A Variable Precision Rough Set Theory Decision Support System: With an Application to Bank Rating Prediction

Benjamin Griffiths

Doctor of Philosophy

Accounting and Finance Section
Cardiff Business School
Cardiff University

August 2008

Acknowledgements

I would like to gratefully acknowledge my brother Richard Griffiths for his most generous support and advice throughout, and prior to, the undertaking of this thesis. Without whom, this research would not have been possible. I would also like to thank my supervisor Dr. Malcolm Beynon, for his highly diligent supervision, and the time and effort spent helping with my work, which undoubtedly surpasses that normally expected of a supervisor. As with his own research, Dr Beynon has expected my work to be of the highest standard, a standard I will now always strive to achieve in the future.

Additional thanks must be given to Pete Harnett and Dr. Paul Young, for the very valuable time (literally!) they took, to give me support and advice on my research. They provided me with an incite into the realities and challenges I needed to consider outside the academic environment, and provided excellent feedback during the research development.

Finally, very special thanks are to be given to my family and friends, who have supported me in so many ways, your generosity does not go unnoticed and is a tribute to you all.

Abstract

This dissertation considers, the Variable Precision Rough Sets (VPRS) model, and its development within a comprehensive software package (decision support system), incorporating methods of re-sampling and classifier aggregation. The concept of β-reduct aggregation is introduced, as a novel approach to classifier aggregation within the VPRS framework. The software is applied to the credit rating prediction problem, in particularly, a full exposition of the prediction and classification of Fitch's Individual Bank Strength Ratings (FIBRs), to a number of banks from around the world is presented.

The ethos of the developed software was to rely heavily on a simple 'point and click' interface, designed to make a VPRS analysis accessible to an analyst, who is not necessarily an expert in the field of VPRS or decision rule based systems. The development of the software has also benefited from consultations with managers from one of Europe's leading hedge funds, who gave valuable insight, advice and recommendations on what they considered as pertinent issues with regards to data mining, and what they would like to see from a modern data mining system.

The elements within the developed software reflect each stage of the knowledge discovery process, namely, pre-processing, feature selection, data mining, interpretation and evaluation. The developed software encompasses three software packages, a pre-processing package incorporating some of the latest pre-processing and feature selection methods; a VPRS data mining package, based on a novel "vein graph" interface, which presents the analyst with selectable β-reducts over the domain of β; and a third more advanced VPRS data mining package, which essentially automates the vein graph interface for incorporation into a re-sampling environment, and also implements the introduced aggregated β-reduct, developed to optimise and stabilise the predictive accuracy of a set of decision rules induced from the aggregated β-reduct.

A Variable Precision Rough Set Theory Decision Support System: With an Application to Bank Rating Prediction

Contents

List of Tables

List of Figures

Chapter 1

Introduction

> "It is often said that the soul of a technology-driven economy is continuous innovation. No successful enterprise can pat itself on the back for last year's software..."
>
> Aburdene (2007, pp. vii)

Throughout history, information, or knowledge in many forms, has always been a tradable commodity. It is a curious fact that historians consider history, as opposed to pre-history, as beginning with the earliest examples of written information; be it, the ancient Egyptian hieroglyphs or Mesopotamian cuneiform, used some 5,000 years ago (Carr, 2001).

Within modern times, the information industry has become considered one of the most important sectors within today's evolving economic structure (Whitehorn and Whitehorn, 1999; Wu, 2002; Business Intelligence Channel, 2008; Rooney et al., 2005). There are a number of types of information industries, providing information on a wide range of areas such as, scientific, technical, medical, media, and relevant to this dissertation, business and financial information (Fayyad et al., 1996).

Business information providers (IPs), such as, Reuters Group, Bloomberg and Dow Jones Newswires, in themselves, supply their customers with a vast variety of information and financial market data (Bloomberg, 2008; Dow Jones Newswires, 2008; Reuters Group, 2008). Other notable

IPs, such as Bureau van Dijk supply online, research targeted databases, concerned with public and private companies, banks and insurance firms (Bureau van Dijk, 2008).

In his 1984 book Megatrends, Naisbitt (pp. 24) wrote, "We are drowning in information, but starved for knowledge...", this is a sentiment which still holds true over 20 years later. With the increasing rate at which data is being collected (Brachman et al., 1996), and given business IPs ability to task their sources to collect such vast quantities of data, is, as Naisbitt suggested, fruitless unless knowledge can be gleaned from it (also see, Naisbitt, 1988, 1996). The business journalist Aburdene (2007), argues that the information revolution is now over, and that we are at the beginning of a new epoch, where, what she terms, the 'concious individual' who can create technology to exploit the information industry, will be the driving force behind a new revolution.

Utilising modern technology (computers), the process by which knowledge is extracted from databases of information, is commonly known as data mining and is seen as a major step of the more broader discipline of Knowledge Discovery in Databases KDD (Han et al., 2003). Where knowledge management, is the business process, by which companies organise, collect and assimilate this knowledge into their systems (Zorn and Taylor, 2003).

Due to the volume of data available, and facilitated by the advances made in modern computers, new techniques are being developed, both in industry and academia, to exploit this increasing abundance of information. Tay et al. (2003, pp. 1) notes that:

> "A new generation of techniques and tools is emerging to intelligently assist humans in analyzing mountains of data, finding useful knowledge and in some cases performing analysis automatically..."

This dissertation develops one of the more recent data mining methods, based on Variable Precision Rough Sets (VPRS) (Ziarko, 1993a), an extension of Rough Set Theory (RST) (Pawlak, 1982), within the field of quantitative financial analysis. In more detail, bespoke, prototype Decision Support Software is developed (a specific type of KDD system, described later), where the software incorporates a full suite of facilities capable of tackling some of the most relevant issues within the field KDD and data mining. In particular, one of the main focuses of this dissertation was the development a framework to facilitate, re-sampling and ensemble methods, within the VPRS model (described later).

This research has benefited from support and advise, by professional investment managers, from one of Europe's leading hedge funds. Whose expressed interest, is grounded in the fact that, for the daily challenges that face the modern data analyst, there is a dearth of accessible data mining tools developed for their industry (Harnett and Young, 2004, 2007). They have offered advice on what they would expect, and would like, from a modern data mining package, and described the problems within their current systems. Though, this is not to say, that the developed software is specific to their field of work, however, the developed software is considered here, in the context of financial quantitative research, and is expounded using the problem of, predicting bank rating classifications.

The remaining sections of this introductory chapter are outlined below:

- Section 1.1. **KDD and Data Mining**. This section provides an introductory overview of KDD and data mining, including: the roles of RST, in particular, VPRS as a modern approach to data mining; and re-sampling and ensemble methods within KDD, as modern evaluation, stabilisation and optimisation methods.
- Section 1.2. **Development of the VPRS Decision Support Software**. This section provides a general description of the extensive software developed for this dissertation, describing the methods implemented to assist an analyst through the KDD process.
- Section 1.3. **Chapter Synopsis**. This section presents a chapter synopsis, outlining the remainder of the dissertation.

1.1 KDD and Data Mining

Due to their contemporary nature, there are no strict definitions of KDD and data mining. Frawley et al. (1992, pp. 58) described KDD as the, "non-trivial extraction of implicit, unknown, and potentially useful information from data". Although the terms, KDD and data mining, are often used synonymously (data mining is considered to be the more popular term, Piatetsky-Shapiro, 2000), a clear distinction can be drawn, as stated by Fayyad et al. (1996, pp. 39):

> "KDD refers to the overall process of discovering useful knowledge from data, and data mining refers to a particular step in this process. Data mining is the application of specific algorithms for extracting patterns from data..."

Hence, data mining can be considered as just one stage within the KDD process. Figure 1.1.1 presents the stages of the KDD process (Brachman and Anand, 1996; Fayyad et al., 1996), with each stage, a transition of the data from the source database to the discovered knowledge. The stages of the knowledge discovery process described here, reflect more closely the analysis undertaken throughout this dissertation and may not be representative of all KDD processes.

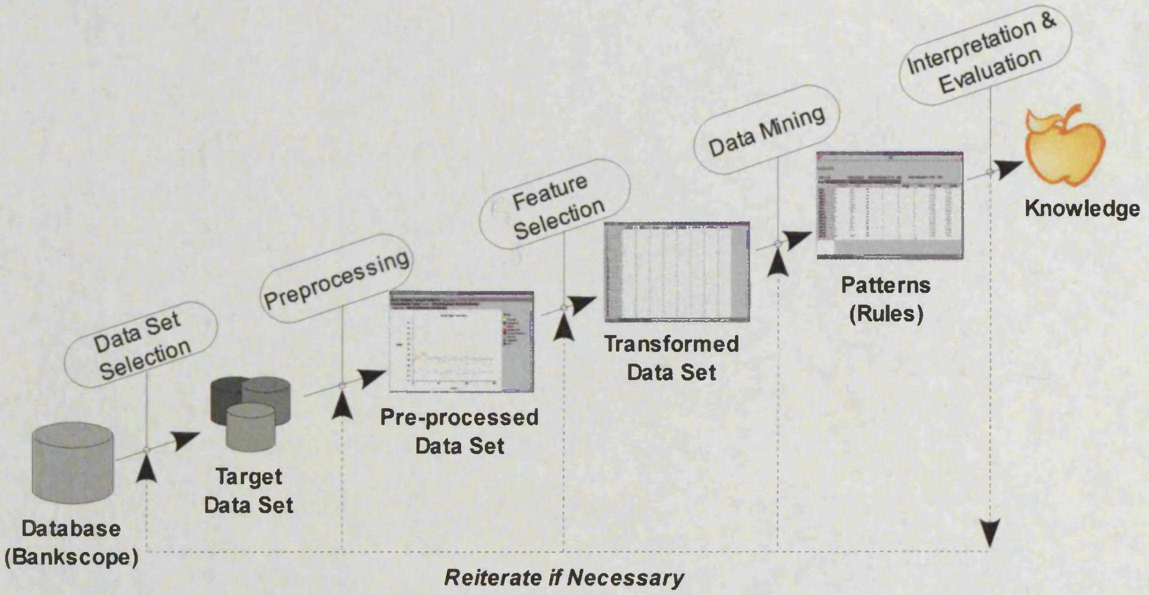


Figure 1.1.1: Stages of the KDD Process

The KDD process, as illustrated in Figure 1.1.1, is an interactive, iterative, process starting with

Data Set Selection. Whereby, a target data set is acquired from a data source. Typically, a subset is taken from a larger database. For example, the data set used within this dissertation is taken from Bureau van Dijk's online banking database Bankscope (2007), which gives the option to select subsets of their database of world banks (only European banks etc.). The data set selection stage may also entail, the initial selection of variables, required for the subsequent analysis.

Once the data set has been acquired, for many KDD problems, the next stage in the KDD process is the **Pre-processing** of the data (as a prerequisite for the later data mining stage). Operations of pre-processing include, methods for handling missing data, tackling imbalanced data, and discretisation of continuous valued data into discrete data (a requirement of some data mining methods, including RST). The aim of pre-processing is to improve the quality of the final discovered knowledge (the predictive performance and interpretability).

Often, data sets contain variables (also described as features or attributes), which are redundant or irrelevant, hence, as shown in Figure 1.1.1, a **Feature Selection** stage is required prior to data mining. Reduction of the data through feature selection can remove these variables (from those selected during the data set selection stage), allowing the subsequent data mining stage to be more efficient and effective at recognising meaningful patterns for the purpose of classification or prediction.

The first three stages of the KDD process, could be described as supportive processes for the main stage, that is, **Data Mining**. There are numerous data mining methods (Weiss and Kulikowski, 1991; Giudici, 2003; Kantardzic, 2003; Han and Kamber, 2006), and selection of a specific method is, dependant on the target application of the KDD process. The data mining method may be required either for classification or prediction. Typically, data mining methods used to predict categorical data are referred to as classification methods, whereas the prediction of continuous valued variables are referred to as numerical prediction, or simply prediction methods (Weiss and Indurkhya, 1998). From a research context, the data mining method may have already been selected, and the purpose of the knowledge discovery process is with a mind to understanding and testing the method rather than optimising its predictive capabilities (Fayyad et al., 1996). For the purpose of this dissertation the discovered knowledge, is referred to as the classifier.

The fifth and final stage of KDD, namely, **Interpretation and Evaluation**, involves testing the final classifier, to asses its potential predictive performance. The analyst may also, in some circumstances, attempt to interpret and rationalise the results. For example, whether or not a list of rules constructed through the data mining process makes logical sense. The analyst can also benefit here, through the statistical analysis of the results and graphical visualisations. The final step of the KDD process would be to apply the discovered knowledge to unseen data, in an attempt to make an accurate classification or prediction.

The following subsections describe: in subsection 1.1.1, VPRS as a modern alternative to the more established data mining methods, and in subsection 1.1.2, re-sampling and ensemble methods as, modern approaches to evaluation and classifier optimisation.

1.1.1 VPRS Within Data Mining

Developed by Pawlak (1982, 1991), Rough Set Theory (RST) is a set theoretical approach for dealing with imprecise or vague concepts within knowledge. With regards to data mining, it offered a new methodology for, construction and application of decision tables, based on inconsistent data sets (Ziarko, 2003). VPRS is an extension of RST, which relaxes an assumption that, the given classifications within the data set are totally correct (Mi et al., 2004). Hence, VPRS allows for a level of uncertainty, which may be inherent within some data sets (Ziarko, 1993a). This concept of uncertain classification, is relevant to the financial data that will be used here, namely, bank rating data (Tay et al., 2003).

RST based approaches, have intrinsic aspects that make them attractive as a modern analytical method (Ziarko, 2003; Beynon et al., 2004; Ilczuk and Wakulicz-Deja, 2007). RST is a nonparametric method, that makes no assumptions regarding the underlying distributions of the data variables, which is a limitation of other methods, such as regression (Dobson, 2001; Johnson and Wichern, 2007). RST, when utilised for classification, provides the user with a list of readable, interpretable, decision rules. The transparency and interpretability aspects that RST based methods offer (readable rules), are seen as key issues for the modern financial analyst (Harnett and Young, 2004, 2007). With regards to data mining methods in general, Olecka (2007, pp. 139) noted that, "...patterns must be, not only valid and understandable, but also explainable...", patterns being the discovered knowledge (classifier, rules etc.). Therefore, classifier methods providing readable lists of rules, are now becoming more favourable than, say, "black box" systems such as neural networks, whose method of knowledge representation (a network of weighted equations), is difficult to analyse (Lu et al., 1995; Craven and Shavlik, 1997; Lawrence et al., 1998; Roy, 2000).

With regards to feature selection (stage three of the KDD process shown in Figure 1.1.1), most data mining methods require feature selection to be performed prior to the data mining process, or use wrapper methods (described later) to perform feature selection during the process (Liu and Motoda, 2002). However, to some extent, feature selection is integral to RST based methods, including VPRS, through what is known respectively as reducts and β-reducts (described in Chapter 2). These are subsets of variables that maintain the data's semantics or 'meaning', whilst eliminating irrelevant or redundant variables from the data set. This semantic preserving element of feature selection is particularly desirable, as it facilitates the interpretability of the resulting knowledge (Jensen, 2004). However, for the work within this dissertation, some feature selection or "pre-feature selection" is incorporated for larger data sets (described in depth in Chapter 4).

1.1.2 Evaluation and Classifier Optimisation through Re-sampling and Ensemble Methods

As data mining methods have evolved, so have the techniques for evaluating the potential future performance of the constructed classifiers (Weiss and Kulikowski, 1991, Han and Kamber, 2006). With regards to classifier predictive accuracy, these methods are collectively referred to as re-sampling plans (Efron, 1982; Shao and Tu, 1995), and have been extensively used within regression, but until more recently, less so in machine learning. Breiman's (1996a) classification and regression trees perhaps being the exception. The most widely used methods of re-sampling are: leave-one-out, *k*-fold cross-validation and bootstrapping (Han and Kamber, 2006). Re-sampling is a repetitive process, that affectively finds an average predictive accuracy, based on variants of the classifier, constructed using different subsets of the data set from each repetition.

Additionally, and born out of the field of re-sampling, are ensemble methods (Dietterich, 2000a, 2000b). Whereas re-sampling methods have been developed to improve the evaluation stage of KDD, ensemble methods through re-sampling, have been developed to improve the data mining stage. They are designed to stabilise and optimise the process of classification and prediction through, in a sense, constructing an average or aggregated classifier from the classifiers constructed during the re-sampling process. The most notable ensemble method being Breiman's **b**ootstrap **aggregating** or bagging (Breiman, 1996a).

In terms of the position of this dissertation, there has been only limited research into re-sampling and ensemble methods with regards to RST (Bazan et al., 1994; Leifler, 2002; Jiang and Abidi, 2005; Stefanowski, 2004, 2007), and even less so within VPRS (Griffiths and Beynon, 2007, 2008). As stated, the developed software, with regards to the VPRS model, facilitates and implements a novel approach to evaluation, stabilisation and optimisation, through the application of re-sampling and ensemble methods, and in particular through a proposed method of β-reduct aggregation (described in depth in Chapter 3).

1.2 Development of the VPRS Decision Support Software

The software developed here, can be considered a specific type of KDD system, namely, a Decision Support System (DSS). Introduced in the early 1960's (Raymond, 1966; Turban, 1967; Keen, 1980), a DSS supports an expert or the analyst, with complex real world decision making. A DSS extracts inference or knowledge, from data, during the data mining process. They can also discover new relationships (patterns) within data that were, hitherto unknown (Weiss and Kulikowski, 1991).

It is important here, to draw a distinction with another type of KDD system, known as an expert system (Power 2002, 2008; Coppin, 2004), because of the poor reputation of expert system (Bell, 1985; Bachmann, 1993). The perceived role of expert systems was to reproduce an expert's reasoning and problem solving skills, but many of these systems met with limited success or outright failure (Bell, 1985; Bachmann, 1993). DSSs however, do not seek to replace the expert, but as their name suggests, support their decision making. DSSs have not attracted the unfavourable press which has marred the field of expert systems. Additionally here, the description of the developed software as a DSS, as opposed to an expert system, was received more favourably in the early discussions with Harnett and Young (2004), who expressed scepticism of expert systems.

The software developed within this dissertation, is envisaged to be a DSS, that supports an analyst's decision making, and allows them full autonomy throughout the KDD process. That is, the system will allow the analyst to make the final decisions, during each stage of the KDD process. An objective of the developed DSS, was to make it as user friendly as possible, providing an intuitive user interface that makes VPRS more accessible to the analyst, who may not be an expert in the field of VPRS (or rule based systems in general). The system was developed to empower the analyst, allowing them choice and flexibility to choose from a range of pre-processing, feature selection and evaluation methods; and setting certain parameters associated with those methods during the KDD process.

There are two main stages to the developed software. As shown in Figure 1.2.1, the first stage being a software package that incorporates the pre-processing and feature selection aspects of the KDD process, and a second stage, taking the direction of one of two separate VPRS software packages (one basic and one advanced, described next) that incorporates the data mining, interpretation and evaluation aspects of the KDD process.

