.0	**2**	1.0
lung-Michigan	**2**	1.0	**2**	1.0
lungcancer-ontario	4	1.0	**3**	1.0
MLL_Leukemia	4	1.0	**3**	1.0
NervousSystem	**4**	1.0	**4**	1.0
prostate_tumorVSNormal	**4**	1.0	**4**	1.0
BCR-ABL	**3**	1.0	**3**	1.0
E2A-PBX1	**2**	1.0	**2**	1.0
Hyperdip50	5	1.0	**4**	1.0
MLL	4	1.0	**2**	1.0
T-ALL	2	1.0	**2**	1.0
TEL-AML1	4	1.0	**3**	1.0
pos_neg_100	**12**	1.0	**12**	1.0

Table A.3 numbers of generated terms and discarded terms

Dataset	Prism		IEBRG	
	Generated terms	Dropped terms	Generated terms	Dropped terms
ALL-AML	**2**	0	**2**	0
colonTumor	**3**	0	4	0
DLBCLOutcome	4	0	**3**	0
DLBCLTumor	4	0	**3**	0
DLBCL-Stanford	**3**	0	**3**	0
LungCancer-Harvard2	**2**	0	**2**	0
lung-Michigan	**2**	0	**2**	0
lungcancer-ontario	4	0	**3**	0
MLL_Leukemia	4	0	**3**	0
NervousSystem	**4**	0	**4**	0
prostate_tumorVSNormal	**4**	0	**4**	0
BCR-ABL	**3**	0	**3**	0
E2A-PBX1	**2**	0	**2**	0
Hyperdip50	5	0	**4**	0
MLL	4	0	**2**	0
T-ALL	**2**	0	**2**	0
TEL-AML1	4	0	**3**	0
pos_neg_100	12	0	**12**	0

Table A.4 runtime in million seconds for IEBRG vs Prism

Dataset	Prism	IEBRG
ALL-AML	41642	**2765**
colonTumor	21313	**1469**
DLBCLOutcome	85472	**4438**
DLBCLTumor	92316	**5203**
DLBCL-Stanford	32970	**1750**
LungCancer-Harvard2	49627	**19829**
lung-Michigan	56721	**5406**
lungcancer-ontario	9797	**985**
MLL_Leukemia	89003	**9672**
NervousSystem	60377	**5516**
prostate_tumorVSNormal	179991	**12688**
BCR-ABL	35876	**14797**
E2A-PBX1	42611	**11970**
Hyperdip50	70237	**18064**
MLL	40939	**6688**
T-ALL	60799	**12891**
TEL-AML1	43643	**15532**
pos_neg_100	9282	**8422**

Table A.5 clash rate for IEBRG vs Prism

Dataset	Prism	IEBRG
ALL-AML	0.0	0.0
colonTumor	0.0	0.0
DLBCLOutcome	0.0	0.0
DLBCLTumor	0.0	0.0
DLBCL-Stanford	0.0	0.0
LungCancer-Harvard2	0.0	0.0
lung-Michigan	0.0	0.0
lungcancer-ontario	0.0	0.0
MLL_Leukemia	0.0	0.0
NervousSystem	0.0	0.0
prostate_tumorVSNormal	0.0	0.0
BCR-ABL	0.0	0.0
E2A-PBX1	0.0	0.0
Hyperdip50	0.0	0.0
MLL	0.0	0.0
T-ALL	0.0	0.0
TEL-AML1	0.0	0.0
pos_neg_100	0.0	0.0

Appendix V Recalls on Rule Generation

Table A.6 Recalls for anneal data

Class index	Prism	IEBRG
0	13%	0%
1	27%	51%
2	92%	93%
3	N.A	N.A
4	97%	97%
5	88%	100%