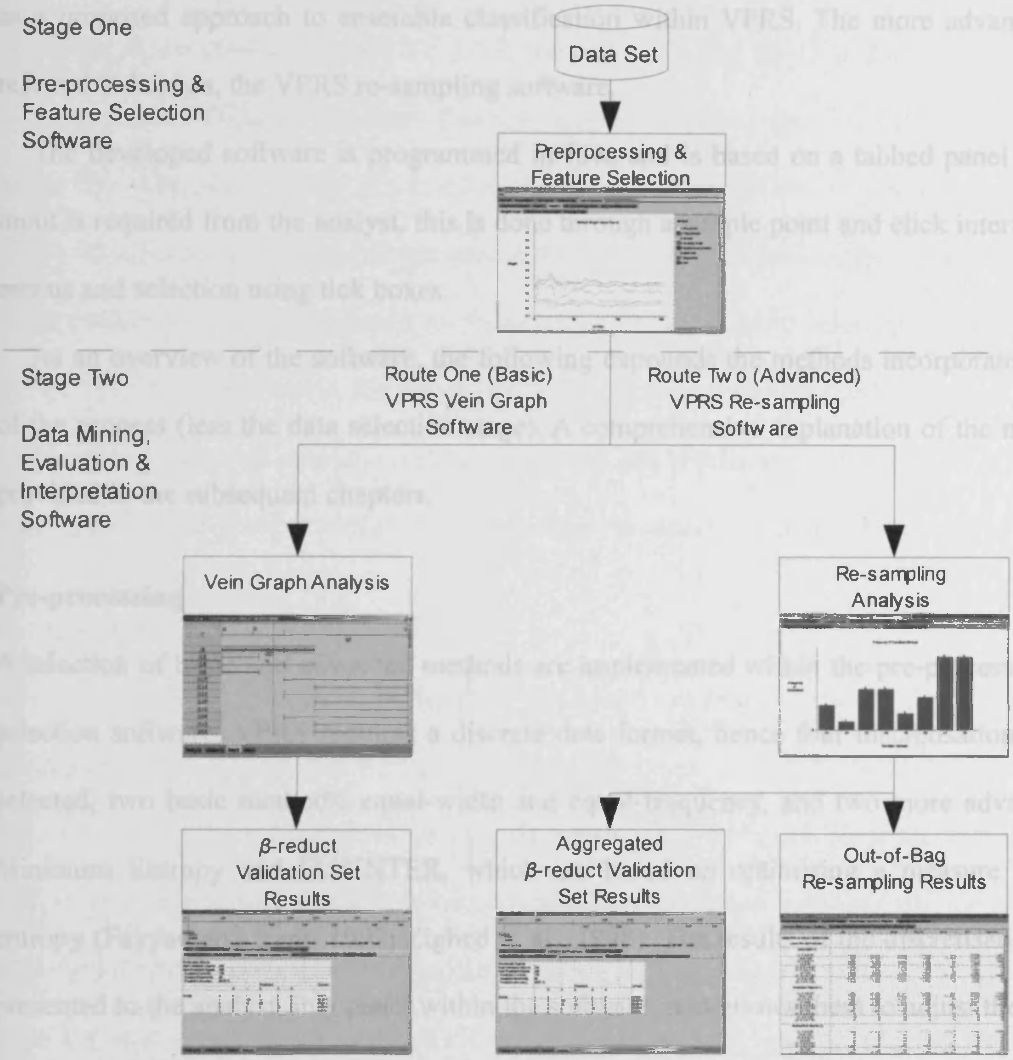

Stage One
Pre-processing & Feature Selection Software
Data Set
Preprocessing & Feature Selection
Stage Two
Data Mining, Evaluation & Interpretation Software
Route One (Basic) VPRS Vein Graph Softw are
Route Tw o (Advanced) VPRS Re-sampling Softw are
Vein Graph Analysis
Re-sampling Analysis
β-reduct Validation Set Results
Aggregated β-reduct Validation Set Results
Out-of-Bag Re-sampling Results

Figure 1.2.1: The Two Stages of the Developed Decision Support Software, Illustrating the Choice Between the Basic VPRS Vein Graph Analysis (Route One) and the More Advanced VPRS Re-sampling Analysis (Route Two)

As stated and illustrated in Figure 1.2.1, two versions of the VPRS analysis software have been developed, a basic package and a relatively more advanced package. The basic package, implements an interactive analysis approach based on the ideas surrounding the VPRS β-reduct vein graph[1] as presented in Beynon (2001), and is referred to here as, the VPRS vein graph software. The second, more advanced package, automates the β-reduct selection process, and incorporates some of the most recent developments in classifier evaluation, optimisation and stabilisation, namely, re-sampling and ensemble methodology. At the theoretical level, it introduces the aggregated β-reduct, as a proposed approach to ensemble classification within VPRS. The more advanced package is referred to here as, the VPRS re-sampling software.

The developed software is programmed in Java and is based on a tabbed panel system. Where input is required from the analyst, this is done through a simple point and click interface, pull-down menus and selection using tick boxes.

As an overview of the software, the following expounds the methods incorporated at each stage of the process (less the data selection stage). A comprehensive explanation of the methods will be provided in the subsequent chapters.

Pre-processing

A selection of basic and advanced methods are implemented within the pre-processing and feature selection software. VPRS requires a discrete data format, hence four discretisation methods were selected, two basic methods, equal-width and equal-frequency, and two more advanced methods, Minimum Entropy and FUSINTER, which are based on optimising a measure of information entropy (Fayyad and Irani, 1992; Zighed et al., 1998). The results of the discretisation process, are presented to the analyst, in a panel within the software, that allows them to adjust the discretisation, or select a different discretion method for specific variables if required.

The software also allows the analyst, the choice of three methods for handling imbalanced data sets (a data set that has a skewed classification distribution), namely up-balancing, down-balancing and average-balancing (Japkowicz, 2000). With regards to missing data, only two basic methods have been employed, they are, a simple mean imputation of missing values, and a k-fold nearest

[1] First referred to as "the vein graph" within this dissertation, previously referred to as Information Veins in Griffiths and Beynon (2008).

neighbour approach (methods for handling missing data are less developed than other pre-processing methods, Weiss and Indurkhya, 1998).

Feature selection

Although the developed VPRS software incorporates a level of feature selection through identifying β-reducts, it was found through the work undertaken during this dissertation, that a level of pre-feature selection (feature selection undertaken prior to identifying β-reducts) may be required for larger data sets. Here, two recent feature selection methods are implemented, ReliefF (Kononenko, 1994) which is similar to a k-nearest neighbour type approach, and a development of an RST based feature selection method proposed by Beynon (2004). The feature selection results are presented to the analyst in a table within the developed software, which allows them to choose the final set of variables for the subsequent data mining analysis. The results are augmented by a series of graphs which provide the analyst with further information to support their variable selection choice.

Data mining

The VPRS vein graph software will allow the analyst to select β-reducts using a novel point and click graphical interface, that is, the vein graph. The decision rules associated with the choice of β-reducts are derived, and displayed on a separate panel of information for inspection by the analyst.

The more advanced VPRS re-sampling software implements an original method for β-reduct aggregation. Whereby, using an automated approach, β-reducts are selected during the re-sampling process (which involves a number of repetitions), and are then selectable by the analyst for β-reduct aggregation. The analyst has the choice of which set of aggregated β-reducts they wish to select, based on evidence recorded during the re-sampling process, and presented to them within a number of information panels (described next under evaluation and interpretation).

The final product of β-reduct aggregation, is a list of sorted aggregated decision rules, where stable and potentially most useful decision rules, are the most prominent. The aggregated decision rules are presented with associated metrics, indicating properties such as, stability and potential predictive performance. The analyst has the final choice of aggregated rule selection through a simple point and click interface, where the analyst's decisions are supported by a substantial amount of information recorded and summarised during the re-sampling process.

Evaluation and interpretation

A number of evaluation methods have been implemented within the software. At the most basic level (VPRS vein graph software), the analyst has the choice to retain a certain percentage of the data as a validation set. They may either choose to use: stratified sampling that maintains the distribution of the classifications, or a novel method, more appropriate for imbalanced data sets, based on a statistical approach and introduced here for the first time.

The VPRS re-sampling software, through re-sampling, implements a more advanced scheme of evaluation. These re-sampling methods potentially provide better estimates of future predictive performance by, in a sense, finding average predictive accuracies, from a number of constructed classifiers. The results of re-sampling are presented in panels displaying information such as, re-sampling: predictive accuracies (also called out-of-bag estimates), quality of classifications, quality of approximation and so forth (described later), and a breakdown of rules associated with the selected β-reducts. A number of graphs are also provided to aid the analyst's decision making.

As mentioned, the software records information throughout the data mining process, and provides the analyst with summaries and breakdowns. Within the software, with regards to testing the classifier (set of rules), the predictions based on training and validation sets, are comprehensively broken down into a number of panels of information. Firstly, the predictions are broken down into, data the classifier can predict, and data the classifier must predict using an approach called nearest rule classification. Secondly, they are further broken down into, data predicted correctly and data predicted incorrectly. A method known as the confusion matrix, is used within these panels, which further elucidates the predictive performance across all possible decision classes. Finally, a full listing of all predicted data items (observations, objects) is given, presenting the analyst with information relating to each specific classification.

This section has outlined the developed VPRS software and the methods implemented within it, but it is only fair to acknowledge that there are a number of other existing RST based systems, developed within academia.

Most of the earlier systems concentrated on the basic RST model, whereas, the later systems incorporated extensions of the original RST (Peters and Skowron, 2007; Shen and Jensen, 2007). Amongst the most prominent systems is ROSETTA (Rough Sets Data Base, System, 2008), which supplies the analyst with a comprehensive set of tools to undertake an RST analysis. ROSETTA is not aimed at a particular application domain, but would require an analyst with knowledge of RST. That is, the analyst is more exposed to the underlying mathematics of RST. Whereas, the software developed here, seeks to remove the need for a deeper understanding of the mechanics of RST/VPRS, to accommodate analysts who are not experts in VPRS, or do not require the deeper more complex incite that ROSETTA would provide.

Also worth mentioning with regards to this dissertation is GROBIAN (Rough Sets Data Base, System, 2008), an earlier RST system (dating from 1996) that incorporated training and testing methodology (a methodology used here and descried in Chapter 3). A more recent software system known as JAMM, is based on another extension of the original RST, known as Dominance Based Rough Sets (DRSA). JAMM can be found at (http://idss.cs.put.poznan.pl/site/jamm.html)

Other RST software systems worthy of note are KDD-R, TRANCE and PRIMEROSE. A comprehensive list of RST based software systems, including those mentioned here, can be found online at the Rough Sets Data Base System (2008), which also provides references and a range of other information relating to RST.

1.3 Chapter Synopsis

The motivation, issues and objectives of this dissertation have been presented within this, the first chapter. This final section, provides a synopsis of the forthcoming chapters. The order of the chapters, does not directly reflect the order of the stages involved in the KDD process expressed in Figure 1.1.1, because Chapter 4, the chapter on data pre-processing and feature selection, requires for a specific feature selection method developed by Beynon (2004), an understanding of the RST theory laid out in Chapter 2.

Chapter 2, Rough Set Theory and Variable Precision Rough Sets Model

Chapter 2 formally introduces RST, including the notion of reducts, as a method of feature selection that is semantic preserving. Basic decision tables and rule generation are described.

The VPRS generalisation of RST is presented in full, with discussion on the β value as a level of probabilistic inclusion to deal with uncertainty, and β-reducts as the VPRS generalisation of a reduct. Decision tables providing minimal covering rules based on prime implicants are elucidated. The final section of this chapter describes the vein graph as a quantitative tool, that allows analysts, who are not experts in VPRS, to undertake a VPRS analysis of their data.

Chapter 3, Re-sampling, Ensemble Methods and Introduction to β-reduct Aggregation

Chapter 3 introduces methods for estimating the future predictive performance of a classifier, discussing the basic principles and methods, before describing, the more advanced re-sampling methods. Taking re-sampling as a foundation, it goes on to describe classifier ensembles as methods, for aggregating classifiers to improve stability and predictive performance. These methods are then adapted, for use within the VPRS model, and subsequent construction of aggregated β-reducts.

Chapter 4, Data Pre-processing and Feature Selection

Chapter 4 presents an overview of the methods implemented within this dissertation for the purpose of data pre-processing and feature selection. It describes the four methods of discretisation mentioned previously, namely, equal-width, equal-frequency, minimum entropy and FUSINTER; two methods for 'pre-feature' selection, that is, ReliefF and Beynon's (2004) RST based method; two methods for handling missing values, mean imputation and a *k*-nearest neighbour approach; and three methods for handling imbalanced data, up-balancing, down-balancing and average-balancing.

Chapter 5, Credit Ratings, Fitch Individual Bank Strength Ratings and Initial Data Selection

The bank rating classification problem is described in Chapter 5, and provides the application area for which the developed VPRS software is demonstrated. The initial sections of Chapter 5 will introduce the broader problem of predicting credit rating classifications and the need for modern analytical methods. It then focuses more on bank ratings, including the considered Fitch's Individual Bank Strength Ratings (FIBR) (Fitch, 2007). A review of studies relating to the prediction of bank rating classification, will form the basis for the analysis proposed in the later chapters. That is, the variable models utilised in those related studies, in particular the CAMELS model (Feldman et al., 2003; Derviz and Podpiera, 2004), are used for guidance on how to select the variables, and ultimately the data to be used in the subsequent analyses, shown within the following chapters.

Chapter 6, Introduction to Software, Pre-processing and Feature Selection Results

Chapter 6 introduces and demonstrates, the pre-processing element of the developed software. Describing in general, the technical layout and operation of the software. It provides the results of the software implementation of the pre-processing methods discussed in Chapter 4, applied to the FIBR data. It discusses issues relating to how discretisation and balancing affects the performance of the subsequent data mining analyses. The results of the feature selection methods, ReliefF and the RST based method, are also compared and discussed. The set of selected variables from the results presented in Chapter 6, are used subsequently in Chapters 7 and 8.

Chapter 7, Vein Graph Software Analysis of the Example and FIBR Data

For consistency with examples given in Chapter 2, and for demonstration purposes, Chapter 7 initially introduces the software implementation of the VPRS vein graph analysis on a small example data set. The remainder of the chapter presents a VPRS vein graph analysis of the FIBR data, discussed in Chapter 5. This chapter also illustrates the intuitive user interface (an interactive point and click vein graph), and the extensive range of interpretation and evaluation procedures implemented within the software, as mentioned in section 1.2.

Chapter 8, VPRS Re-sampling and Aggregated β-reduct Analysis of the FIBR Data

Chapter 8 presents the more advanced VPRS re-sampling analysis of the FIBR data, based on the three re-sampling methods described in Chapter 3. The results are compared with the more basic results reported in Chapter 7. Chapter 8, also investigates the impact to stability and predictive performance of the introduced β-reduct aggregation method, incorporated within the VPRS re-sampling software, and described in Chapter 4. It illustrates an extensive range of interpretation and evaluation methods. Additionally, for consistency with the VPRS vein graph software, it evaluates the aggregated β-reducts through a suite of evaluation methods common with the VPRS vein graph software. For the purpose of benchmarking, the final section of this chapter presents a number of re-sampling results, based on five benchmark data sets, and comparisons are also drawn against other classification methods.

Chapter 9, Conclusion and Future Work

The overall results of the FIBR analyses are discussed. The performance of the developed software as a DSS is summarised, for each stage of the KDD process implemented within the software, namely, preprocessing, feature selection, interpretation and evaluation. In particular, the suitability of the VPRS model as a data mining solution is discussed. Future work in terms of possible improvements to the system are presented, including adaptations to some of the methods implemented within the developed software, and consideration of other possible changes, in terms of theoretical directions within RST/VPRS, with regards to classification associated with the reality of real world data sets.

Chapter 2

Rough Set Theory and Variable Precision Rough Sets Model

Set theory, attributed to Cantor (1883), is considered by many mathematicians and philosophers, as one of the fundamental bases for modern day mathematics (Pawlak, 2005). It provides the rigour necessary for the integrity of mathematical concepts that must be clear and exact (Frege, 1903). Put simply, a set is a collection of objects or elements that are in some way related (Allenby, 1997). The relation between these elements may be stated mathematically or verbally, but the definition must be precise, in this sense, a set is said to be crisp or exact (Pawlak, 2004).

In an attempt to solve certain paradoxical statements associated with set theory (Potter, 2004), there have been a number of proposed extensions and indeed some alternative algebras (Pawlak, 2004). It should be acknowledged though, that set theory also has critics, including Bishop (1967) and Wittgenstein (2001).

More recently, applied mathematicians have looked at the application of set theory to "real world" applications, in particular, when dealing with the notions of vagueness or impression that may arise due to incomplete data sets or misclassified data (possibly due to subjective opinion). It was found that, for many situations, the notion of a crisp set although essential for the rigours of mathematics, is too constrictive (Zadeh, 1965; Pawlak, 1991; Ziarko, 1993a, 1993b). There are a number of fields where the ability to model imprecision is important (Tsumoto et al., 2004), particularly within computer science and relevant to this dissertation, namely the classification of objects (observations) within a data set, based on sets of descriptive attributes (variables).

A number of extensions to set theory have been proposed to deal with imprecision, most notably fuzzy set theory (Zadeh, 1965) and Rough Set Theory (RST) (Pawlak, 1982). Fuzzy set theory defines vagueness through graduated membership, that is, an element belongs to a set with degree k ($0 \leq k \leq 1$). RST defines vagueness through the notion of a boundary region, if the boundary region of an associated set is empty, then the set is said to be precise or crisp, else the set is described as imprecise or rough. RST is not an alternative to set theory, but rather "...embedded in it" (Pawlak, 2005, pp. 7), in essence RST is an extension to set theory, to allow for the description of vague concepts.

There are a number of properties of RST that have made it popular as a modern data mining method (Li and Wang, 2004). Here, the particular properties of interest are, data reduction through the selection of attribute[2] (variable) subsets, termed reducts, that maintain the information semantics (meaning) of the data (described later); and the ability to construct readable sets of "*If... then...*" decision rules. Furthermore, the non-parametric nature of RST (assuming no underlying distribution of the data), adheres closely to the principle of soft computing as stated by Düntsch and Gediga (1997, pp. 5), that is, "Let the data speak for itself".

However, in respect to classification problems, RST may in itself, be too restrictive. Specifically, in circumstances where objects may be indiscernible to any specific classification, because of insufficient or incomplete knowledge (Katten and Cooper, 1998). Ziarko (1993a, 1993b), proposed the Variable Precision Rough Set (VPRS) model as a generalisation of RST, that allows for a level of misclassification. Ziarko introduced the notion of partial or probabilistic classification based on a classification error β defined in the domain [0.0, 0.5), later An et al. (1996) re-defined β as a

[2] The word "variable", is more commonly used within data mining, but within RST, the word "attribute" is used more frequently. Hence, within this chapter variables are referred to as attributes for consistency with the extant literature (further described in section 2.1).

probabilistic measure of majority inclusion in the domain (0.5, 1.0] (this definition of β is adopted within this dissertation). Furthermore, Ziarko (1993a, 1993b) introduced the β-reduct, which extended the notion of a reduct, to deal with data reduction with regards to the partial classification of objects within VPRS.

The selection of a suitable β value (Ziarko, 1993a, 1993b; Ziarko and Xiao, 2004; Li et al., 2007), has received less attention than the discovery of reducts (or β-reducts). Beynon (2001) considered a visual representation of certain aspects of VPRS, aimed towards aiding the analyst in understanding the interim analysis. Based on a visualisation of the β-reducts over their respective domains of β, it aids the analyst in the selection of the most suitable β-reduct according to their specific application requirements. This visual representation has been titled here the "vein graph", as the lines resemble veins, and because of the connotations with veins of knowledge within a mountain of data, an analogy used by Linoff (1998). The vein graph is one of the core elements of the VPRS decision support software developed within this dissertation.

It should be acknowledged, VPRS is not the only extension to RST, and that other RST based methodologies have been developed (Shen and Jensen, 2007), to tackle, other challenges relating to real world data sets. Greco et al.'s (2005) Dominance based Rough Set Approach (DRSA), has recently been proposed to deal with the ordinal properties of data, and enhances RST by providing more informative decision rules (Chapter 9 subsection 9.2.2, further discusses DRSA). There have also been a number of approaches to extending RST to incorporate the ideas of Bayes' Theorem (Wong and Ziarko, 1986; Ślęzak and Ziarko, 2003; Ślęzak, 2004; Goldstein and Wooff, 2007). However, the elements of Bayes' theorem described later in subsection 2.2.3, do not reflect the usage of Bayes' theorem as reported in the referenced papers, because here, in contrast to those referenced papers, no assumptions are made with regards to the underlying data distribution as a priori to analysing the data (Ziarko, 2003; Pawlak, 2005). Another promising direction involves combining rough and fuzzy sets, Pawlak (2005) notes that the theories complement each other and that their hybridisation has proved successful in many applications (Pal and Skowron, 1999; Jensen 2004; Jensen and Shen, 2008).

The following sections of this chapter, introduce the core theory and mathematics required for this dissertation, as described below:

- Section 2.1. **Rough Set Theory and Decision Tables**. This section describes the basic mathematics of RST, including, attribute dependency and the notion of a reduct. The theory is conveyed through the exposition of a decision table containing an example data set.
- Section 2.2. **Variable Precision Rough Set Model**. This section describes the VPRS extension to RST, including the notion of β as a measure of majority inclusion and a β-reduct as the extension to a reduct. Again the theory is demonstrated through the continued use of the example data set given in the previous section.
- Section 2.3. **The Vein Graph**. This section provides a full exposition of the vein graph as introduced by Beynon (2001). Based on VPRS, the vein graph enables the analyst to visualise identified β-reducts over different domains of β.
- Section 2.4. **Summary**. This section summarises the main points of the previous sections, and briefly describes their implementation with the developed VPRS vein graph software.

2.1 Rough Set Theory and Decision Tables

Central to RST, and many other data mining methods, is the notion of an information system (Pawlak, 1991, 2004, 2005), which is defined as the universe of objects (observations) U $\{o_1, o_2,...\}$, characterised by the set of attributes A $\{a_1, a_2,...\}$. More appropriate here, is a specialised case of the information system known as the decision table, which further defines the universe of objects U as being characterised by the set of condition attributes $C = \{c_1, c_2,...\}$ and classified to the set of decision attributes $D = \{d_1\}$ (here only one decision attribute is considered), where $A = C \cup D$.

Hence, a decision table can be denoted as $S = (U, C, D)$. The value of an attribute associated with a particular object is referred to as an attribute value (Beynon and Peel, 2001; Ślęzak, 2004).

Table 2.1.1 illustrates an example data set, represented as a decision table containing seven objects $\{o_1, o_2, o_3, o_4, o_5, o_6, o_7\}$, characterised by six condition attributes $\{c_1, c_2, c_3, c_4, c_5, c_{6,}\}$, and classified by the decision attribute $\{d_1\}$ (example taken from Beynon, 2001). Table 2.1.1 could represent seven patients (objects), being categorized as either ill ($d_1 = 0$) or healthy ($d_1 = 1$), where for example, the condition attribute c_4 may represent temperature (has temperature $c_4 = 1$, and does not have temperature $c_4 = 0$). The attribute values, are of a discrete data type, taking values of only 1 or 0, hence this data would be described as having a low level of granularity (Ślęzak et al., 2005a, 2005b). This example data set will be referred to throughout this chapter for demonstration purposes (and Chapters, 4 and 6).

Objects	c_1	c_2	c_3	c_4	c_5	c_6	d_1
o_1	1	1	1	1	1	1	0
o_2	1	0	1	0	1	1	0
o_3	0	0	1	1	0	0	0
o_4	1	1	1	0	0	1	1
o_5	1	0	1	0	1	1	1
o_6	0	0	0	1	1	0	1
o_7	1	0	1	0	1	1	1

Table 2.1.1: Example Data Set

The sets of condition classes $E(C) = \{X_1,\dots, X_i: i = |E(C)|\}$ and decision classes $E(D) = \{Z_1,\dots, Z_j: j = |E(D)|\}$ within RST, are each defined as the set of equivalence classes of objects that are indiscernible over the condition attributes and decision attribute, respectively. Where the equivalence relation $E(\cdot)$ is referred to as the indiscernibility relationship. From Table 2.1.1, the equivalence classes associated with the set of condition classes $E(C)$ are:

$$X_1 = \{o_1\}, X_2 = \{o_2, o_5, o_7\}, X_3 = \{o_3\}, X_4 = \{o_4\}, X_5 = \{o_6\},$$

and the equivalence classes associated with the set of decision classes $E(D)$ are:

$$Z_1 = \{o_1, o_2, o_3\}, Z_2 = \{o_4, o_5, o_6, o_7\}.$$

As stated in the introduction to this chapter, RST was developed as a method for handling vague or imprecise concepts within knowledge. Central to the theory, is the notion of approximating an equivalence class, with equivalence classes defined over a separate disjoint set of attributes. Of particular interest here, is the approximation of the decision classes by the condition classes, using what is termed the lower and upper approximations denoted as $\underline{B}$ and $\overline{B}$, defined here as:

$$\underline{B}(Z_j) = \cup_{\forall X_i \subseteq Z_j}\{X_i : X_i \in E(C)\}, \tag{2.1.1}$$

$$\overline{B}(Z_j) = \cup_{\forall X_i \cap Z_j \neq \emptyset}\{X_i : X_i \in E(C)\}. \tag{2.1.2}$$

Where the lower approximation defines, the subset of objects in Z_j which are certainly defined as objects belonging to Z_j, based on the condition classes X_i. The upper approximation contains those objects in Z_j which are either totally or partially defined as objects belonging to Z_j, based on the condition classes X_i. Additionally, the boundary region defines those objects which are only partially defined as objects belonging to Z_j, with respect to the condition classes X_i, and denoted as:

$$BN(Z_j) = \overline{B}(Z_j) - \underline{B}(Z_j). \tag{2.1.3}$$

An empty boundary region, or $BN(Z_i) = \emptyset$, indicates that the set Z_j is precise or exact with respect to X_i, otherwise if there are objects within the boundary region it is described as imprecise or rough.

Following on from the example data set, and demonstrating with the decision classes Z_1 and Z_2, then:

$$\underline{B}(Z_1) = \{o_1, o_3\}, \overline{B}(Z_1) = \{o_1, o_2, o_3, o_5, o_7\}, BN(Z_1) = \{o_2, o_5, o_7\}.$$
$$\underline{B}(Z_2) = \{o_4, o_6\}, \overline{B}(Z_2) = \{o_2, o_4, o_5, o_6, o_7\}, BN(Z_2) = \{o_2, o_5, o_7\}.$$

Clearly there are objects within the boundary regions $BN(Z_1)$ and $BN(Z_2)$, hence the decision classes Z_1 and Z_2, can be considered imprecise or rough with regards to the set of condition classes $E(C)$.

2.1.1 Rule Generation and Attribute Dependency

Each row within a decision table, such as Table 2.1.1, based on the concomitant attribute values of the condition attributes *C*, describes a rule with a decision outcome d_1. The rules can be described in a simple "*If... then...*" format. Taking for example, row two within decision Table 2.1.1, the rule based on the object o_2, can be interpreted as: "*If* $c_1 = 1, c_2 = 0, c_3 = 1, c_4 = 0, c_5 = 1$ *and* $c_6 = 1$ *then* $d_1 = 0$". Describing decision rule construction more formally; for the decision table $S = (U, C, D)$, a decision rule may be determined from object o_n an element of *U*, as a sequence $c_1(o_n),\ldots,c_{|C|}(o_n) \rightarrow d(o_n)$, this is called the decision rule induced by o_n in *S* and denoted, in short, as $C_{o_n} \rightarrow D_{o_n}$. The set of all possible rules, based on objects within Table 2.1.1, would be termed the maximal rule set (An et al., 1996; Ziarko and Xiao, 2004). For many applications, where a level of rule interpretability is important, the maximal rule set may not be practical (particularly if the rule set is large), subsection 2.2.3 describes a generalisation of the maximal rule set, that increases interpretability.

Clearly, when comparing objects o_2 and o_5 within Table 2.1.1, they have the same condition attribute values, but different decision values, the rules induced by these objects are termed inconsistent (Pawlak, 2004). Conversely, objects o_5 and o_7 have the same condition attribute values and the same decision value, hence the rules induced by these objects are termed consistent.

The number of consistent rules represented within a decision table, can be used as a measure of the consistency and more formally known here as, the attribute dependency between the decision attributes *D* and the condition attributes *C*, where *D* is said to depend on *C* in degree $\gamma(C,D)$ $(0 \leqslant \gamma(C,D) \leqslant 1)$. The value $\gamma(C,D)$ is referred to as the quality of classification (QoC) (Beynon, 2001), and calculated as:

$$\gamma(C,\, D) = \frac{card(\cup_{\forall Z_j \in E(D)} \underline{B}(Z_j))}{card(U)}. \qquad (2.1.1.1)$$

The QoC for the example data set within Table 2.1.1, is calculated as:

$$\begin{aligned}\gamma(C,\ D) &= \frac{card(\underline{B}(Z_1)\cup\underline{B}(Z_2))}{card(U)},\\ &= \frac{card(\{o_1,\ o_3\}\cup\{o_4,\ o_6\})}{card(o_1,\ o_2,\ o_3,\ o_4,\ o_5,\ o_6,\ o_7)},\\ &= \frac{4}{7} = 0.571 \text{ (to 3 d.p.).}\end{aligned}$$

A value of $\gamma(C,\ D) < 1$ indicates that D is partially dependent on C and that the decision table is inconsistent, whereas, $\gamma(C,\ D) = 1$ indicates a total dependency and a consistent decision table. Hence, there is a level of inconsistency associated with the example data set in Table 2.1.1, since $\gamma(C,\ D) = 0.571 < 1$.

2.1.2 Reducts

As mentioned in Chapter 1, it is often desirable to reduce the number of attributes to remove irrelevant or redundant attributes, this process is referred to as feature selection (see Chapter 4 section 4.2 for more detail). One of the disadvantages of many feature selection algorithms is their inability to preserve what is termed the semantics of the data (Jensen, 2004). Here, with regards to RST, the semantics of the data are preserved by maintaining the dependency between the condition and decision attributes, that is, by insuring the calculated value of QoC is the same for the feature subset of attributes as the full set of attributes (Beynon, 2001; Pawlak, 1991, 2004, 2005). Within RST, a feature subset of attributes that has the same QoC as the full set of attributes, is referred to as a reduct. Pawlak (2004, pp. 16) describes a reduct as:

> "... a reduct is the minimal subset of attributes that enables the same classification of elements of the universe as the whole set of attributes. In other words, attributes that do not belong to a reduct are superfluous with regard to classification of elements of the universe."

More formally, if $P \subseteq C$, then P is a reduct of C if the equation $\gamma(P,\ D) = \gamma(C, D)$ holds true, and no subset $\hat{P} \subseteq P$ allows $\gamma(\hat{P},\ D) = \gamma(C, D)$. Put simply, P is a reduct of C if the QoC associated with P is equal to the QoC associated with C, and no subset of P can achieve the same

QoC.

Using the example data set in Table 2.1.1, if $P = \{c_1, c_2, c_5\}$ then the condition classes of $E(P)$ are, $P_1 = \{o_1\}$, $P_2 = \{o_2, o_5, o_7\}$, $P_3 = \{o_3\}$, $P_4 = \{o_4\}$ and $P_5 = \{o_6\}$, clearly, these are the same condition classes associated with the full set of condition attributes C, as calculated previously (see section 2.1). Hence, $\gamma(P, D) = \gamma(C, D)$, that is, $\gamma(P, D) = 0.571$, and no proper subset of P has a QoC equal to 0.571, so P is a reduct of C (for illustration, the decision table associated with P is shown in Table 2.1.2.1).

Objects	c_1	c_2	c_5	d_1
o_1	1	1	1	0
o_2	1	0	1	0
o_3	0	0	0	0
o_4	1	1	0	1
o_5	1	0	1	1
o_6	0	0	1	1
o_7	1	0	1	1

Table 2.1.2.1: Decision Table Based on Reduct $P=\{c_1, c_2, c_5\}$

Computation of the reducts associated with a decision table is commonly undertaken through the use of a discernibility matrix (Pawlak, 1991; Ziarko and Xiao, 2004), but this can be a computational intensive task (depending on the data set). Indeed, Komorowski et al. (1999) state that, the number of reducts associated with an information system, with number of attributes m, is of the order $\binom{m}{\lfloor m/2 \rfloor}$, additionally Skowron and Rauszer (1992) noted that computation of all reducts is a NP-complete problem, hence indicating the non-trivial nature of reduct discovery.

There are however a number of time efficient heuristic methods (thus potentially non-optimal) for reduct discovery, which are more time efficient, such as genetic algorithms (Wroblewski, 1995; 1998). Recently, Wang et al. (2005) utilised the particle swarm algorithm (Kennedy and Eberhart, 1995) for reduct identification, citing that unlike other evolutionary algorithms such as genetic algorithms, it does not require complex operators, but remains as effective. Other reduct discovery algorithms seek only to find a single reduct, as this is often all that is required (Jensen, 2004). Notable among these algorithms are, the QuickReduct algorithm (Chouchoulas and Shen, 2001) and the ReverseReduct algorithm (Chouchoulas et al., 2002). Jaganathan et al. (2006) presented a hybrid system, combining what they termed the Enhanced QuickReduct and the ant colony optimisation algorithm, for the purpose of data pre-processing. For other heuristic and suboptimal approaches, see Lin and Yin (2004) and Susmaga (2004).

2.2 Variable Precision Rough Set Model

As mentioned within Chapter 1, there are often circumstances, particularly with regards to real world data, where objects have been misclassified, possibly due to a level of subjectivity within an expert's decision making. An example pertinent to this dissertation, being bank ratings (Pasiouras et al., 2006), other examples can be found in bio-informatics and medical diagnosis (Adibi et al., 1993; Laita et al., 2001; Beynon and Buchanan, 2003; Rudnicki and Komorowski, 2004; Widz et al., 2004).

Ziarko (1993a, pp. 39), whilst discussing RST, commented that its, "...inability to model uncertain information..." was frequently emphasized by people attempting to utilise RST for analyses, and that, "This limitation severely reduces the applicability of the rough set approach...". RST assumes that the objects classified within the given data set are correctly classified (Ziarko, 1993a; Mi et al., 2003). Furthermore, Ziarko noted that, RST assumes that all objects within the universe U of a data set, are known, and that any conclusion based on a model derived from that data set, is only truly applicable, with a full level of certainty, to that data set. In reality, the set of available objects only represents a sample of the universe U, from which more general conclusions must be derived, for application to a larger population of objects.

The Variable Precision Rough Set (VPRS) model was introduced by Ziarko (1993a, 1993b) as an extension to RST, which allowed for a level of misclassification under uncertain reasoning, and is more applicable to real world data (Dembczyński et al., 2007). As mentioned, there are other extensions to RST such as Dominance based Rough Set Approach (DRSA) (Greco et al., 2005; also see for descriptions of other extensions, Shen and Jensen, 2007; Jensen and Shen, 2008). Indeed, Dembczyński et al. (2007), bring together VPRS and DRSA, to handle, what they termed, excessive inconsistency within real world data for which their DRSA method, could not cope. Mi et al. (2004), based on Beynon's (2001) work, illustrated within VPRS, what they termed conflicting rules. Whereby, β-reducts could be identified, that did not maintain the distribution of the condition classes within the original system (whole data set), thus not preserving the semantics of the data set, as was originally intended (see subsection 2.1.2). Hence, they describe a system that seeks to maintain the distribution of the condition classes associated with the β-reducts (further considered in Chapter 9 section 9.2.2).

Central to the VPRS extension of RST, was the generalisation of the set inclusion relationship (associated with equation 2.1.1), to a majority inclusion relationship, by means of introducing a probabilistic value β in the range (0.5, 1.0] (An et al., 1996), and re-defining the RST upper, lower and boundary approximations.

More formally, considering the proportion of the objects within the condition class $X_i \in E(C)$, that are also associated with a decision class $Z_j \in E(D)$, if that proportion is greater than or equal to the pre-defined β value, then the condition class X_i is said to be in, what is now termed, the β-positive region of Z_j. The β-positive region of Z_j, for the set of condition attributes C, is defined as:

$$\mathrm{POS}_C^\beta(Z_j) = \bigcup_{Pr(Z_j|X_i) \geqslant \beta} \{X_i : X_i \in E(C)\}. \tag{2.2.1}$$

Conversely, the β-negative region is defined as those objects belonging to the condition classes whose proportion within the associated decision class Z_i is less than or equal to $1-\beta$. Formally:

$$\mathrm{NEG}_C^\beta(Z_j) = \bigcup_{Pr(Z_j|X_i) \lessdot 1-\beta} \{X_i : X_i \in E(C)\}. \tag{2.2.2}$$

The β-negative region can also be considered as the set of the condition classes which can be classified to the complement of Z_j, namely $\neg Z_j$, with the proportion of objects greater than or equal to β, that is $\text{POS}_\beta(\neg Z_j) = \text{NEG}_\beta(Z_j)$.

The boundary region, now re-defined as the β-boundary region, refers to those condition classes whose proportion within the associated decision class Z_i is less than β but greater than $1-\beta$. That is, the set of the condition classes that belongs neither to the decision class Z_j, nor its complement $\neg Z_j$, with certainty greater than β. The β-boundary region is formally defined as:

$$\text{BND}_C^\beta(Z_j) = \bigcup_{1-\beta<Pr(Z_j|X_i)<\beta} \{X_i : X_i \in E(C)\}. \tag{2.2.3}$$

Note that, a β value equal to unity would result in the β-positive region coinciding with the lower approximation as defined in RST (see Equation 2.1.1). The upper approximation would coincide with the union of the β-positive and the β-boundary regions (see Equation 2.1.2), and the β-negative region would be the complement of the upper approximation (see Equation 2.1.3). For further reference, Beynon (2003) describes a graphical representation of the relationship between RST and VPRS.

For the example data set shown in Table 2.1.1, the β-positive, β-negative and β-boundary regions are shown, considered with, $\beta = 0.8$, and $Z_2 = \{o_4, o_5, o_6, o_7\}$, then:

$$\begin{aligned}
\text{POS}_C^{0.8}(Z_2) &= \{o_4\} \cup \{o_6\} = \{o_4, o_6\}, \\
\text{NEG}_C^{0.8}(Z_2) &= \{o_1\} \cup \{o_3\} = \{o_1, o_3\}, \\
\text{BND}_C^{0.8}(Z_2) &= \{o_2, o_5, o_7\},
\end{aligned}$$

and for $\beta = 0.55$, then:

$$\begin{aligned}
\text{POS}_C^{0.55}(Z_2) &= \{o_2, o_5, o_7\} \cup \{o_4\} \cup \{o_6\} = \{o_2, o_4, o_5, o_6, o_7\}, \\
\text{NEG}_C^{0.55}(Z_2) &= \{o_1\} \cup \{o_3\} = \{o_1, o_3\}, \\
\text{BND}_C^{0.55}(Z_2) &= \emptyset .
\end{aligned}$$

By considering a lower β value of 0.55 (an acceptable lower level of majority inclusion compared to $\beta = 0.8$), the set $\{o_2, o_5, o_7\}$ moves from the β-boundary region into the β-positive region. Clearly, the value of β has an affect upon the QoC, hence, with respect to VPRS, the QoC must be re-defined to incorporate the β value (compare with Equation 2.1.1.1), and is formally denoted as (Beynon, 2001):

$$\gamma^{\beta}(C, D) = \frac{card(\cup_{\forall Z_j \in E(D)} \mathrm{POS}_C^{\beta}(Z_j))}{card(U)}. \qquad (2.2.4)$$

Moreover, there is an inverse relationship between the QoC and the β value. That is, a lower β value allows for a higher level of classification (majority inclusion). In regards to the example, with $\beta = 0.8$, $\gamma^{0.8}(C, D) = 0.571$ but with $\beta = 0.55$, $\gamma^{0.55}(C, D) = 1$. This inverse relationship raises the question, what is the most appropriate β value (Beynon, 2001; Mi et al., 2004)? Indeed, it may cause a dilemma for any analyst, who intends on using a decision table to induce a set of decision rules, who must decide between:

- Selection of a relatively low value of β, that allows for a high QoC, but also infers a possible high level of misclassification.

or

- Selection of a relatively high value of β, which allows for a low QoC, but with a lower level of misclassification.

Before considering the selection of β, a measure of the accuracy of the rules induced during VPRS can also be calculated. Following Katzberg and Ziarko (1996), and Pawlak (1991), here, the accuracy of the whole rule set is considered, hence, the number of objects given a correct classification, out of those objects given a classification, is defined as:

$$\alpha^{\beta}(C, D) = \frac{card(\cup_{\forall X_i, Z_j}\{X_i \cap Z_j : (card(X_i \cap Z_j)/card(X_i)) \geqslant \beta\})}{card(\cup_{\forall Z_j \in E(D)} \mathrm{POS}_C^{\beta}(Z_j))}. \qquad (2.2.5)$$

Here, this expression is referred to as the Quality of Approximation (QoA) (Katzberg and Ziarko, 1996). Considering the example data set in Table 2.1.1 and $\beta = 0.8$, then:

$$\begin{aligned}\alpha^{0.8}(C,\, D) &= \frac{card(\,(X_1 \cap Z_1) \cup (X_1 \cap Z_2) \ldots \cup (X_5 \cap Z_2) : (card(X_i \cap Z_j) / card(X_i)) \geqslant 0.8\}\,)}{card(\mathrm{POS}_C^{0.8}(Z_1) \cup \mathrm{POS}_C^{0.8}(Z_2))}, \\ &= \frac{card(\,\{o_1\} \cup \{o_3\} \cup \{o_4\} \cup \{o_6\})}{card(\{o_1,\, o_3\} \cup \{o_4,\, o_6\})}, \\ &= \frac{4}{4} = 1.\end{aligned}$$

and for $\beta = 0.55$, then:

$$\begin{aligned}\alpha^{0.55}(C,\, D) &= \frac{card((X_1 \cap Z_1) \cup (X_1 \cap Z_2) \ldots \cup (X_5 \cap Z_2) : (card(X_i \cap Z_j) / card(X_i)) \geqslant 0.55)}{card(\mathrm{POS}_C^{0.55}(Z_1) \cup \mathrm{POS}_C^{0.55}(Z_2))}, \\ &= \frac{card(\{o_1\} \cup \{o_5,\, o_7\} \cup \{o_3\} \cup \{o_4\} \cup \{o_6\})}{card(\{o_1,\, o_3\} \cup \{o_2,\, o_4,\, o_5,\, o_6,\, o_7\})}, \\ &= \frac{6}{7} = 0.857 \text{ (to 3 d.p.).}\end{aligned}$$

A value of $\alpha^{0.8}(C,\, D) = 1$ indicates a QoA of 100% accuracy, and $\alpha^{0.55}(C,\, D) = 0.857$ indicates a QoA of 85.7% accuracy. Only rules associated with condition classes within the β-positive regions are considered (hence the union of the β-positive regions within the denominator in Equation 2.2.5), because rules induced within VPRS (more specifically, within the work undertaken here), are based on the prime implicants (described in subsection 2.2.3), which are induced from the β-positive region (An et al., 1996; Ziarko and Xiao, 2004).