Table A.7 Recalls for balance-scale data

Class index	Prism	IEBRG
0	44%	82%
1	0%	4%
2	33%	56%

Table A.8 Recalls for credit-a data

Class index	Prism	IEBRG
0	75%	81%
1	46%	73%

Table A.9 Recalls for credit-g data

Class index	Prism	IEBRG
0	73%	85%
1	37%	24%

Table A.10 Recalls for iris data

Class index	Prism	IEBRG
0	100%	100%
1	16%	92%
2	92%	88%

Table A.11 Recalls for breast-cancer data

Class index	Prism	IEBRG
0	55%	69%
1	59%	45%

Table A.12 Recalls for breast-w data

Class index	Prism	IEBRG
0	93%	94%
1	87%	86%

Table A.13 Recalls for diabetes data

Class index	Prism	IEBRG
0	74%	83%
1	44%	50%

Table A.14 Recalls for heart-statlog data

Class index	Prism	IEBRG
0	60%	71%
1	73%	65%

Table A.15 Recalls for hepatitis data

Class index	Prism	IEBRG
0	47%	50%
1	80%	87%

Table A.16 Recalls for ionosphere data

Class index	Prism	IEBRG
0	70%	55%
1	99%	99%

Table A.17 Recalls for kr-vs-kp data

Class index	Prism	IEBRG
0	30%	90%
1	75%	77%

Table A.18 Recalls for lymph data

Class index	Prism	IEBRG
0	100%	50%
1	72%	80%
2	57%	74%
3	50%	0%

Table A.19 Recalls for mushroom data

Class index	Prism	IEBRG
0	91%	87%
1	95%	94%

Table A.20 Recalls for segment data

Class index	Prism	IEBRG
0	51%	73%
1	88%	100%
2	43%	49%
3	24%	62%
4	40%	45%
5	58%	92%
6	86%	98%

Table A.21 Recalls for zoo data

Class index	Prism	IEBRG
0	95%	100%
1	95%	95%
2	20%	0%
3	8%	100%
4	25%	100%
5	25%	13%
6	0%	100%

Table A.22 Recalls for wine data

Class index	Prism	IEBRG
0	85%	92%
1	75%	89%
2	73%	71%

Table A.23 Recalls for car data

Class index	Prism	IEBRG
0	100%	91%
1	0%	35%
2	0%	10%
3	0%	0%

Table A.24 Recalls for page-blocks data

Class index	Prism	IEBRG
0	97%	96%
1	62%	67%
2	36%	0%
3	83%	72%
4	26%	7%

Table A.25 Recalls for vote data

Class index	Prism	IEBRG
0	99%	91%
1	54%	83%

Table A.26 Recalls for lung-cancer data

Class index	Prism	IEBRG
0	78%	33%
1	87%	87%

Table A.27 Recalls for cmc data

Class index	Prism	IEBRG
0	45%	50%
1	0%	15%
2	25%	18%

Table A.28 Recalls for optdigits data

Class index	Prism	IEBRG
0	18%	65%
1	20%	25%
2	11%	18%
3	20%	9%
4	45%	53%
5	13%	18%
6	51%	25%
7	31%	45%
8	12%	20%
9	15%	9%