2.2.1 The Issue of β Value Selection

Recently, Li et al. (2007) suggested that selection of the β value may be dependant upon the specific application, subsequently requiring an expert to select the most appropriate value. They also proposed that β may be learned through the feature selection process (β-reduct identification and selection, described in subsection 2.2.2), although they give no details, one may assume this involves a simple experimental approach. However, it will be shown later in this dissertation, that their proposals may be an oversimplification of a non-trivial decision, involving a number of factors.

Ziarko and Xiao (2004) utilises a decision matrix as a means to find the β-reducts within a data set provided by a car insurance company. The decision matrix is based on the indiscernibility matrix used in RST to find reducts (Pawlak 1991). They calculated the β value, by first assessing the prior probability of a customer (object) not having an accident (from the dichotomous decision class 'has accident' with attribute values 'yes' or 'no'), and assuming that any rules (constructed from a condition class) within +/–5% of that probability value are significant. They proceed to set asymmetric β values (described next), β-upper and β-lower, based on the probability value +/–5%, respectively.

Further research in VPRS, by Katzberg and Ziarko (1994, 1996), has relaxed the symmetric bond between the β-positive and β-negative regions (previously using identical values of β), by allowing for different levels of β, that is, the aforementioned β-upper (u) and β-lower (l) values; to be assigned to the β-positive and β-negative regions, respectively. Known as the extended VPRS model, it allows the analyst more control over the classification of the condition classes (further termed the $\text{VPRS}_{l,u}$ in Beynon, 2003). The introduction of asymmetric β values (l and u) adds a further dimension to the difficult issue of, selecting the most appropriate β value(s). Succinctly stated by Ziarko (1999, pp. 467), while discussing decision making within probabilistic decision tables, the setting of the precision control parameters l and u is:

> "...an optimisation problem connected to the external knowledge of possible gains and losses associated with correct, or incorrect predictions..."

and suggests an approach based on game theory for finding the optimal parameter values for l and u. Beynon (2003) presented an in-depth study of the effects of β-upper and β-lower selection on what is termed the (l, u) QoC and the (l, u) degree of dependency, through the introduction of a visualisation technique named the (l, u)-graph which offers an intelligent approach to the selection of what they termed (l, u)-reducts.

However, the original VPRS model is considered here, without loss of generality to its development. The following sections 2.2.2 and 2.2.3, are concerned with the β-positive region. That is, rules will be induced based from the prime implicants of a given data set (described in subsection

2.2.3). The process of finding prime implicants and constructing a rule set from them, is associated with the β-positive region (An et al., 1996; Ziarko and Xiao, 2004), hence negating the requirement to consider asymmetric β values for the work undertaken here.

Within the original VPRS framework, Ziarko (1993a, 1993b) suggested setting β to a threshold value, that allowed any set Z_j to be what they termed β-discernible. They described Z_j as being β-discernible if it has an empty β-boundary region for the selected value of β. For the example data set (Table 2.1.1), and based on the set of condition classes $X_i \in E(C)$, then the threshold value for $X_2 = \{o_2, o_{5,}, o_6\}$, with regards to $Z_2 = \{o_4, o_5, o_6, o_7\}$, is 0.667 (to 3 d.p.). Hence, setting a β value greater than 0.667 would prohibit X_2 from being classified within the β-positive region. In contrast, setting β to be any real value in the range (0.5, 0.667] would permit X_2 to be in the β-positive region, it follows, X_2 would be β-discernible within the given range. They further describe those condition classes not associated to any decision class for any value β as absolutely rough, whereas those that do belong to a decision class for a range of β as relatively rough.

The selection and representation of β within restricted ranges, with regards to what is termed β-reducts was considered by Beynon (2001), and is one of the main theoretical focuses of this dissertation and described more thoroughly in section 2.3.

2.2.2 Exposition of β-reducts

The concept of a reduct, introduced in the previous subsection 2.1.2, is central to applications utilising RST as a feature selection method for semantic preserving (Kuo and Yajima, 2003; Jensen and Shen, 2004; Li et al., 2004; Mi et al., 2004). The VPRS extension of the reduct, termed here the β-reduct, allows for the reduction of data through the selection of a subset of features (attributes) that preserves the semantics of the data under a level of misclassification. Ziarko (1993a) formally describes β-reducts, where $P \subseteq C$ is the considered β-reduct, as having two associated properties:

1. $\gamma^{\beta}(P, D) = \gamma^{\beta}(C, D)$.

2. No proper subset $\hat{P} \subset P$, subject to the same β value can also give the same quality of classification, that is $\gamma^{\beta}(\hat{P}, D) \neq \gamma^{\beta}(P, D) \; \forall \hat{P} \subset P$.

Following from the example data in Table 2.1.1, Table 2.2.2.1 displays a subset $P = \{c_3, c_6\}$ of the condition attributes C, that satisfy the properties for being considered a β-reduct, with $\beta = 0.55$. That is, there are no proper subsets of $\{c_3, c_6\}$ that can be considered β-reducts, and $\gamma^{0.55}(P, D) = \gamma^{0.55}(C, D)$.

Objects	c_3	c_6	d	Condition Class E($\{c_3, c_6\}$) = $\{P_1, P_2, P_3\}$
o_1	1	1	0	$P_1 = \{o_1, o_2, o_4, o_5, o_7\}$, Pr($Z_1 \mid P_1$) = 0.4, Pr($Z_2 \mid P_1$) = 0.6
o_2	1	1	0	
o_3	1	0	0	$P_2 = \{o_3\}$, Pr($Z_1 \mid P_2$) = 1.0
o_4	1	1	1	
o_5	1	1	1	$P_3 = \{o_6\}$, Pr($Z_1 \mid P_3$) = 1.0
o_6	0	0	1	
o_7	1	1	1	

Table 2.2.2.1: Decision Table and Condition Classes of β-reduct $\{c_3, c_6\}$

The lowest majority inclusion proportion with respect to $\{c_3, c_6\}$, is 0.6, and is associated with the condition class P_1. Clearly, the value $\beta = 0.55$ is less than 0.6, hence within the allowable range for which $\{c_3, c_6\}$ can be considered a β-reduct, indeed β could have been set to any value in the range (0.5, 0.6]. Beynon (2001) discussed a novel approach of representing the range of values associated with a system of β-reducts through the use of what we have termed the vein graph (for reasons mentioned previously), that aids the user in the selection of the most appropriate value of β and selection of the most appropriate β-reduct for their particular application. The next section describes a full exposition of the vein graph applied to the example data set.

Beynon (2001) also suggests that there is no specifically prescribed method for the selection of β-reducts, and that a full derivation is a non-trivial computational problem. As mentioned previously, there are heuristics and suboptimal solutions for finding reducts, Ślezak and Wróblewski (2003) extends the genetic algorithm approach, to find what they termed approximate entropy reducts. Typically though, approaches such as those mentioned and that described in Ziarko and

Xiao (2004), require pre-selected β values, which is not appropriate to the system exposited within this dissertation (in particular the vein graph described in section 2.3). As here, the focus is based on calculating the range of β values for all possible β-reducts, between a generally lower β value (towards 0.5) and their threshold values, thus, the β value is not required to be pre-calculated.

The β-reduct generation approach adopted here, is simply based on those two properties of a β-reduct mentioned previously and outlined in Ziarko (1993a, 1993b). Put simply, the power set **P**(C) is constructed (all possible β-reducts), then by iteratively selecting each set from the power set, and calculating all possible threshold values associated with that particular set, the possible ranges of β for which it could be considered a β-reduct (where $\gamma^{\beta}(C, D) = \gamma^{\beta}(P, D)$) were recorded. Finally, for each range of β for which the subset is possibly a β-reduct, it was compared with all previously recorded β-reducts to asses if any subsets of the current superset were β-reducts over the same range (point 2 in Ziarko's, 1993a, description of a valid β-reduct). Where there was an overlap between ranges of β, the range of the superset was curtailed not to include the range where the subset was recorded as a β-reduct. This approach is better understood through a visualization of the process, as such, the figures presented in section 2.3 of this chapter should aid the reader.

2.2.3 Prime Implicants and Rule Metrics

Once the β-positive region has been identified, that is, the condition classes that belong to a decision class to at least degree β, a procedure known as the decision matrix can be used to identify the prime implicants, as a step towards, constructing a minimal covering rule set (An et al., 1996; Ziarko and Xiao, 2004). The prime implicants are the specific condition attribute values that uniquely distinguish each of the condition classes associated with the β-positive region. Table 2.2.3.1 displays the process for calculating the prime implicants associated with β-reduct $\{c_3, c_6\}$.

			P_1		P_2		P_3	
			$c_3 = 1$	$c_6 = 1$	$c_3 = 1$	$c_6 = 0$	$c_3 = 0$	$c_6 = 0$
P_1	$c_3 = 1$	$c_6 = 1$	-	-	-	0	0	0
P_2	$c_3 = 1$	$c_6 = 0$	-	1	-	-	0	-
P_3	$c_3 = 0$	$c_6 = 0$	1	1	1	-	-	-
Prime Implicants				1	1	0	0	

Table 2.2.3.1: Prime Implicants of the Condition Classes X_1, X_2 and X_3, Associated with β-reduct $\{c_3, c_6\}$

Within Table 2.2.3.1, the values in the grey shaded area, identify the condition attribute values that distinguish between each of the three condition classes $P_1 = \{o_1, o_2, o_4, o_5, o_7\}$, $P_2 = \{o_3\}$ and $P_2 = \{o_6\}$ shown in Table 2.2.2.1. For example, the condition attribute values $c_3 = 1$ and $c_6 = 1$, associated with the condition class with the column headed P_1, differs to the condition attribute values $c_3 = 1$ and $c_6 = 0$ associated with the condition class in the row headed P_2, by the condition attribute c_6. That is, for the condition classes P_1 and P_2, the condition attribute values for c_3 are equal (record as a dash within the table), but the condition attributes values for c_6 differ. Hence, the cell relative to the condition attribute c_6 and those condition classes, has the value 1 recorded in it (highlighted in bold within the grey shaded area), to indicate the condition attribute and the specific value which differentiates the condition classes P_1 and P_2. The prime implicants displayed at the bottom of the table are the values that uniquely distinguish between the condition classes given in the column headings, and all other condition classes given in the row headings. Here for example, P_1 is uniquely distinguished from P_2 and P_3 by $c_6 = 1$.

Based on the prime implicants, a set of minimal covering rules can be constructed (described next). Each rule is based on a single condition class and decision class value, from those condition classes associated with the β-positive region. Table 2.2.3.2 presents the minimal covering rule set associated with β-reduct $\{c_3, c_6\}$, in an "*If... then...*" format, a number of rule metrics, Support, Correct, Strength and Certainty, are also shown, and are described next.

Rule		c_3		c_6		d_1	Support	Correct	Strength	Certainty
$r_1(P_1, Z_2)$	If	-	and	1	then	1	5	3	0.714	0.6
$r_2(P_2, Z_1)$	If	1	and	0	then	0	1	1	0.143	1.0
$r_3(P_3, Z_2)$	If	0	and	-	then	1	1	1	0.143	1.0

Table 2.2.3.2: "*If... then...*" Minimal Covering Rule Table with Concomitant Metric Values, for β-reduct $\{c_3, c_6\}$

The rule index within Table 2.2.3.2, is shown with the associated condition and decision class in brackets (e.g. $r_1(P_1, Z_2)$). They are described as minimal covering rules, because, no two rules have equivalent condition and decision attribute values associated with them, hence a "minimal" rule set. Additionally, objects classified by the rules are not necessarily classified correctly, hence "minimal covering" rule set. This is in contrast to the maximal deterministic rule set, with regards to RST described in subsection 2.1.1 (for further description of a maximal deterministic rule set see Ziarko, 1998). Within Table 2.2.3.2, a dash "-" represents the fact that the condition attribute value associated with the rule is irrelevant, based on the prime implicant information taken from Table 2.2.3.1.

There are a number of metrics (calculated values), specifically probabilistic metrics, that can be defined with regards to the minimal covering rule set. Based on Bayes' theorem (Pawlak, 2004; Beynon, 2001; Ziarko and Xiao, 2004), these metrics allow the analyst to infer certain information about a specific rule, such as, the proportion of the objects within the universe it covers (strength), or the probability of it being classified correctly (certainty).

The basic measure by which the other probabilistic measures are calculated, is called support (Equation 2.2.3.1). The support value represents the number of objects given a classification (correctly or incorrectly) by a given decision rule $r_k(X_i, Z_j)$ ($k \leqslant n$, the number of objects). Support is calculated as:

$$\text{Support}^{\beta}(r_k(X_i, Z_j)) = card(X_i). \qquad (2.2.3.1)$$

Equation 2.2.3.2 defines, given a rule $r_k(X_i, Z_j)$, the number of objects that would be correctly classified, from the condition class associated with $r_k(X_i, Z_j)$, and calculated as:

$$\text{Correct}^{\beta}(r_k(X_i,\ Z_j)) = card(X_i \cap Z_j). \tag{2.2.3.2}$$

The rule strength (Equation 2.2.3.3), is a probabilistic value, and defines the proportion of the objects within the universe of objects (a data set), given a classification by the given rule $r_k(X_i, Z_j)$. Strength is calculated as:

$$\text{Strength}^{\beta}(r_k(X_i,\ Z_j)) = \frac{\text{Support}^{\beta}(r_k(X_i,\ Z_j))}{card(U)}. \tag{2.2.3.3}$$

Rule certainty (Equation 2.2.3.4), is also a probabilistic value, and defines the proportion of the objects from the condition class associated with a given rule $r_k(X_i, Z_j)$, that are classified correctly. Certainty is calculated as:

$$\text{Certainty}^{\beta}(r_k(X_i,\ Z_j)) = \frac{\text{Correct}^{\beta}(r_k(X_i,\ Z_j))}{\text{Support}^{\beta}(r_k(X_i,\ Z_j))}. \tag{2.2.3.4}$$

Below, are example metric calculations for $r_1(P_1, Z_2)$ (from the minimal covering rule Table 2.2.3.2), associated with β-reduct $\{c_3, c_6\}$, where, $P_1 = \{\text{o}_1, o_2, o_4, o_5, o_7\}$, and $Z_2 = \{o_4, o_5, o_6, o_7\}$ and $\beta = 0.55$:

$$\begin{aligned}
\text{Support}^{0.55}(r_2(P_1,\ Z_2)) &= card(\{o_1,\ o_2,\ o_4,\ o_5,\ o_7\}) = 5,\\
\text{Correct}^{0.55}(r_2(P_1,\ Z_2)) &= card(\{o_1,\ o_2,\ o_4,\ o_5,\ o_7\} \cap \{o_4,\ o_5,\ o_6,\ o_7\}) = 3,\\
\text{Strength}^{0.55}(r_2(P_1,\ Z_2)) &= \frac{5}{7} = 0.714,\\
\text{Certainty}^{0.55}(\{c_2,\ c_5\}, D) &= \frac{3}{5} = 0.6.
\end{aligned}$$

Ziarko (1998, 2001, 2003) and Ziarko and Xiao (2004), discussed the importance of probabilistic decision rules in depth. With regards to the choice of VPRS over RST, Ziarko (1998, pp. 179) states in general terms, that:

> "...most of the practical problems occurring in machine learning, pattern recognition and data mining inter-data relationships are probabilistic in nature, leading to non-deterministic decision tables."

The minimal covering rule table, can be considered a probabilistic non-deterministic rule table, as described by *ibid.* Whereas, the maximal rule table defined with regards to RST is a deterministic rule table (also see Ziarko 2001, for further definitions). Essentially, Ziarko is proposing that the probabilistic non-deterministic rule table associated with VPRS is, of more practical use, than the deterministic rule table associated with RST.

2.3 The Vein Graph

Allowing the analyst to select a β value and consequently a β-reduct appropriate to their application requirements (predictive accuracy, interpretability etc.), is a consideration of the software developed within this dissertation. This section describes Beynon's (2001) approach to visualizing β-reducts, namely through the vein graph. This approach was developed and implemented within the VPRS vein graph software, to facilitate the selection, by an analyst, of β-reducts over the range of β.

As has been demonstrated in the previous subsection 2.2.2, for any given β-reduct, it is associated with a range of β values in which β can be chosen, and which satisfies those properties of a β-reduct given in subsection 2.2.2. In respect to this Ziarko (1993a, 1993b), proposed two useful propositions relating to ranges of β:

Proposition 3.1. If a condition class X_i is given a classification with $0.5 < \beta \leqslant 1.0$, then X_i is also discernible at any level $0.5 < \beta_1 \leqslant \beta$.

Proposition 3.2. If a condition class X_i is not given a classification within $0.5 < \beta \leqslant 1.0$, then X_i is also indiscernible at any level $\beta < \beta_1 \leqslant 1.0$.

In the light of these propositions, Beynon (2001, pp. 596) noted that:

> "These statements of Ziarko indicate some move to the exposition of the role of ranges of β rather than specific β values."

Motivated by this observation, Beynon introduced a novel method for visualising ranges of β associated with specific β-reducts, targeted in part, for the requirement of modern data analysis tools for quantitative analysts. Here, we have come to know this as the vein graph (shown later in Figure 2.3.1), but before a full explanation is given, further discussion of β-reducts and threshold values is required.

As described previously, for any relatively rough condition class X_i, there is an associated threshold value greater than 0.5 for which β may take, that allows X_i to be discerned to a decision class Z_j. Let the β threshold value for any given X_i be denoted as β_{thd_i}, hence any value $\beta \leqslant \beta_{thd_i}$ also allows X_i to be discernible. The lowest of the β_{thd_i} values for any given set of condition classes is defined as $\beta_{min_i} = min_i \beta_{thd_i}$. Hence, through the selection of β, only those condition classes X_i, where $\beta_{thd_i} \geqslant \beta$, are discerned to a decision class. To summarise, those condition classes X_i whose β_{thd_i} values are in the interval $[\beta_{min_i}, \beta)$, are not discernible based on the majority inclusion rule, whereas those condition classes whose β_{thd_i} are in the range $[\beta, 1)$ are discernible. A direct consequence of this is that the QoC for the set of condition classes is less than unity if $\beta_{min_i} < \beta$.

Table 2.3.1 displays all the β-reducts associated with the example data set show in Table 2.1.1, including the ranges for which they are valid, and their associated QoC.

β-reduct index	Condition Values	β range	QoC
β-reduct 1	$\{c_1, c_3\}$	(0.5, 0.6]	1
β-reduct 2	$\{c_3, c_6\}$	(0.5, 0.6]	1
β-reduct 3	$\{c_4\}$	(0.5, 0.667]	1
β-reduct 4	$\{c_2, c_5\}$	(0.5, 0.667 | 0.75]	1
β-reduct 5	$\{c_4\}$	(0.667, 0.75]	0.57
β-reduct 6	$\{c_1, c_2, c_3\}$	(0.667, 1]	0.57
β-reduct 7	$\{c_2, c_3, c_5\}$	(0.667, 1]	0.57
β-reduct 8	$\{c_2, c_5, c_6\}$	(0.667, 1]	0.57
β-reduct 9	$\{c_1, c_4, c_5\}$	(0.75, 1]	0.57
β-reduct 10	$\{c_2, c_3, c_4\}$	(0.75, 1]	0.57
β-reduct 11	$\{c_2, c_4, c_5\}$	(0.75, 1]	0.57
β-reduct 12	$\{c_3, c_4, c_5\}$	(0.75, 1]	0.57
β-reduct 13	$\{c_4, c_5, c_6\}$	(0.75, 1]	0.57

Table 2.3.1: All β-reducts Associated with the Example Data Set

Referring to β-reduct $\{c_3, c_6\}$, as shown in Table 2.3.1 (see subsection 2.2.2 for associated calculations), its QoC was $\gamma^{0.55}(P, D) = 1$, that is, $\beta < \beta_{min,}$ (i.e. 0.55 < 0.6). The allowable range of β associated with β-reduct $\{c_3, c_6\}$ is (0.5, 0.6], which is a subinterval of the β-range (0.5, 0.667] associated with the full set of attributes C, for which $\gamma^{\beta}(C, D) = 1$. If we define the allowable range of β, as requiring the same QoC associated with C as β^{γ}, then β^{1} = (0.5, 0.667]. Beynon (2001) describes those β-reducts whose β range is contained in β^{γ}, as being restricted β-reducts, and those whose β range is equal to or contains β^{γ}, as being unrestricted β-reducts. So the example β-reduct $\{c_3, c_6\}$, would be considered a restricted β-reduct. β-reduct $\{c_1, c_2, c_3\}$, is an example of an unrestricted β-reduct associated with the example data set (calculations not shown), with allowable range of β in (0.667, 1.0], with concomitant $\gamma^{\beta}(P, D) = 0.57$, where C is associated with $\beta^{0.57} = (0.667,\ 1.0]$. Note, as a consequence of point 2 of Ziarko's definition of a valid β-reduct, β-reduct $\{c_1, c_2, c_3\}$ is not a valid reduct in the range (0.5, 0.667], because β-reduct $\{c_1, c_3\}$ is a subset of $\{c_1, c_2, c_3\}$, and is a valid β-reduct over the range (0.5, 0.667].

Note that, within Table 2.3.1, the β range of β-reduct $\{c_2, c_5\}$, is recorded as (0.5, 0.667 | 0.75]. The value to the right of the "|" line 0.75, denotes the upper limit on what is termed the hidden or extended β-reduct (to be discussed next). The β-reduct $\{c_4\}$ appears twice as β-reduct 3 and β-reduct 5, because the β-reduct based on the single condition attribute c_4 is a β-reduct over two separate ranges of β, associated with two separate levels of QoC. Within Table 2.3.1, there are two unrestricted β-reducts in the β^{1} range, namely β-reducts 3 and 4, and there are three unrestricted β-reducts in the $\beta^{0.57}$ range, namely β-reducts 6, 7 and 8.

Figure 2.3.1 is Beynon's (2001) vein graph visualization of the information presented within Table 2.3.1. The β subintervals and their concomitant QoCs, associated with C, are illustrated on the top line of the graph in Figure 2.3.1. The β-reducts (shown on the left) and their concomitant ranges of β are illustrated below the C line. As stated within the introduction, the method was entitled the vein graph, as the lines are analogous to veins of knowledge within the data (Linoff, 1998).

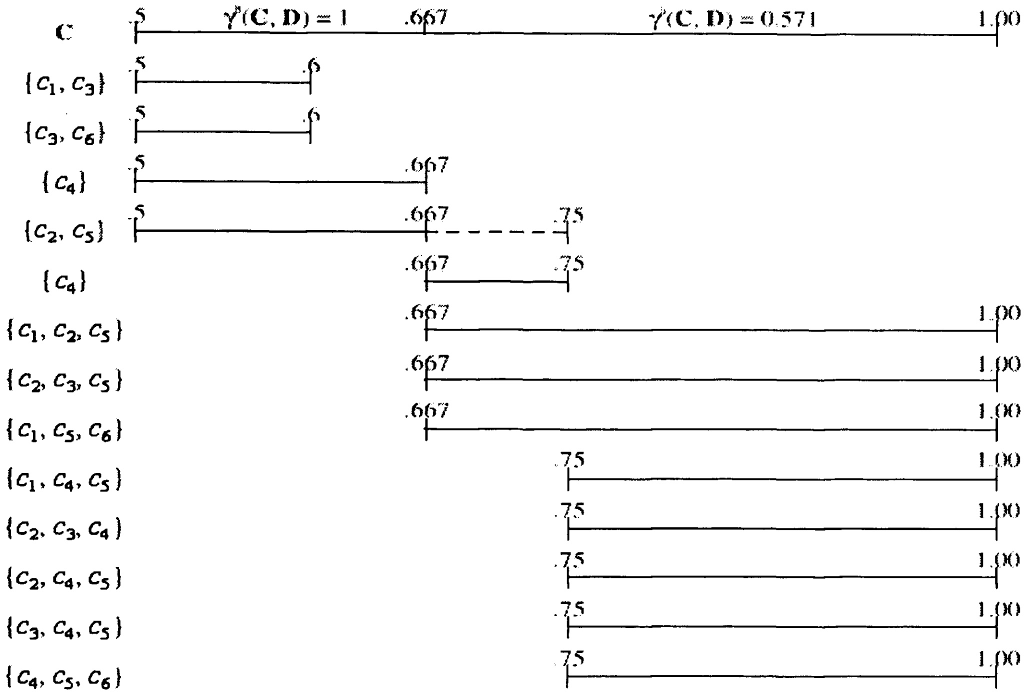


Figure 2.3.1: Vein Graph for the Example Data Set

Within Figure 2.3.1 (taken from Beynon, 2001), there are four β-reducts associated with a QoC $\gamma^{\beta}(C, D) = 1.0$ (i.e. in the range β^{1}), and nine associated with the range $\beta^{0.57}$. The unrestricted β-reducts are clearly visible as those β-reducts having the same QoC over the same range of β as the full set of attributes C, where the $\gamma^{\beta}(C, D)$ values are shown on the sub ranges β = (0.5, 0.667], $\gamma^{\beta}(C, D) = 1.0$; and β = (0.667, 1.0], $\gamma^{\beta}(C, D) = 0.571$. It can also be seen that the two β-reducts relating to $\{c_4\}$, are associated with different levels of QoC over separate sub-domains of β. The graph emphasises the important connection between the selection of the β value and the concomitant QoC.

Looking more closely at β-reduct $\{c_2, c_5\}$, a portion of the line is dashed between 0.667 and 0.75. The value 0.75 is the calculated upper bound or β_{min_i} threshold value associated with the β reduct. However, for the β reduct to have been selected as a valid β-reduct, the analyst would have had to have chosen a β value in the range (0.5, 0.67] (i.e. to have the same QoC as the full set of condition attributes C). Beynon (2001) terms this β value in the range (0.5, 0.67], as the external β value that is imposed without any independent calculations, except knowledge of the QoC associated with the full set of condition attributes C. Furthermore, based on a selection of the external β value in that range, the β-reduct $\{c_2, c_5\}$ would be found, but the actual level of confidence associated with the β-reduct is in fact above the upper allowable β value range (0.5, 0.67], that is 0.75. This extended range is referred to as, the internal β value associated with the β-reduct, and β-reducts such $\{c_2, c_5\}$, which exhibit this extended range or level of confidence are referred to as, extended β-reducts.

Additionally, with a pre-selected β value in the range (0.667, 0.1], the β-reduct $\{c_2, c_5\}$ would not have been identified, hence in these circumstances it is referred to as a hidden β-reduct. Beynon (2001) considers that supplying the analyst with this extra information allows them more choice in their decision making. Moreover, if an analyst has a specific QoC in mind, then they could be provided with the β values and β-reducts that achieve that priori, but they would only be provided with the external β-reducts that do not consider the internal levels of confidence, that would be associated with any calculated β-reduct. Hence, presenting the analyst with a graph such as the vein graph, provides them with a full exposition of information concerned with the choice of β-reducts available.

The developed VPRS vein graph software, implements the vein graph as a novel "point and click" system, that allows an analyst to select β-reducts from the vein graph. Based on the selected β-reduct, the software then automates the process of constructing the minimal covering rules from the prime implicants. The constructed rule set is applied to training and validation data (described in the following chapter), to evaluate the possible future performance of the selected β-reduct, allowing the analyst to compare between the β-reducts presented within the vein graph. A full elucidation of the VPRS vein graph software is presented in Chapter 7, through the analysis of the FIBR data, and the example data set presented in this chapter.

2.4 Summary

This chapter formally introduced Rough Set Theory (RST) in section 2.1, and in particular, its generalisation, Variable Precision Rough Set (VPRS) model in section 2.2, as a tool for the mathematical analyses of imprecise or vague concepts. The specific area of application has been predictive analysis, formulated around the notion of decision tables, and also incorporating data reduction (feature selection) through the utilisation of β-reducts (in VPRS).

Section 2.3 introduced and emphasised the vein graph as a visual aid for the analyst. It provides them with information from which they may select the most appropriate majority inclusion value β, and hence, identify and subsequently select the most appropriate β-reduct. It has been shown, within VPRS, that selection of the β value may be complicated by factors such as hidden or extended β-reducts that would be disregarded under other circumstances.

A major focus of this dissertation is to provide a front ended software environment which allows an analyst, who is not an expert in VPRS, to undertake VPRS analyses. One core feature of the developed software is the implementation of the vein graph, as an interactive software application, that allows for the simple "point and click" selection of β-reducts from the graph. Following selection of a β-reduct, the analyst will be provided with the minimal covering rule decision table, which is then evaluated using the methods described within the following chapter. Chapter 7, presents the VPRS vein graph analyses of the example data referred to throughout this chapter (Table 2.1.1), and the FIBR data described in Chapter 5.

Chapter 3

Re-sampling, Ensemble Methods and β-reduct Aggregation

With regards to classification problems, constructing a classifier from a data set, for the purpose of future classification of unclassified objects, is essentially the primary aim of data mining (Larose, 2005). Within data-mining, it is crucial to be able to evaluate the constructed classifier, to facilitate benchmarking against other competitor methods, and to give the analyst an indication of future performance of the classifier (Weiss and Kulikowski, 1991; Giudici, 2003).

A number of factors such as, computation time, scalability and interpretability should be considered with regards to a classifier's future performance. However, predictive accuracy, as a diagnostic measure, can be considered as the ubiquitous indicator of a classifier's future performance (Kantardzic, 2003). Put simply, the predictive accuracy of a classifier is calculated as, the proportion of correctly classified objects from the total number of objects given a classification (Weiss and Indurkhya, 1998). The data set upon which the predictive accuracy is calculated, can greatly affect the calculated value. Moreover, the predictive accuracy is actually only the estimated predictive accuracy of the true predictive accuracy, since, it is only possible to know the true value by testing the classifier on all possible objects within the universe. For many data mining applications, based on the data available, this cannot be achieved, and so, only an estimate of the true predictive accuracy is possible.

Attempts to accurately estimate the true predictive accuracy, with particular regards to statistical models and sub-sampling, have been studied since the early twentieth century (Larson, 1931; Horst, 1941). In the latter half of the twentieth century, more advanced methods of estimation, using cross-validation were developed, to tackle, what Stone (1974, pp. 111) referred to as the, "...shrinkage phenomenon..." (described later). These cross-validation methods are now more broadly known as re-sampling methods and have been extensively studied within the statistics community (Hills, 1966; Duda and Hart, 1973; Stone, 1974; Efron, 1982, 1983; Shao and Tu, 1995).

More recently, and born out of the philosophy of re-sampling, "ensemble" methods have been developed to stabilise and optimise classifiers, attempting to improve their predictive accuracy (Han and Kamber, 2006). Ensembles or aggregated classifiers, utilise re-sampling methods in a sense, to construct an "average classifier" (Hothorn and Lausen, 2005). Giudici (2003) notes that, in some respects this approach is considered similar to Bayesian model-averaging methods (Dietterich, 2000b; Brooks et al., 2003; Green et al., 2003). Notable among the ensemble methods is bagging or **bootstrap aggregating**. Introduced by Breiman (1996a), it is based on the bootstrap re-sampling method (Efron, 1979, 1982, 2003), and has been applied to many different classifier methods, within various spheres of application (Horthorn and Lausen, 2003a; Huang et al., 2004a; Li et al., 2006; Kotsianti and Kanellopoulos, 2007).

Re-sampling within RST has received limited attention, where dynamic reducts are perhaps the most notable application (Bazan et al., 1994; Leifler, 2002; Jiang and Abidi, 2005), which attempts to find the most stable reduct from a data set (does not use re-sampling as a method for estimating predictive accuracy). Aggregation within RST, has received even less attention, with perhaps the only notable work being Stefanowski (2004, 2007), who followed Breimen's (1996a) bagging method, but applied it to their rule construction method based on RST. With regards to VPRS, which to some extent is still a nascent development on RST, there does not appear to be any re-sampling or ensemble related material published so far (other than that presented by Griffiths and Beynon, 2007, 2008).

The following sections within this chapter outline sub-sampling and re-sampling as methods to estimate the predictive accuracy; ensemble methods to optimise and stabilise a classifier; and β-reduct aggregation as a novel approach to implementing ensemble methods, within a VPRS framework. The proceeding sections are briefly described below:

- Section 3.1. **Sub-sampling**. Introduces the simple train and test methodology for estimating the predictive accuracy.
- Section 3.2. **Additional Classifier Performance Considerations**. Covers some additional aspects of estimating the predictive accuracy, such as, bias and variance. It also presents the confusion matrix, as a method for determining the predictive accuracy of individual decision classes.
- Section 3.3. **Re-sampling Methods**. Reviews re-sampling methods utilised for estimating predictive accuracy. Mainly concentrating on the three most established re-sampling methods, namely, leave-one-out, *k*-fold cross-validation and bootstrapping.
- Section 3.4. **Ensemble Methods**. Reviews the recent developments in ensemble and classifier aggregation methods, with a full exposition given to bagging.
- Section 3.5. **VPRS Re-sampling and β-reduct Aggregation**. Presents a novel method for implementing ensemble methodology within VPRS and introduces the aggregated β-reduct.
- Section 3.6. **Additional Classification Issues within VPRS**. Describes the process of classification through, either a β-reduct, or an aggregated β-reduct, and describes the nearest rule methodology, for classifying objects that cannot be classified by a matching rule (a rule that has matching condition attribute values).
- Section 3.7. **Summary**. Briefly summarises the topics and developments within this chapter.

3.1 Sub-sampling

Perhaps the most naïve approach, to estimating the true predictive accuracy of a classifier, is to test the classifier on the data that was used to construct it. Known as the apparent or re-substitution predictive accuracy,[3] for smaller data sets it typically provides a misleading or over-optimistic result (Weiss and Kulikowski, 1991; Nolan, 1997; Kantardzic, 2003). Figure 3.1.1 illustrates the process of calculating the apparent predictive accuracy.

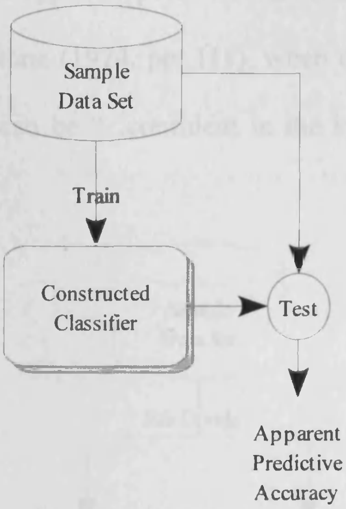


Figure 3.1.1: Calculation of the Apparent Predictive Accuracy

The apparent predictive accuracy is typically over-optimistic, because as shown in Figure 3.1.1, by training and testing on the same data set, the estimated predictive accuracy does not take into account objects, not represented within the data set. It is this over-optimistic estimation, that Stone (1974), was referring to when describing the shrinkage phenomenon (quoted in the introduction to this chapter).

The earliest work to overcome this over-optimistic estimate, employed the simple train and test methodology to calculate an unbiased estimate of predictive accuracy (Larson, 1931; Horst, 1941; Simon, 1971).

[3] The related literature more commonly talks in terms of the apparent error rate, where the error rate is the proportion of incorrectly classified objects over the total number of objects given a classification. Here, to avoid confusion we are only referring to performance in terms of predictive accuracy (the complement of the error rate), as it is more widely used within the RST related literature.

3.1.1 Simple Train and Test Methodology

In his work on successful marriages, Horst (1941), recognised what he called the drop in predictability between the original sample, which he used to construct his classifier, and an independent check sample. This is an early instance of the simple train and test methodology, the process is illustrated in Figure 3.1.1.1.

Essentially, the early work on estimating predictive accuracy, indicated that, to avoid the over-optimistic value indicative of the apparent predictive accuracy, the analyst ideally had to test the classifier on "unseen" objects. Stone (1974, pp. 111), when commentating on this approach, notes that the statistician who uses it can be "...confident in the knowledge that the set-aside data will deliver an unbiased judgement...".

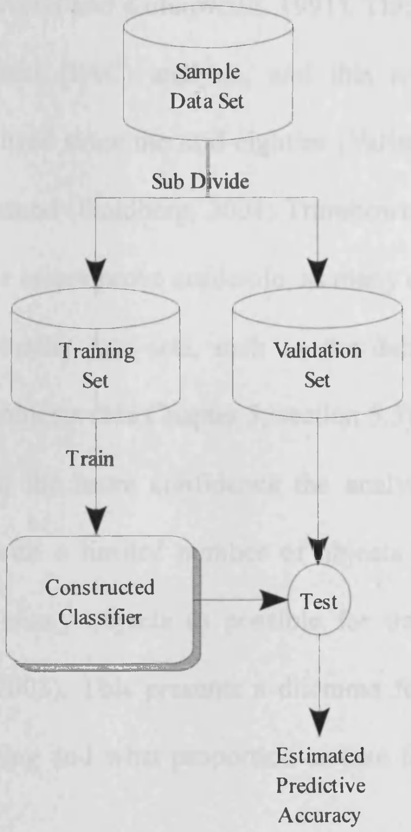


Figure 3.1.1.1: Calculating an Unbiased, Train and Test Estimate of the Predictive Accuracy

Describing the simple train and test methodology more formally, and illustrated by Figure 3.1.1.1. A subset of the sample data, known as the training set, is taken at random and used to train the classifier, the predictive accuracy is then estimated on the remainder of the data set, termed the validation set. The key associated issue here, being, what size validation set is required for a good estimate of the true predictive accuracy?

Analyses have shown that the estimated predictive accuracy is, in fact, asymptotic with the true value. That is, given a sufficiently large validation set, the estimated predictive accuracy converges to the true value (Valiant, 1985). Based on basic probability and statistical considerations, and regardless of the true population distribution, some very strong results are known. For example, with a validation set of 1,000 objects, the estimated predictive accuracy is accurate (note the "estimation" is accurate, not the predictive accuracy), and with 5,000 objects, it is virtually identical to the true predictive accuracy (Weiss and Kulikowski, 1991). These analytical results are based on Probability Approximately Correct (PAC) analysis, and this analytical method for evaluating classifier performance has been used since the mid-eighties (Valiant, 1985; Haussler, 1988), and is still a widely used analytical method (Goldberg, 2001; Trumbower et al., 2006). However, within many data mining problems these issues prove academic, as many classification problems are based on, relatively speaking, much smaller data sets, such as, the data set that is utilised within this dissertation, which contains 620 objects (see Chapter 5, section 5.5).

The larger the validation set, the more confidence the analyst can take from the estimated predictive accuracy. However, with a limited number of objects given in a sample data set, the preference would be to use as many objects as possible for training the classifier (Weiss and Kulikowski, 1991; Kantardzic, 2003). This presents a dilemma for the analyst, who must decide what proportion to take for training and what proportion to take for validation. Kantardzic (2003, pp. 84) notes that:

> "There are no good guidelines available on how to divide the samples into subsets. No matter how the data is split, it should be clear that different random splits, even with specific size of training and testing sets, would result in different error estimations."

It is this inconsistency in the estimated predictive accuracy for smaller data sets, and the dilemma of subset proportions that motivated the development of cross-validation methods, which essentially, seek to find an average predictive accuracy (described in section 3.3).

The standard approach for the train and test methodology, is to randomly divide the training set and validation sets into 2/3 and 1/3 partitions, respectively (Weiss and Kulikowski, 1991; Han and Kamber, 2006). Although other proportions such as 3/4 to 1/4 are used (see for example, Giudici, 2003).

Under certain circumstances it may be desirable to ensure that the distribution of the decision classes within the training set, reflects the distribution of the decision classes within the sample data set, particularly when considering skewed (imbalanced) samples (a data set that has varying size decision classes, see Chapter 4). Hong and Weiss (2001), when referring to their work on fraud detection, remarked that, the skewed distribution within their data set would baffle traditional data mining algorithms unless “stratified" samples were used within the training set. Stratifying, as it is termed, ensures that for each decision class the number of representative objects within the subset are proportional to the decision class sizes within the original data set, thus maintaining the distribution of the decision classes (Han and Kamber, 2006). Table 3.1.1.1 demonstrates an example of stratified sampling, based on taking a 50% training set (ergo, 50% validation set) from a data set with five decision classes and 620 objects.[4]

[4] The results of Table 3.1.1.1 and Table 3.1.2.1 (next subsection), were taken from the FIBR bank data set described later in Chapter 5 and utilised in Chapters 6 to 8. Used here as a pertinent example of the effect of sampling on an imbalanced (skewed) data set.