Table A.29 Recalls for contact-lenses data

Class index	Prism	IEBRG
0	40%	40%
1	25%	50%
2	67%	67%

Table A.30 Recalls for colonTumor data

Class index	Prism	IEBRG
0	59%	50%
1	88%	83%

Table A.31 Recalls for DLBCLOutcome data

Class index	Prism	IEBRG
0	69%	53%
1	23%	27%

Table A.32 Recalls for DLBCL-Stanford data

Class index	Prism	IEBRG
0	79%	96%
1	57%	61%

Table A.33 Recalls for lung-Michigan

Class index	Prism	IEBRG
0	99%	99%
1	0%	0%

Table A.34 Recalls for lungcancer-ontario

Class index	Prism	IEBRG
0	75%	67%
1	0%	7%

Table A.35 Recalls for centralNervousSystem-outcome data

Class index	Prism	IEBRG
0	52%	29%
1	36%	62%

Table A.36 Recalls for pos_neg_100 data

Class index	Prism	IEBRG
0	32%	16%
1	70%	91%

Table A.37 Recalls for prostate_outcome data

Class index	Prism	IEBRG
0	25%	0%
1	15%	69%

Table A.38 Recalls for weather data

Class index	Prism	IEBRG
0	22%	33%
1	60%	40%

Appendix VI Empirical Results on Noise Tolerance

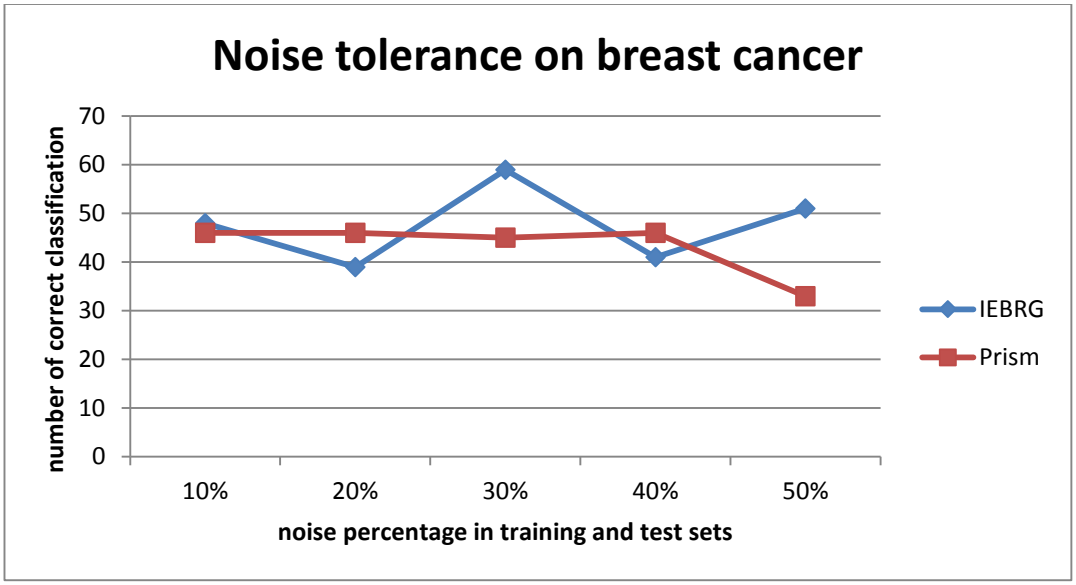

Fig. A.6 Noise tolerance on breast cancer

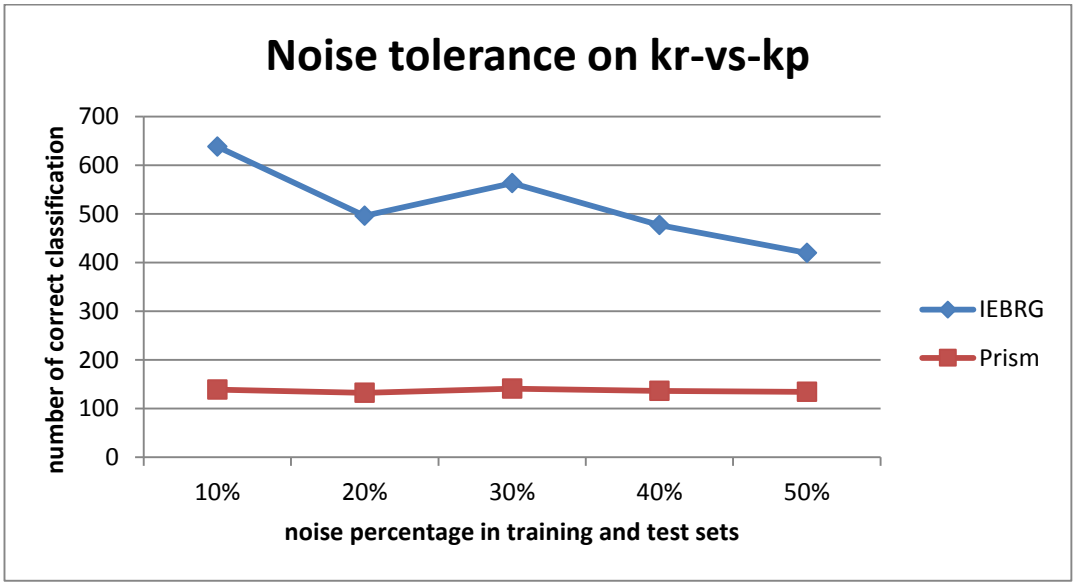

Fig.A.7 Noise tolerance on kr-vs-kp

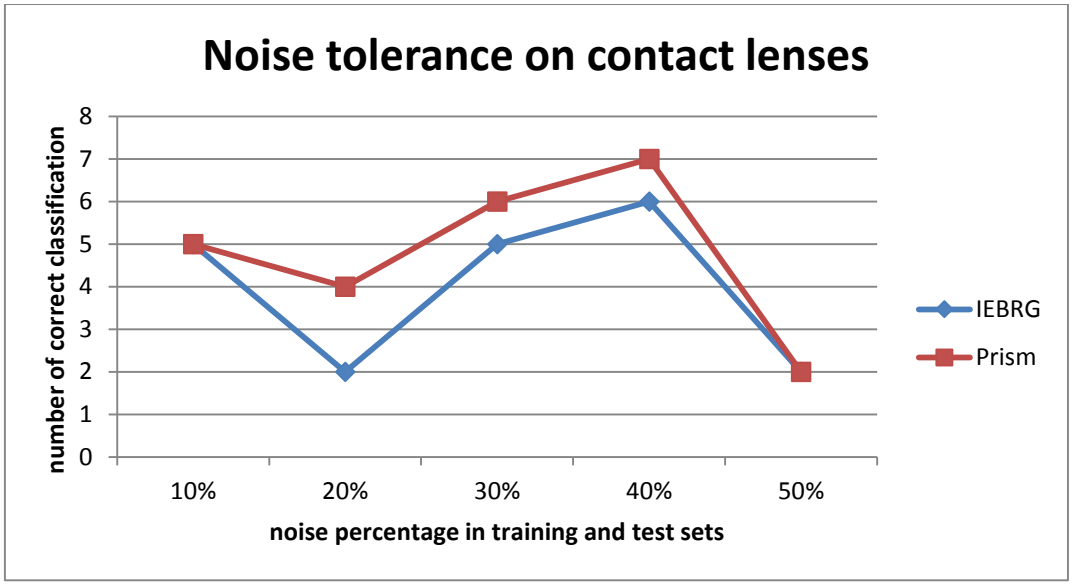

Fig.A.8 Noise tolerance on contact lenses

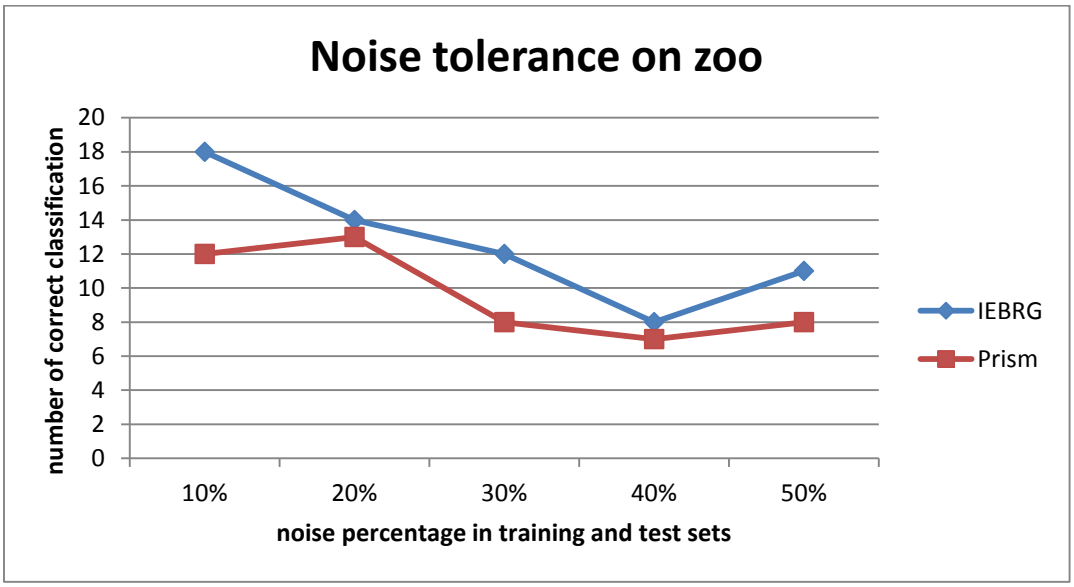

Fig.A.9 Noise tolerance on zoo

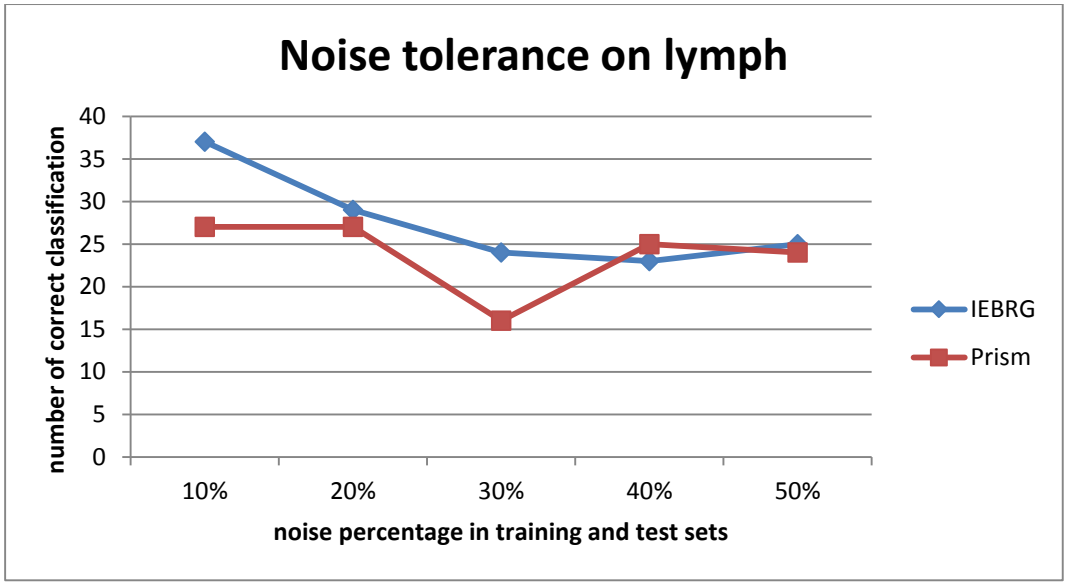

Fig.A.10 Noise tolerance on lympth

Appendix VII Characteristics of Data Sets

Name	Attribute Types	#Attributes	#Instances	#Classes
anneal	discrete, continuous	38	798	6
credit-g	discrete, continuous	20	1000	2
diabetes	discrete, continuous	20	768	2
heart-stalog	continuous	13	270	2
ionosphere	continuous	34	351	2
iris	continuous	4	150	3
kr-vs-kp	discrete	36	3196	2
lymph	discrete, continuous	19	148	4
segment	continuous	19	2310	7
zoo	discrete, continuous	18	101	7
wine	continuous	13	178	3
breast-cancer	discrete	9	286	2
car	discrete	6	1728	4
breast-w	continuous	10	699	2
credit-a	discrete, continuous	15	690	2
heart-c	discrete, continuous	76	920	4
heart-h	discrete, continuous	76	920	4
hepatitis	discrete, continuous	20	155	2
mushroom	discrete	22	8124	2
vote	discrete	16	435	2
lung-cancer	discrete	32	57	3
labor	discrete, continuous	17	57	2
contact-lenses	discrete	4	24	3
banlance-scale	discrete	4	625	3