Decision Class	Number of Objects	Training (Taken)	Validation (Remaining)
1	16	8	8
2	319	160	159
3	163	82	81
4	107	54	53
5	15	8	7
Totals	620	312	308

Table 3.1.1.1: Taking a 50% Stratified Training Set

Within Table 3.1.1.1, 50% of the objects from each decision class have been taken to create a training set, rounding up where necessary, hence, there is not an exact a 50/50 split overall (312 objects within the training set, and 308 within the validation set). However, both the training and validation sets, retain a distribution close to that of the original data set.

In subsection 3.1.2, a novel method adapted from a known statistical technique and similar to stratification is presented as an alternative approach to selecting subsets. Although, re-sampling methods (described in section 3.3) mitigate the problem of subset sizes, re-sampling can be computationally expensive (in terms of time and memory required). So, under certain circumstances, where computer resources are an issue, it may still be preferable to use the train and test methodology. Moreover, for larger data sets the train and test methodology is perfectly adequate (Weiss and Kulikowski, 1991).

It should also be noted that, when considering ensemble methods (section 3.4), and with particular regards to the β-reduct aggregation process developed here (section 3.5), the process does not facilitate re-sampling of aggregated β-reducts (for reasons given in section 3.4.1). Hence, there is a requirement for a validation set, for evaluation of the aggregated β-reducts, and for testing and benchmarking against the VPRS vein graph analysis (as described in Chapter 2).

3.1.2 Statistical Sub-sampling Method

Much of the work relating to the estimation of the predictive accuracy within data mining, has borrowed heavily from known statistical approaches, particularly, re-sampling (Efron, 2003). Additionally, Kantardzic's quote (shown previously, on page 50), suggested that, there are no good guidelines on how to divide a sample into subsets (referring to data mining in particular). However, there has been statistical work relating to the sample size required, to be considered representative of a given population (Rao, 2000).

Israel (2007, pp. 1) notes, the sample size (of a population) is influenced by a number of factors "...the purpose of the study, population size, the risk of selecting a 'bad' sample, and the allowable sample error...". Israel presents an overview of different equations used to calculate sample sizes, dependant on specific data characteristics (for further reading see also, Cochran, 1963; Smith, 1983; Israel, 1992; Rao, 2000), and suggests that Yamane's (1967) Equation 3.1.2.1, is a simple but effective solution where little is known about the population distribution, and calculated as:

$$n = \frac{N}{(1+N(c)^2)}, \qquad (3.1.2.1)$$

where n is the calculated sample size, N is the population and c represents the required confidence level (e.g. for 95% confidence c = 0.05). Applying Equation 3.1.2.1 to the sample data set utilised within this dissertation (FIBR data described in Chapter 5), each decision class will be considered independently as a sub-population (Israel, 1992), with N equal to the decision class size, from which a representative sample size n must be taken at random. These representative samples will be combined to make the training set. Table 3.1.2.1 illustrates the calculated values for the 95% confidence level (c = 0.05).

Decision Class	Number of Objects	Training (Taken)	Validation (Remaining)
1	16	15	1
2	319	177	142
3	163	115	48
4	107	84	23
5	15	14	1
Totals	620	405	215

Table 3.1.2.1: Example Number of Objects Taken from each Decision Class by Equation 3.1.2.1 (with $c = 0.05$)

Within Table 3.1.2.1, where a decision class is under represented, by using Equation 3.1.2.1 to calculate the number of objects that can be taken from that class, a greater proportion is taken for training than validation. This is in contrast to simply setting an arbitrary proportion which takes no account of the decision class sizes (as demonstrated previously in Table 3.1.1.1).

Although this approach is quite novel in terms of data mining (considering the size of the sub-samples, that can be taken over each individual decision class), it is certainly less subjective than choosing a proportion value (as in Table 3.1.1.1). In some respects, by using Equation 3.1.2.1, and considering each decision class on an individual basis, this method could be considered a "supervised" sub-sampling method.

Figure 3.1.2.1 represents the plot of Equation 3.1.2.1, for the 95% and 90% confidence levels (values suggested by, Israel, 2007). With the number of objects, within a given sub-population (a decision class), plotted on the horizontal axis, and the concomitant number of objects the equation calculates are required to represent that sub-population, shown on the vertical axis. This dynamic approach for selecting objects from a decision class, based on what proportion can be reasonably taken, has a number of advantages, listed below Figure 3.1.2.1:

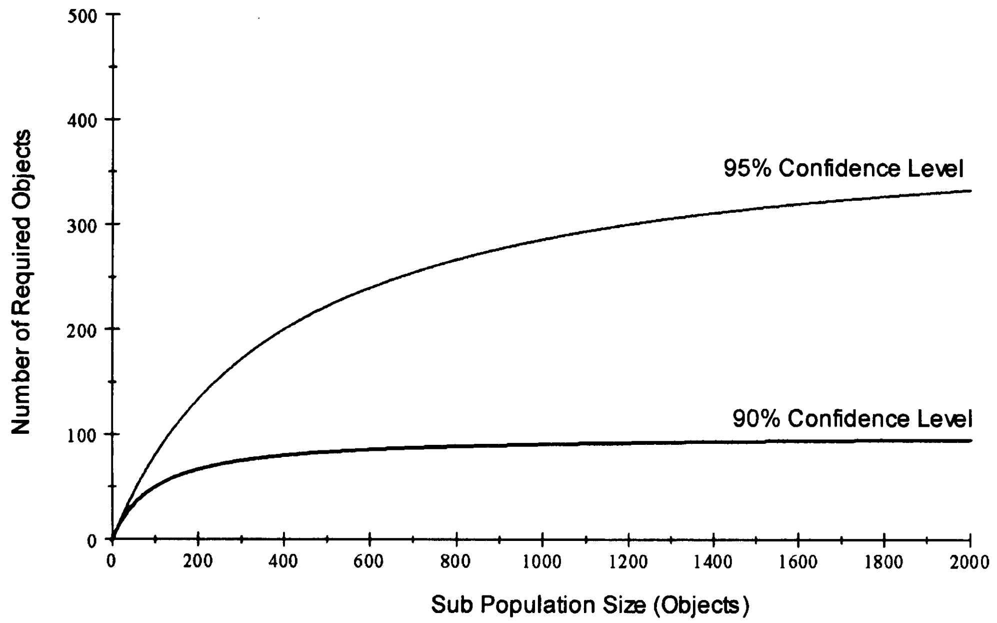


Figure 3.1.2.1: Number of Required Objects to Represent a Sub-population (Decision Class) Based on Equation 3.1.2.1

- As can been seen in Table 3.1.2.1 and Figure 3.1.2.1, for larger sized decision classes (such as decision class 2 in Table 3.1.2.1), the proposed statistical approach, recognises that, as the particular decision class size increases (i.e. sub-population increases), less proportion of the objects are required for training. The graph within Figure 3.1.2.1, also shows that more objects are required for the higher 95% confidence level, and that the gradient of the 95% confidence level, becomes shallower at a slower rate than the 90% confidence level.

- For smaller decision classes (such as decision class 1 in the Table 3.1.2.1), the proposed statistical approach, recognises that, as the particular decision class size decreases, more proportion of the objects are required for training. That is, it would be futile to try and take objects for the validation set, from an under represented decision class.

- This proposed statistical approach, mitigates a possible subjective decision on how to partition the subsets between the training and validation set, and removes that responsibility from the analyst.

- With the proposed statistical approach, the analyst is not implicitly bound, into having to set a specific training set proportion size, which is then blindly applied to all decision classes, irreverent to such issues as imbalanced data (skewed, see Chapter 4), as is the case in Table 3.1.1.1. Furthermore, with regards to imbalanced data, the proposed statistical method presented here, brings some balance to the sub-sampled training data set. This is because, to some extent, it attempts to prevent the larger decision classes from dominating, and smaller decision classes from being under represented (unless they are grossly under represented, where it becomes futile to consider taking objects for validation). This is potentially superior to the balancing methods described later in Chapter 4, as they rely on a certain amount of subjective opinion from the analyst.

The only potential drawback of this statistical approach, relates to the remaining objects that are used for the validation set. If the original sample from which the training set is drawn from, is as a whole small, then there are fewer objects left for the validation set. Hence, the analyst would be unable to take any confidence in the estimated predictive accuracy, based on the validation set. This problem is compounded when considering the predictive accuracy for each decision class independently (as will be described in subsection 3.2.1), especially for the under represented decision classes, which may yield few or no objects for validation (such as in decision classes 1 and 5 within Table 3.1.2.1). However, it must be acknowledged that attempting to train and test on a small data set is futile and that it would be wiser to keep as many objects as possible for training (Weiss and Kulikowski, 1991). Re-sampling methods, for overcoming this problem are described later in section 3.3.

To briefly summarise section 3.1, two basic approaches to estimating the predictive accuracy, namely, the apparent predictive accuracy, which is biased optimistically, and the train and test methodology which is less biased, but raises issues on how to partition a data set into training and validation sets, were described. With regards to the train and test methodology, three approaches to train and test sub-sampling were described: firstly, a simple random sub-sample, which takes no account of the distribution of the decision classes; secondly, stratified sub-sampling, which attempts to maintain the proportions (the distribution) of the decision classes between the data set, the training set, and the validation set; and thirdly, a novel statistical sub-sampling method was introduced, which considers the size of the decision classes on an individual basis (within the rest of this dissertation, we will refer to this method as the statistical sub-sampling method).

The VPRS software developed within this dissertation, for the purpose of evaluation, incorporates the apparent predictive accuracy method, and two of the train and test approaches for estimating the predictive accuracy, namely, stratified sub-sampling and the introduced statistical sub-sampling method.

3.2 Additional Classifier Performance Considerations

The previous section of this chapter, only considered the predictive accuracy in terms of a single calculated value. There are however, a number of other considerations that can be taken into account when investigating the predictive accuracy of a classifier.

The discussion in subsection 3.1.2 raised the concept of considering each decision class independently as a sub-population (with regards to the statistical sub-sampling method). Subsection 3.2.1 builds on this concept and describes the confusion matrix as a means to analyse the predictive performance of a classifier, for each individual decision class.

The over-optimistic (bias) estimation of a classifier's predictive accuracy, synonymous with the apparent predictive accuracy, and the potential for a range (variance) of predictive accuracy estimates based on different validation subsets, was described in the previous section 3.1. As it has a baring on the re-sampling work described later in section 3.3, subsection 3.2.2 elucidates the issues of bias and variance more formally.

3.2.1 The Confusion Matrix

Typically, within much of the classifier related literature, only the overall predictive accuracy is quoted (Poon et al., 1999; Oelericha and Poddig, 2006). This approach is satisfactory if all incorrectly predicted objects are of equal importance. In many applications, the cost of incorrectly predicting an object from one decision class, may be higher than incorrectly predicting an object from another decision class. The case of medical screening (as a first stage of diagnoses) is commonly quoted (Weiss and Kulikowski, 1991; Coppin, 2004), that is, if a patient is ill, but is classified as being healthy (known as a false negative error), then this situation is far more serious than if a patient is healthy and was diagnosed as being ill (a false positive error) (Han and Kamber, 2006).

The medical diagnosis example has a dichotomous decision class (ill or healthy), other applications may have a number of decision classes, such as the bank rating problem considered within this dissertation. Although, incorrectly predicting a bank is not a fatal mistake, the financial cost incurred, can be dependant on the degree to which the bank is incorrectly predicted (Harnett and Young, 2004, 2007).

If it is important to distinguish between the predictions for each decision class, then the confusion matrix (Weiss and Kulikowski, 1991; Han and Kamber, 2006), can be used to present the correctly/incorrectly predicted objects for multiple decision class problems. Table 3.2.1.1 demonstrates a three decision class confusion matrix.

	Predicted			**Predictive Accuracy**
Actual	**1**	**2**	**3**	
1	10	5	3	10/18=55.6%
2	4	40	5	40/49=81.6%
3	0	3	30	30/33=90.9%

Table 3.2.1.1: Example Confusion Matrix with Three Decision Classes

The left column of the confusion matrix in Table 3.2.1.1, shows the actual value of the objects, whereas the columns, under the 'Predicted' heading, display the predicted values for each object. Hence, the correctly predicted objects lie along the leading diagonal of the matrix (top left to bottom right). The predictive accuracy for each individual decision class is given at the end of each row. Taking the decision class 1 as an example, ten objects have been classified correctly, five and three objects have been erroneously classified into decision classes 2 and 3, respectively.

The confusion matrix is implemented within the developed VPRS software (vein graph and re-sampling), and as will be seen in Chapters 7 and 8, it proves to be an indispensable tool for evaluating predictive performance.

3.2.2 Bias, Variance and Overfitting

When using the estimated predictive accuracy to compare between classifier models, or classifiers trained on different subsets of a given data set, there are two additional issues, namely bias and variance which need consideration. It has already been stated that a classifier tested on the data that was used to train it, provides an over-optimistic estimation of the true predictive accuracy. In this case the estimation is said to be biased optimistically (i.e. the apparent predictive accuracy). In contrast, other methods for estimating the predictive accuracy, such as bootstrap re-sampling (see subsection 3.3.3), are quoted as being biased pessimistically (Breiman, 2001). Theoretical aspects of bias have been widely studied in statistics (Efron, 1982; Shao, 1988; Shao and Tu, 1995; Rao, 2000).

Whilst the estimated predictive accuracy taken on the validation set is typically less biased than that estimated on the training set, the range of estimated values will vary, depending not only on the specific sizes of the training and validation sets, but also on the objects randomly selected to be within those sets. This effect is known as the variance of the estimated predictive accuracy (Kantardzic, 2003). Different methods for estimating the predictive accuracy have different properties with regards to bias and variance. Indeed, it is often bias and variance which are used to benchmark between the different re-sampling methods (as will be shown in subsection 3.3.4), and will be utilised later in Chapter 8, to compare between predictive accuracy results.

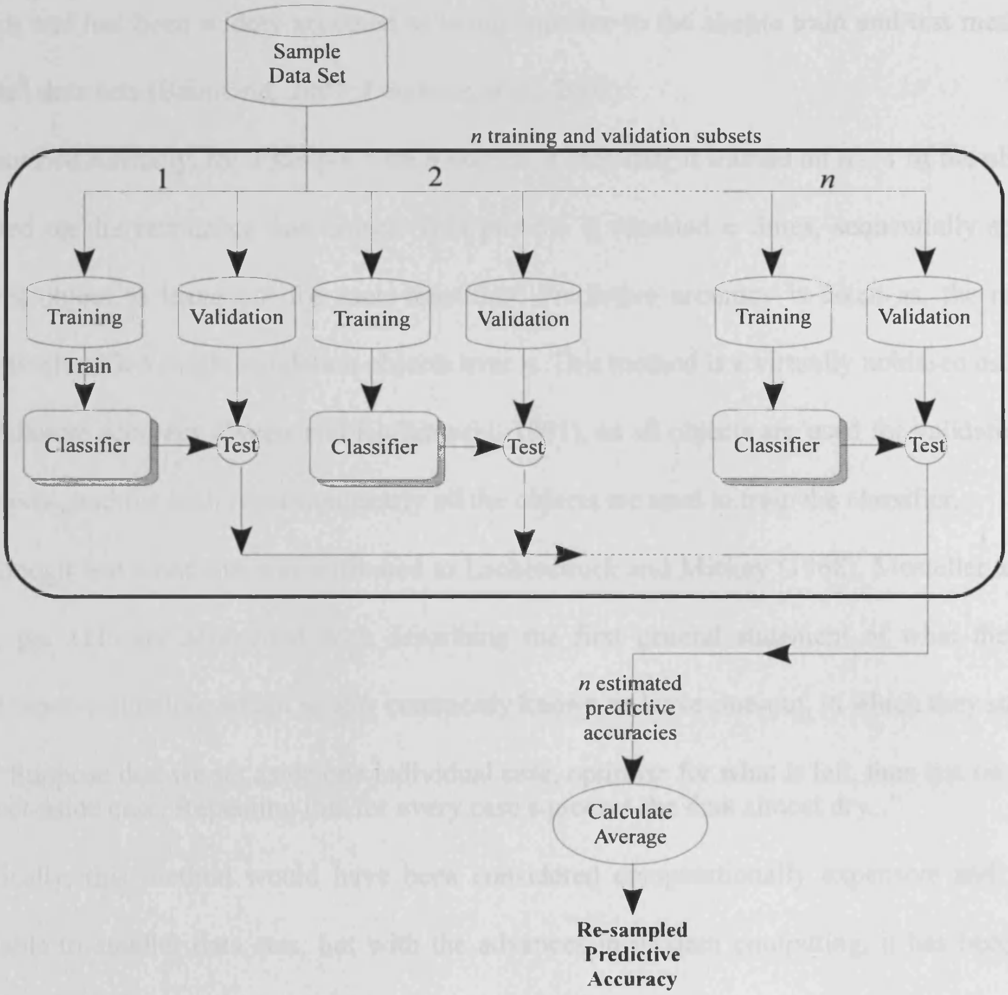
Overfitting, or over-specialising, of a classifier describes a situation where the classifier has been trained in such a way, as to increase the apparent predictive accuracy based on the training set (Kantardzic, 2003; Han and Kamber, 2006). Some classifier methods utilise the training set as a benchmark during their construction, in an attempt to increase the performance of the classifier, for example, the operation of neural networks through the back propagation method (see Geman et al., 1992; Zhang et al., 1999; Han and Kamber, 2006). Overfitting can increase the predictive performance on the training set, but the estimated predictive accuracy is biased very optimistically, and may actually be detrimental to the predictive accuracy on the validation set. Attempting to predict all objects within the training set correctly, including erroneous or misclassified objects, has the potential to weaken more general trends that would perform better on any future classification; it can also cause classifiers to become overly complex (Kantardzic, 2003). Generally, overfitting in terms of VPRS would result in an increase in the number of more complex (more attributes), less generalised (weaker strength) rules, in an attempt to improve the predictive accuracy on the training set. It also has the undesirable affect of, decreasing interpretability, as a consequence of those more complex rules.

3.3 Re-sampling Methods

Sub-sampling methods are an improvement on estimating the predictive accuracy, when compared to the apparent predictive accuracy. However, as suggested in section 3.1, sub-sampling has a number of drawbacks, such as, selection of the training and validation sample sizes and whether or not the data set is large enough to reasonably partition it into a training set and validation set. By the late 1940s and early 1950s, it was recognised that more advanced methods were required to address these drawbacks. Hence, emphasis was put on developing methods that were described as cross-validation, examples include the works of, Cureton (1951), Katzell (1951), Mosier (1951) and Wherry (1951), who contributed suggestions to 'Symposium: The need and means of cross-validation' (1951).

By the early eighties, a number of varying cross-validation methods had been proposed (Stone, 1974; Efron 1979, 1982), each promoting a different "re-sampling plan", with differing associated properties in terms of bias, variance, overfitting and computational requirements (Weiss and Kulikowski, 1991). Cross-validation has become more generally known as re-sampling, where the term cross-validation, itself, has become more synonymous with a specific re-sampling method known as *k*-fold cross-validation. The other two most prominent re-sampling methods being leave-one-out and bootstrapping (described next). Figure 3.3.1 presents an illustration of the general re-sampling methodology, which encompasses the three methods mentioned here (leave-one-out, *k*-fold cross-validation and bootstrapping), described in more depth in the following subsections.

Within Figure 3.3.1, for *n* repetitions, *n* different pairs of training and validation subsets are taken from the sample data set. The estimated predictive accuracy is calculated as the proportion of objects, from all repetitions of the validation set that were predicted correctly; or equivalently, by taking the average of all the estimated predictive accuracies from all repetitions of the validation set (Breiman, 1996b). The size of the training set taken within each repetition can be larger than that taken with regards to the simple train and test methodology, because at some stage within the repetitive process, all objects will be included in an independent validation set (though, not necessarily true for the bootstrapping method described in subsection 3.3.3). This is the main advantage of re-sampling over simple train and test methodology (Weiss and Kulikowski, 1991). That is, not only can more objects be taken for training purposes, but all objects at some stage, will be used for validation purposes.



Fig

ure 3.3.1: Re-sampling Approach to Estimating the Predictive Accuracy

Re-sampling also has an advantage over analytical approaches (such as those used in statistics), because it makes no prior assumptions about the distribution of the data (Kantardzic, 2003); this irreverence to the underlying distributions of the attributes, is more in line with the general RST/VPRS philosophy and the "Let the data speak for itself" line of thought, as quoted in the previous chapter. The following subsections describe three of the most established re-sampling methods, namely, leave-one-out, *k*-fold cross-validation and bootstrapping.

3.3.1 Leave-one-out

Attributed to Lachenbruch and Mickey (1968), leave-one-out is perhaps the simplest re-sampling method, and has been widely accepted as being superior to the simple train and test methodology for small data sets (Baumann, 2003; Lendasse et al., 2003).

Described formally, for a sample with n objects, a classifier is trained on $n-1$ of the objects and validated on the remaining one object. This process is repeated n times, sequentially selecting a different object to leave out for each repetition. Predictive accuracy is taken as, the number of correctly classified single validation objects over n. This method is a virtually unbiased estimator of the predictive accuracy (Weiss and Kulikowski, 1991), as all objects are used for validation during the process, and for each repetition nearly all the objects are used to train the classifier.

Although leave-one-out was attributed to Lachenbruch and Mickey (1968), Mosteller and Tukey (1968, pp. 111) are accredited with describing the first general statement of what they termed simple cross-validation, which is now commonly known as leave-one-out, in which they state:

> "Suppose that we set aside one individual case, optimise for what is left, then test on the set-aside case. Repeating this for every case squeezes the data almost dry..."

Historically, this method would have been considered computationally expensive and was only applicable to smaller data sets, but with the advances in modern computing, it has become more feasible to use the leave-one-out re-sampling method.

3.3.2 *k*-fold Cross-validation

The *k*-fold cross-validation method was proposed by Stone (1974), and is a generalisation of the leave-one-out re-sampling method. Within this method, the sample data set is split into k subsets of equal size, that is, the number of objects in each set are equal (or as near as can be). To demonstrate the method, 200 objects may be split into ten ($k = 10$) equally sized sets of 20 objects (objects selected randomly). Nine of these sets (90% of the sample data set) would be used to train the classifier; the remaining set would be used to test the classifier. For each repetition[5], a different set out of the ten available sets would be left out for use as a validation set.

[5] Figure 3.3.1 refers to n repetitions, however for *k*-fold cross-validation, we will be referring to k repetitions.

The *k*-fold cross-validation estimated predictive accuracy, is calculated as the average of all predictive accuracies from all *k* repetitions. The advantage of the *k*-fold cross-validation is that all the cases in the available sample are used for testing, and for each repetition almost all the cases are used for training the classifier (though not as many relative to the leave-one-out method).

In comparison to the leave-one-out method, *k*-fold cross-validation requires less repetitions, and is subsequently less computationally expensive. Within the extant literature, values of *k* generally range from 10 down to 2 (Efron, 1982; Zhang *et al.*, 1999; Dietterich, 2000a). A simplistic formula to evaluate *k* was presented in Davison and Hinkley (1997), that is $k = \min(n^{1/2}; 10)$, where *n* is the number of objects in the data set (see also Wisnowski *et al.*, 2003).

Considering today's computational power, it may be difficult to justify using *k*-fold cross-validation over leave-one-out, but as a direct consequence of this computing power, the rate at which information is gathered, and the volume which is being collated, is out stripping our ability to analyse it. As a testament Kantardzic (2003) notes:

> "...the problem of data overload looms ominously ahead. Our ability to analyse and understand massive data sets, as we call large data, is far behind our ability to gather and store the data."

In general, the "large data sets" of the past are probably considered medium sized in regards to the modern day comparison. With this in mind, then *k*-fold cross-validation is still very much a relevant and practical method. Han and Kamber (2006) stress that even if computational power allowed more folds ($k = n$, for leave-one-out), it is still preferable to use *k*-fold cross-validation as it has a lower variance than leave-one-out (discussed next). Moreover, they recommend stratified *k*-fold cross-validation for imbalanced (skewed) data sets. Stratified *k*-fold cross-validation is similar to the concept of stratification for the simple train and test methodology (see subsection 3.1.1), it seeks to retain the proportional distributions of the decision classes within the individual *k*-fold subsets (Thomassey and Fiordaliso, 2006).

It should be pointed out though, as stated earlier, given a sufficiently large data set, the simple train and test methodology is a perfectly acceptable method for estimating the true predictive accuracy (stated in subsection 3.1.1).

3.3.3 Bootstrapping

Since its introduction by Efron (1979, 1982), the bootstrap re-sampling method has received much scrutiny and attention within statistical analysis, and is an ongoing area of research (Efron, 2003). The bootstrap method draws a random sample of equal size to the sample data set (size *n)*, using sampling with replacement (hence some objects in the sample may be duplicated), this constitutes the training set (hence the training set is also of size *n*), any objects that do not appear within the training set are utilised as the validation set.

For example, given a sample of 200 objects, sampling with replacement *n* times yields a training set of 200 objects (some objects may appear more than once). On average, the proportion of the objects appearing in both the sample data set and the training set is 0.632 (0.368 are therefore duplicates), hence the average proportion of the validation set is 0.368 (Han and Kamber, 2006). The theoretical motivation behind the bootstrap method, was summed up by Shao and Tu (1995), who point out that a data set of size *n*, has 2^{n-1} non-empty subsets, the leave-one-out re-sampling method only utilises *n* of them, and that the estimated predictive accuracy may be improved by using more than *n* or even 2^{n-1} subsets.

There is little guidance on the number of repetitions required within bootstrapping. Original estimates suggested between 25 to 200 (Breiman, 1996a; Weiss and Kulikowski, 1991), but the value appears to be dependent on the classifier model in question. Indeed Dixon et al. (1987), used 500 bootstrap repetitions, whereas Brownstone and Valletta (2001) used up to 1,000. Andrews and Buchinsky (2000, pp. 23) noted the, "...ad hoc manner", for choosing the number of bootstrap repetitions, and suggest a three-step method for solving the problem, to achieve a "desired level of accuracy", with respect to measures of statistical inference.

With regards to calculating the predictive accuracy, the simplest method within bootstrapping is to take the average of the estimated predictive accuracies from all the repetitions (similar to *k*-fold cross-validation). Known as the *e*0 estimate (Weiss and Kulikowski, 1991), it yields a low variance result, but is typically biased pessimistically. There have been a number of alternative methods for calculating the predictive accuracy with regards to bootstrapping, which are discussed in the next subsection.

3.3.4 Comparison of Re-sampling Methods

As described previously, the apparent predictive accuracy is by far the most biased estimation of predictive accuracy. Using a validation set offers an unbiased estimate, but has high variance for small sample sizes, thus the analyst cannot take any confidence in the estimation. Re-sample methods are computationally more expensive, but result in, less biased, and lower variance estimates. However, there are differences between the bias and variance associated with each re-sampling method. Table 3.3.4.1 summarises the bias (severity within the parenthesis) and variance, for the estimation methods discussed previously in subsections 3.3.1 to 3.3.3.

	Bias	Variance
Apparent Predictive Accuracy	Optimistically (high)	-
Train and Test Methodology	unbiased	High
Leave-one-out	unbiased	High
k-fold cross-validation	Pessimistic (low)	Low
Bootstrapping	Pessimistic (medium)	Low*

* lower than *k*-fold cross-validation

Table 3.3.4.1: Bias and Variance Comparisons of Predictive Accuracy Estimation Methods

As shown within Table 3.3.4.1, leave-one-out is said to be a virtually unbiased estimate of the true predictive accuracy, but has a high variance (Weiss and Kulikowski, 1991). With regards to bootstrapping, as stated in subsection 3.3.3, on average a classifier is only trained on 0.632 of the sample data set, so bootstrapping provides an estimate that is biased pessimistically, but has a very low variance. With regards to *k*-fold cross-validation, it could be described as a "half-way house" between leave-one-out and bootstrapping, as in general, it produces a slightly pessimistic result when compared to leave-one-out, but less pessimistic than bootstrapping, additionally its variance is much lower than leave-one-out, but not as low as bootstrapping (Han and Kamber, 2006).

There have been attempts to adjust for the known effects of bias and variance within *k*-fold cross-validation and bootstrapping. With regards to bootstrapping, the 0.632B linear combination estimator (Efron and Tibshirani, 1997), has demonstrated strong results, and is presented in Equation 3.3.4.1:

$$(0.368 \times e0) + (0.632 \times app), \qquad (3.3.4.1)$$

where *app* is the apparent predictive accuracy (the equation has been adapted to utilise predictive accuracy as opposed to error rate), and the *e*0 estimate was described in the previous subsection. Sima and Dougherty (2006) describe the 0.632B estimator as a convex combination, and note that there has also been some attempts to improve *k*-fold cross-validation by using a similar convex combination approach (Toussaint and Sharpe, 1975; Raudys and Jain, 1991).

There are arguments both for, and against each re-sampling method. With regards to leave-one-out, although it is said to be a virtually unbiased estimator, Baumann (2003, pp. 395) found:

> "...the commonly applied leave-one-out cross-validation has a strong tendency to overfitting, underestimates the true prediction error, and should not be used without further constraints or further validation."

Shao (1993), in what they described as the inconsistency of leave-one-out, found it could be rectified by using *k*-fold cross-validation. Wisnowski et al. (2003), noted that the challenge when using *k*-fold cross-validation is not to over-fit the model, as may happen with leave-one-out.

The possible superiority of bootstrapping over the other re-sampling methods was reported in (Efron, 1983), and it has gained wide acceptance that it can outperform *k*-fold cross-validation for small data sets (Kantardzic, 2003), but other studies have suggested that, under certain circumstances, *k*-fold cross-validation can be superior (Weiss, 1991). Additionally, depending on the number of repetitions required, bootstrapping can prove computationally more expensive over the other re-sampling methods (Giudici, 2003).

Here, the most generally accepted re-sampling methods have been presented, but there are many variations. The usual motivation behind investigating alternative re-sampling methods has been to combine the strengths of *k*-fold cross-validation and bootstrapping (to achieve low bias and low variance estimates, see Table 3.3.4.1). Efron and Tibshirani (1997) discussed combining bootstrapping with *k*-fold cross-validation (including what they described as the leave-one-out bootstrap); Ambroise and McLachlan (2002) applied a '10-fold cross-validation 0.632B' to their micro array data. More recently, Fu et al. (2005), presented a similar concept to that presented in Ambroise and McLachlan (2002), and based their approach on a similar combining principle. Lendasse et al. (2003), describe the Monte-Carlo cross-validation, whereby, repeated validation sets were randomly and sequentially drawn.

3.4 Ensemble Methods

Re-sampling is still an active area of research within statistics, and is gaining more attention within the machine learning community, in particular, bootstrapping as an alternative to *k*-fold cross-validation. Efron (2003, pp. 138) when discussing the future of the bootstrap, commented that:

> "...its workhorse status in machine learning, as seen in the recent book by Hastie, Tibshirani, and Friedman (2001), makes it a statistical success story in the outside world.",

but later concedes that what is not available (pp. 139), "...is theoretical reassurance that the numerical gains... will hold up in general practice." This is a sentiment shared by other authors of data mining based research, who see that data mining being developed by computer scientists for very practical usage, has evolved separately from the rigours of statistical mathematics (Kantardzic, 2003). Giudici (2003) commented that statistical methods should be used to study and formalise data mining methods and that (pp. 6):

> "...we need to develop a conceptual paradigm that allows the statisticians to lead the data mining methods back to a scheme of general and coherent analysis."

However, one promising area of research making a very practical impact within data mining is ensemble methods. Whereby, the numerous classifiers constructed during a re-sampling phase are combined to produce a more "stable" classifier, that can improve predictive accuracy (Han and Kamber, 2006; Skurichina and Duin, 1998). The authors of the more established methods do provide mathematical evidence that underpin their ensemble methods (Breiman, 1996; Freund and Schapire, 1997), but the empirical evidence for the success of ensemble methods is also encouraging (Webb, 2000; Borra and Di Ciaccio, 2002; Hothorn and Lausen, 2003a; Stefanowski, 2004).

The methods described within the previous sections focused on how to accurately estimate the true predictive accuracy of a classifier, with consideration given to bias and variance. The methods in themselves, do not actively seek to improve the predictive accuracy of the classifier. Ensemble methods however, do seek to improve the predictive accuracy of a classifier through re-sampling with little extra computational expense (Han and Kamber, 2006).

Breiman's (1996a) bagging (**b**ootstrap **agg**regat**ing**), and Freund and Schapire's (1997) paper on boosting, are the seminal works on ensemble classifiers. There have been a number of extensions to these ensemble methods, but they can be placed into two categories, as stated by Bauer and Kohavi (1999, pp. 105), "...those that adaptively change the distribution of the training set based on the performance of previous classifiers (as in boosting methods) and those that do not (as in bagging)."

There have been a number of studies investigating and comparing the potential performance enhancing capabilities of these ensemble methods (Dietterich, 2000a; Kuncheva et al., 2001; Borra and Di Ciaccio, 2002). In terms of bias and variance, it is generally known (through cross-validation of ensemble methods), that when compared to bagging, boosting tends towards a lower bias, but higher variance. Furthermore, if the boosting algorithm is left to proceed unchecked, it will over-fit the training set, and become biased optimistically (Friedman, 1999, 2002). Ridgeway (2002, pp. 380) misleadingly suggests that Breiman's (2001) iterated bagging seeks to combine the ideas of bagging and boosting in the expectation that, "...boosting's bias reduction together with bagging's variance reduction could produce excellent predictive models." However, Breiman (2001, pp. 262) had already denied the link, and stated, "...iterated bagging has no connection to boosting...", although he does acknowledge there are similarities with Freidman's (1999) work on gradient boosting. The confusion may lie in the fact that iterated bagging is an attempt to improve bagging by reducing the bias of the constructed classifier.

There are many examples of the successful performance enhancing capabilities of both bagging and boosting (for a good overview see, Dietterich, 2000b; Sewell, 2007). Some more exotic methods have suggested other methods for aggregating classifiers. Bauer and Kohavi (1999) proposed an interesting variant on bagging namely wagging (weight **aggregating**). Where uniform weights are initially associated with each object, and they describe wagging as a method that (pp. 122), "...seeks to repeatedly perturb the training set as in bagging, but instead of sampling from it, wagging adds Gaussian noise to each weight...", from which they then induced their decision tree and Naïve-Bayes classifiers (see, Kohavi et al., 1997; Quinlan, 1993). They found that the results of wagging were comparable with bagging. Webb (2000) took wagging and combined it with boosting in what they named MultiBoosting, which as they described, harnessed the high bias reduction of boosting with wagging's superior variance reduction.

Other notable variants, such as Hothorn and Lausen's (2003b) paper on 'Double bagging' suggests using the in-sample (similar to the training set, described in the next subsection) from the bootstrap to train a classifier, and to simultaneously use the out-sample (similar to the validation set), to perform linear discriminant analysis, then combining the results of both to improve the performance of their constructed classifier. Their further work on 'Bundling classifiers' (Hothorn and Lausen, 2005), builds on the idea of combining different classifier methods, in an attempt to combine the best elements of a range of classifier methods within a decision tree framework.

Bryll et al. (2003) presented another promising direction, in the form of attribute bagging, whereby random subsets of the attributes are taken (instead of subsets of objects). They point to the faster computation time, and claimed that (pp. 1298), "...attribute partitioning methods are superior to data partitioning methods (e.g. bagging and boosting) in ensemble learning." Attribute bagging is a promising method, because it combines the advantages of increasing the predictive accuracy and stability of a classifier (as in other ensemble methods), whilst also performing attribute selection (feature selection, see Chapter 4). However, within their study, a substantial validation set had to be set aside (validation of ensemble classifiers is discussed in the next subsection). Perhaps combining attribute bagging with the established data partitioning bagging, would yield a powerful classifier "wrapper" method, that could tackle feature selection and the combined problems of bias and variance (stability), whilst improving predictive accuracy. However, this approach could result in a computationally expensive process, if data partitioning bagging was performed for every iteration of the attribute bagging process.

For the purpose of this dissertation, we have concentrated on Breiman's (1996a) original bagging methodology, described fully in the next subsection. Additionally, the developed VPRS software will not only allow the analyst to perform bootstrap aggregation, but also aggregation over the leave-one-out and *k*-fold cross-validation re-sampling methods.

3.4.1 Bagging

Bagging was introduced as an ensemble method by Breiman (1996a), as a way to improve accuracy and stability of a classifier. Stability, is linked to the issue of variance, whereby, classifiers trained and validated on different subsets of a sample data set can have a varying range of predictive accuracies (variance). It follows that, improving the stability, should generally improve the predictive accuracy of a classifier, by insuring that the predictive accuracy on any unseen data set is within a tight interval.

Bagging aims to combine the classifiers constructed during bootstrap re-sampling, to create an improved aggregated classifier. Figure 3.4.1.1 displays the general model for an aggregated classifier.

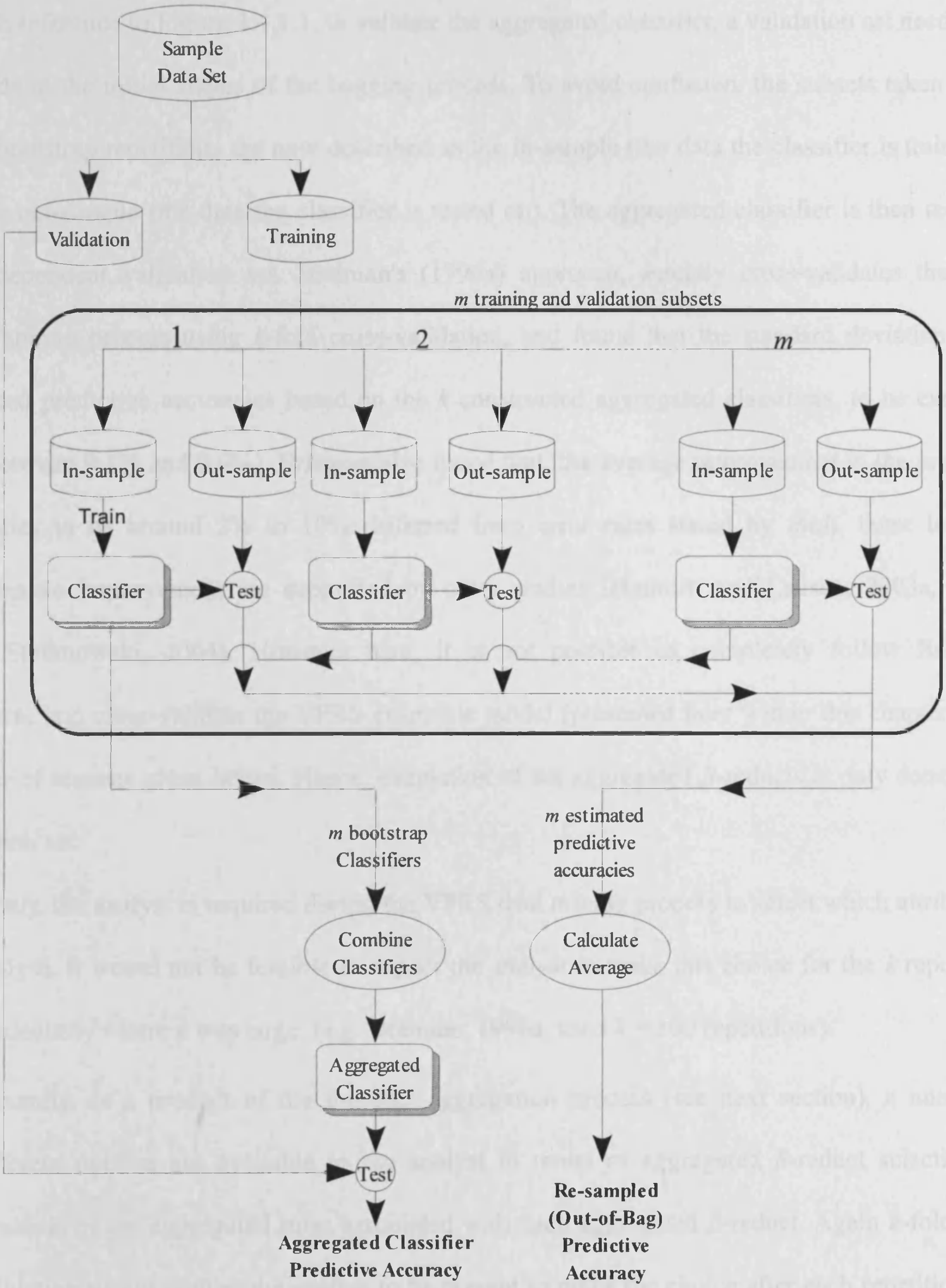


Figure 3.4.1.1: Construction of an Aggregated Classifier (such as in the Bagging Method)

The methods for aggregating classifiers can be model dependant, but in general, after the *m* bootstrap classifiers have been trained (as shown in Figure 3.4.1.1), each model can "vote" on the classification of any unseen objects (thus forming the aggregated classifier), and the classification of the unseen objects, are then based on the majority vote.