Name	Attribute Types	#Attributes	#Instances	#Classes
weather	discrete, continuous	5	14	2
nursery	discrete	9	12960	5
ti-tac-toe	discrete	9	958	2
yeast	continuous	8	1484	2
page blocks	continuous	10	5473	5
opt digits	continuous	64	5620	10
dorothea	continuous	100000	1950	2
elcoli	continuous	23	336	2
glass	continuous	10	214	7
moke problems	discrete	7	432	2
shuttle	discrete	10	58000	7
cmc	discrete, continuous	10	1473	3
ALL-AML	continuous	7130	72	2
colonTumor	continuous	2001	62	2
DLBCLOutcome	continuous	7130	58	2
DLBCLTumor	continuous	7130	77	2
DLBCL-Stanford	continuous	4027	47	2
LungCancer-Harvard2	continuous	12534	32	2
lung-Michigan	continuous	7130	96	2
lungcancer-ontario	continuous	2881	39	2
MLL_Leukemia	continuous	12583	72	3
NervousSystem	continuous	7130	60	2
prostate_tumorVSNormal	continuous	12601	136	2
BCR-ABL	continuous	12559	327	2
E2A-PBX1	continuous	12559	327	2
Hyperdip50	continuous	12559	327	2

Name	Attribute Types	#Attributes	#Instances	#Classes
MLL	continuous	12559	327	2
T-ALL	continuous	12559	327	2
TEL-AML1	continuous	12559	327	2
pos_neg_100	continuous	928	13375	2
analcatdata_happiness	discrete, continuous	4	60	3
analcatdata_asbestos	discrete, continuous	4	83	3

FORM UPR16

Research Ethics Review Checklist

University of Portsmouth

Please complete and return the form to Research Section, Quality Management Division, Academic Registry, University House, with your thesis, prior to examination

Postgraduate Research Student (PGRS) Information			**Student ID:**	**472297**
Candidate Name:	**Han Liu**			
Department:	**Computing**	**First Supervisor:**	**Dr Alexander Gegov**	
Start Date: (or progression date for Prof Doc students)	**1 Feb 2013**			

Study Mode and Route:	Part-time ❑	MPhil ❑	Integrated Doctorate (NewRoute) ❑
	Full-time ×	MD ❑	Prof Doc (PD) ❑
		PhD ×	

Title of Thesis:	Rule Based Systems for Classification in Machine Learning Context
Thesis Word Count: (excluding ancillary data)	43720

If you are unsure about any of the following, please contact the local representative on your Faculty Ethics Committee for advice. Please note that it is your responsibility to follow the University's Ethics Policy and any relevant University, academic or professional guidelines in the conduct of your study

Although the Ethics Committee may have given your study a favourable opinion, the final responsibility for the ethical conduct of this work lies with the researcher(s).

UKRIO Finished Research Checklist:
(If you would like to know more about the checklist, please see your Faculty or Departmental Ethics Committee rep or see the online version of the full checklist at: http://www.ukrio.org/what-we-do/code-of-practice-for-research/)

a) Have all of your research and findings been reported accurately, honestly and within a reasonable time frame?	**YES**
b) Have all contributions to knowledge been acknowledged?	**YES**
c) Have you complied with all agreements relating to intellectual property, publication and authorship?	**YES**
d) Has your research data been retained in a secure and accessible form and will it remain so for the required duration?	**YES**
e) Does your research comply with all legal, ethical, and contractual requirements?	**YES**

*Delete as appropriate

Candidate Statement:	
I have considered the ethical dimensions of the above named research project, and have successfully obtained the necessary ethical approval(s)	
Ethical review number(s) from Faculty Ethics Committee (or from NRES/SCREC):	DBB4-87D6-F0CE-6C8B-31C4-6724-43AC-FB5F
Signed: *(Student)* Han Lin	**Date: 14 Oct 2015**
If you have *not* submitted your work for ethical review, and/or you have answered 'No' to one or more of questions a) to e), please explain why this is so:	
Signed: *(Student)*	**Date:**

UPR 16 (2013) – November 2013