With reference to Figure 3.4.1.1, to validate the aggregated classifier, a validation set needs to be set aside at the initial stages of the bagging process. To avoid confusion, the subsets taken during the *m* bootstrap repetitions are now described as the in-sample (the data the classifier is trained on) and the out-sample (the data the classifier is tested on). The aggregated classifier is then tested on the independent validation set. Breiman's (1996a) approach, actually cross-validates the entire bootstrapping process using *k*-fold cross-validation, and found that the standard deviation of the estimated predictive accuracies based on the *k* constructed aggregated classifiers, to be extremely low (between 0.1% and 0.9%). Brieman also found that, the average improvement in the predictive accuracies to be around 2% to 10% (inferred from error rates stated by *ibid*), these levels of performance improvement are supported by other studies (Hothorn and Lausen, 2003a, 2003b, 2005; Stefanowski, 2004). However here, it is not possible to completely follow Brieman's approach, and cross-validate the VPRS ensemble model (presented later within this chapter), for a number of reasons given below. Hence, evaluation of the aggregated β-reducts is only done on the validation set.

- Firstly, the analyst is required during the VPRS data mining process to select which attributes to analyse. It would not be feasible to expect the analyst to make this choice for the *k* repetitions, particularly where *k* was large (e.g. Breiman, 1996a, used $k = 100$ repetitions).
- Secondly, as a product of the β-reduct aggregation process (see next section), a number of different options are available to the analyst in terms of aggregated β-reduct selection and selection of the aggregated rules associated with each aggregated β-reduct. Again *k*-fold cross-validation would require the analyst to be present to make the choice after each repetition of *k*.

To utilise cross-validation, this point and the previous point would require full automation, perhaps a consideration for the future, but infeasible within the framework of the VPRS software developed here.

- Thirdly, from preliminary work (Griffiths and Beynon, 2007, 2008), it was found that, during the VPRS analysis, each bootstrap of the data set can produce vastly different sets of results, particularly with regards to the selected β-reducts. The number of bootstrap repetitions required within a VPRS based analysis was found to be greater (500 plus) than that advocated in the extant literature (e.g. Breiman, 1996a, used 50 bootstraps). Mainly because, a range of different β-reducts are selected during the β-reduct aggregation process (described in the next subsection, and shown in Chapter 8), thus, more repetitions are required to ensure confidence in the results. Hence, it would be computationally infeasible to cross-validate the aggregated classifier constructed within the framework of the VPRS software developed here (within a reasonable time frame).

Here, the statistical method described in subsection 3.1.2 is employed, to set aside a validation set. It should be noted though, that within the developed VPRS software, the re-sampling predictive accuracies are recorded, and can be used as an indication of the future performance of a classifier. The extant literature on ensemble methods, refers to this use of the re-sampling estimated predictive accuracy as the "out-of-bag" estimate (Breiman, 1996b; Hothorn and Lausen, 2003a). Moreover, Breiman (2001, pp. 11), noted that, "...using the out-of-bag error estimate removes the need for a set aside test set [validation set]...", but concedes that the estimate will be pessimistic with regards to the bootstrap estimation because the classifier is only trained, on average, with about two thirds of the available training set (see also, Tibshirani, 1996; Wolpert and Macready, 1999).

The performance enhancing effects of bagging are widely accepted, Han and Kamber (2006, pp. 367) stated that:

"The bagged classifier often has significantly greater accuracy than a single classifier

> derived from *D*, the original data. It will not be considerably worse and is more robust to the effects of noisy data. The increased accuracy occurs because the composite model reduces the variance of the individual classifier. For prediction, it was theoretically proven that a bagged predictor will *always* have improved accuracy over a single predictor derived from *D*."

With regards to comparisons between bagging and boosting (the most well established alternative method), boosting has been found to outperform bagging, but not universally so. Boosting performs particularly poorly for noisy data sets, and can in fact, produce a classifier which is worse than one produced from a single run (Bauer and Kohavi, 1999; Dietterich, 2000a).

3.5 VPRS Re-sampling and β-reduct Aggregation

As stated in the introduction and Chapter 2, dynamic reducts (Bazan et al., 1994; Leifler, 2002; Jiang and Abidi, 2005), is perhaps, the closest work utilising RST and re-sampling, relating to the work undertaken in this dissertation. Wherein, a reduct is described as dynamic, if it appears within those reducts identified from a sample data set, and within all subsets of the sample data set taken during re-sampling. In most cases, this definition is too restrictive and a proportion measure $0 \leqslant \epsilon \leqslant 1$ was introduced. That is, if a reduct appeared in a proportion of the subsets that was greater than the threshold ϵ, defined by the analyst, then it would be considered dynamic. As reducts generated from a sample data set are sensitive to change in the data (e.g. by removal or addition of objects), the identification of dynamic reducts is in the hope that more stable reducts will be found, and hence that they will perform better as a classifier on unseen data (Jensen, 2004). Bazan (1998) showed that, generally, dynamic reducts performed no better than the conventional reducts. It can be speculated that no improvement was seen, because in a sense nothing is done to stabilise the rules associated with the reducts that have been shown to be dynamic (stable). Thus the suggestion here, is to consider a process of aggregating the most stable reducts (β-reducts specifically) and associated rules, in an attempt to induce the most stable rule set, and potentially increase its predictive performance.

In terms of ensemble methods and RST, Stefanowski (2004, 2007) has presented the most notable work. Whereby, they applied bagging to their MODLEM rule induction algorithm (based on RST, Stefanowski, 1998), and they found that (pp. 341), "... bagging significantly outperformed the single classifiers..." in 14 out of 18 of their sample data sets.

There has been limited attention given to ensemble methods within RST/VPRS, perhaps due to the complexities of reduct identification (Jensen, 2004) and rule generation (both theoretically and computationally), further compounded by the question of reduct/rule aggregation. Stefanowski (2004, 2007) opted for bagging their MODLEM approach, because it provided an efficient single classifier, within a reasonable computational time frame. Jiang and Abidi (2005) emphasised the advantage of their methods involving rules induced from dynamic reducts, which produced concise more generalised rules, and also maintained predictive accuracy. The undesirable relation between potentially improving the predictive accuracy of a classifier through aggregation, and the additional complexity involved with an aggregated classifier, was summed up succinctly by Breiman (1996a, pp. 137), when commenting on their classification tree approach, "What one loses, with the trees, is a simple and interpretable structure. What one gains is increased accuracy."

The method introduced within the VPRS software developed here, does not restrict the analyst to a threshold value ϵ as in dynamic reducts, but rather presents them with the important information concerning all selected β-reducts, when undertaking a VPRS re-sampling analysis (the selection process is described next). Here, the software, allows the analyst to asses which β-reduct, that has been selected on the most number of repetitions, so potentially stable, to aggregate. The method of β-reduct aggregation described later in subsection 3.5.2, combines the minimal covering rule sets associated with each occurrence of the selected β-reduct, and again allows the analyst to select the rules which occur most often, hence the most general (stable) rules. This mitigates the problem of aggregated classifier complexity as the analyst can "skim" off the most general rules to construct the final classifier. Before β-reduct aggregation is described further, some explanation must be given to how the β-reduct selection process has been automated.

3.5.1 β-reduct Automated Selection

Within Chapter 2, the vein graph was described as a method for presenting all β-reducts associated with a sample data set, from which the analyst could choose a β-reduct for analysis. In a re-sampling environment, it would be impracticable for the analyst to choose a β-reduct for each repetition. Here, to automate the process of β-reduct selection, a criteria has been defined to select a single β-reduct from each repetition. Originally used in Griffiths and Beynon (2008), which was an extension from that outlined in Beynon et al. (2004), the criteria is specified as (see Chapter 2 for terminology):

i. β_{min} threshold greater than specified β value. Infers that the selected β-reduct(s) will have a β_{min} threshold value (defined in Chapter 2) greater than a β-value defined by the analyst.

ii. The highest quality of classification possible, associated with the β-reduct(s) identified in i). Infers the identified β-reduct(s) from i), will assign a classification to the largest possible number of objects in the training set. A consequence being, the rules constructed may misclassify a number of objects. It is then more probable that the β sub-domains associated with the selected β-reduct(s) for this criterion, are at the lower end of the β domain (0.5, 1].

iii. The highest β_{min} value from those associated with the β-reduct(s) identified in i) and ii). Infers the selected β-reduct(s), will have the highest proportional level of majority inclusion, with regards to the condition classes associated with the decision classes, of the relevant β-positive regions.

iv. Least number of decision rules associated with the β-reduct(s) identified in i) to iii). Infers that the selected β-reduct(s), will have the most general rules (higher strengths). This follows the science tenet of Occam's Razor (Domingos, 1999), the principle is expressed in Latin as the *lex parsimoniae* ("law of parsimony" or "law of succinctness"), and is often paraphrased as "All things being equal, the simplest solution tends to be the right one!"

v. Least number of condition attributes in the β-reduct(s) identified in i) to iv). Infers an identified β-reduct will have the least complex decision rule set. It implies, if a choice is available, then the simpler model should be chosen. Hence, a β-reduct with less condition attributes would be chosen, over one with more condition attributes, to exact a simpler model. The role of this criterion is akin to the point made in iv).

vi. The largest sub-domain of β associated with the β-reduct(s) from those selected in i) to v). Infers a selected β-reduct will have been chosen from the largest choice of β value (largest sub-domain). This criterion replicates a level of stochastic subjectivity, namely, a random choice of a β value would more probably mean 'this' selected β-reduct would be chosen.

Initial studies undertaken during the software development, indicated that the criteria outlined above, is adequate for selecting a single β-reduct. At such a point where, only a single β-reduct is identified (hence selected), there is no requirement to go through the remaining criteria. In the unlikely case of more than one β-reduct having being identified after point vi), a single β-reduct is then randomly selected from those remaining.

Here, points i) and iv) are in addition to those suggested by Beynon et al. (2004), and were considered through work presented in Griffiths and Beynon (2008). Point i) allows the analyst to have more control over the accuracy of the identified β-reduct(s). Point iv) has been inserted, as it was found that point v) was too impacting or influential on the identified β-reducts, almost defaulting to the selection of a β-reduct(s) based on a single condition attribute (a problem that was identified during the development of the software, and illustrated in Griffiths and Beynon, 2007). It should be noted that, the order of the criteria is not fixed and a different order may be more appropriate (with different inference, as will be demonstrated in Chapter 8 section 8.1).

3.5.2 β-reduct Aggregation

This subsection proposes a method for identification and construction of aggregated β-reducts. For the *m* re-sampling repetitions undertaken during the aggregation process (Figure 3.4.1.1), each β-reduct and its details, such as QoC, associated rule set, predictive accuracy and the number of occurrences of a specific β-reduct, are recorded.

The analyst, presented with all the recorded details (as statistical measures and graphs etc.), can now make a choice of selecting the most appropriate β-reduct to aggregate. For example, based on 500 repetitions, a β-reduct may be selected on 300 instances, this is good evidence for the stability of the β-reduct (although not universally so, as will be shown in Chapter 8). The analyst would then select this β-reduct, the associated rule sets for each occurrence of that β-reduct would then be aggregated (described next), thus forming the aggregated rule set.

Once the aggregated rule set has been constructed, the analyst will be allowed to select which rules within the aggregated rule set they wish to use. Note that, even though condition attributes associated with the aggregated β-reduct may be equivalent, due to the re-sampling process, the rules associated with the β-reduct may differ, because of the presence or absence of objects during each repetition, in theory, the most (general) stable rules should occur more frequently. The selected rules are then validated on the set aside validation set (see Figure 3.4.1.1). The validation set allows comparison to be drawn between different aggregation preferences, that is, aggregation based on leave-one-out, *k*-fold cross-validation or bootstrapping. It also facilitates comparisons to be drawn between the final aggregated β-reduct and a single run VPRS analysis, and if required, the validation set can be used to benchmark against alternative classifier methods.

The following discussion, presents a simple example of the proposed β-reduct aggregation method. Considering VPRS within a re-sampling environment, with $m = 5$ repetitions and the training set size $n = 10$ objects, hence five β-reducts have been selected. Tables 3.5.2.1 to 3.5.2.3, present three hypothetical decision rule tables associated with three of the five β-reducts, that were equivalent in terms of the condition attributes $\{c_1, c_2\}$, that is to say, β-reduct $\{c_1, c_2\}$ has occurred on three of the five repetitions.

Rule		c_1		c_2		d_1	Support	Correct	Strength	Certainty
1	If	1	and	0	then	0	4	3	0.400	0.75
2	If	-	and	1	then	0	4	3	0.400	0.75
3	If	0	and	-	then	1	2	2	0.200	1.00

Table 3.5.2.1: First Decision Table Associated with the β-reduct $\{c_1, c_2\}$

Rule		c_1		c_2		d_1	Support	Correct	Strength	Certainty
1	If	1	and	0	then	0	4	3	0.400	0.75
2	If	-	and	1	then	1	2	2	0.200	1.00
3	If	0	and	-	then	1	2	2	0.200	1.00

Table 3.5.2.2: Second Decision Table Associated with the β-reduct $\{c_1, c_2\}$

Rule		c_1		c_2		d_1	Support	Correct	Strength	Certainty
1	If	1	and	0	then	0	4	4	0.400	1.00
2	If	0	and	-	then	1	2	2	0.200	1.00

Table 3.5.2.3: Third Decision Table Associated with the β-reduct $\{c_1, c_2\}$

With reference to the Tables 3.5.2.1 to 3.5.2.3, to aggregate each decision rule table, one instance of any specific rule based on the condition attribute values $\{c_1, c_2\}$ with a specific decision outcome d_1, is recorded in a new aggregated rule decision table, see Table 3.5.2.4.

Rule		c_1		c_2		d_1	Occurrence Support	Occurrence Correct	Occurrence Strength	Occurrence Certainty	Occurrence (R_{occ})
1	If	1	and	0	then	0	12	10	12/30 = 0.400	0.83	3
2	If	-	and	1	then	0	4	3	4/30 = 0.100	0.75	1
3	If	-	and	1	then	1	2	2	2/30 = 0.067	1.00	1
4	If	0	and	-	then	1	6	6	6/30 = 0.200	1.00	3

Table 3.5.2.4: Aggregated Rule Table, for the now Aggregated β-reduct $\{c_1, c_2\}$

In Table 3.5.2.4, the number of occurrences of a specific rule are also recorded under the 'Occurrence (R_{occ})' column. For instance, within the decision rule Tables 3.5.2.1 to 3.5.2.3, the rule "*If* $c_1 = 1$ *and* $c_2 = 0$ *then* $d_1 = 0$" appears once in each of the decision Tables (rule 1 in each case). Hence, the rule is recorded once in the aggregated rule decision Table 3.5.2.4, and the rule occurrence R_{occ} is recorded as three. The 'Occurrence Support' and 'Occurrence Correct' are taken as the sum of the Support and Correct values associated with each occurrence of the specific rule.

In defining the probabilistic measures, 'Occurrence Strength' and 'Occurrence Certainty', one must consider what the concepts of strength and certainty represents now, within the aggregated β-reduct environment (as opposed to a standard β-reduct). For a single run, Strength (see Chapter 2 section 2.2.3 Equation 2.2.3.3) essentially indicates what proportion of objects within the training set that a given rule can classify (and some indication on its future performance on unseen objects). Hence following this concept, Occurrence Strength has been defined as the average proportion of the objects a specific rule would give a classification to, within each training set repetition, defined more formally as:

$$Occurrence\,Strength = \frac{Occurrence\,Support}{nR_{occc}}. \tag{3.5.2.1}$$

A similar argument can be posed with regards to Occurrence Certainty. That is, Certainty (see Chapter 2 section 2.2.3 Equation 2.2.3.4) indicates the proportion of objects that a rule would correctly classify, from those it can give a classification to (also some indication of the possible future performance of the rule). So 'Occurrence Certainty' is defined as the average proportion of objects that a rule would correctly classify, from those it can give a classification to, within each training set repetition, defined more formally as:

$$Occurrence\ Certainty = \frac{Occurence\ Correct}{Occurence\ Support}. \qquad (3.5.2.2)$$

The Occurrence Certainty, cannot be greater than unity. However, the sum of the Occurrence Strengths may be greater than unity, unlike Strength associated with VPRS and Strength associated with RST (see, Pawlak, 2004), where the combined Strengths of the rules can at most, only add up to unity (in the case of an empty boundary region).

Finally, taking rules 2 and 3 from Table 3.5.2.4, as an example of how to manage an instance where two or more rules are based on the same condition attribute values but have differing decision class values; rule 2 would always be chosen over rule 3 when classifying unseen objects, because rule 2 has the greater Occurrence Strength. Hence, rule 3 is, in fact, superfluous within the decision rule table, as such, the rule can be ignored and removed from the table. Table 3.5.2.5 reflects the finalised aggregated rule decision table, with rule 3 removed (rule 4 in Table 3.5.2.4, is now re-indexed as rule 3 within Table 3.5.2.5)

Rule		c_1		c_2		d_1	Occurrence Support	Occurrence Correct	Occurrence Strength	Occurrence Certainty	Occurrence (R_{occ})
1	If	1	and	0	then	0	12	10	12/30 = 0.400	0.83	3
2	If	-	and	1	then	0	4	3	3/30 = 0.100	0.75	1
3	If	0	and	-	then	1	6	6	6/30 = 0.200	1	3

Table 3.5.2.5: Finalised Aggregated Rule Table

Table 3.5.2.5 presents an example, of how the final aggregated rule table will be displayed within the VPRS re-sampling software. The next short section describes the process of how to classify data using a rule table (standard or aggregated), and the method of nearest rule to classify objects that do not match the condition attribute values of any rule from a given rule set.

3.6 Classification Issues within VPRS

Within this chapter, the classifier has been referred to in a generic sense. Essentially, the decision rule table, the end product of a VPRS analysis, is the classifier associated with the developed VPRS software (vein graph and re-sampling). This section discusses the utilisation of the decision rule table with regards to object classification of training and validation sets.[6]

Previously (in Chapter 2), the decision rule tables were only discussed in terms of classifying the objects upon which they were trained. Furthermore, it was suggested that, the rules would only classify those objects, which had matching condition attribute values. Within this section, the objects within the validation set, and the objects that do not match the condition attribute values of any given rule from a rule set, are considered.

Those objects that do not match the condition attribute values of any rule, are classified by the 'nearest' rule method, as presented in Słowiński (1992). There are a number of other distance measures which could have been considered, such as Manhattan distance (Han and Kamber, 2006) or a method based on interpolation (Huang and Shen, 2006, 2008) may also prove effective, but here we have focused on Słowiński's (1992) method.

Equation 3.6.1 calculates a measure of distance (*dist*) between a decision rule and an object, computed from those condition attributes that determine the rule. For an object x, described by the condition attribute values $c_1(x)$, $c_2(x)$,..., $c_{|C|}(x)$, the distance of rule y described by $c_1(y)$,..., $c_i(y) \rightarrow d(y)$ (where i is less than, or equal to, the number of condition attributes associated with the β-reduct, and only the prime implicants of rule y are considered), is measured by:

$$dist = \frac{1}{i}\left\{\sum_{l=1}^{i}\left[k_l\left(\frac{|c_l(x)-c_l(y)|}{v_{l^{max}}-v_{l^{min}}}\right)^p\right]\right\}^{\frac{1}{p}}, \tag{3.6.1}$$

[6] Classification with regards to VPRS, is discussed more in-depth within this Chapter, rather than within Chapter 2, because the validation set was first discussed within this chapter.

where, p is a natural number selected by the analyst, $v_{l_{max}}$ and $v_{l_{min}}$ are the maximal and minimal attribute values of c_l, respectively, k_l is the importance coefficient of condition attribute c_l, and i is the number of condition attributes in a decision rule. It follows, the value of p determines the importance of the nearest rule. A small value of p allows a major difference with respect to a single condition attribute to be compensated by a number of minor differences, with regard to other condition attributes, whereas a high value of p will over-value the larger differences and ignore minor ones. Here, the values are set to, $p = 2$, and $k_l = 1$ for all l (equal importance amongst the condition attributes), thereby implying least squares fitting to a given rule. See Słowiński (1992) for full discussion on this measure.

This distance measure equation is demonstrated next, where Table 3.6.1 displays a set of rules used to classify an example set of seven objects in Table 3.6.2. The type of data set, training, validation, in-sample or out-sample, is of no consequence for this example. The strength and certainty in Table 3.6.1 could represent either, Strength and Certainty for a β-reduct, or Occurrence Strength and Occurrence Certainty for an aggregated β-reduct, again it is of no consequence for this simple classification example.

Rule		c_1		c_2		c_3		d_1	Strength	Certainty
1	If	1	and	-	and	1	then	0	0.5	1
2	If	-	and	1	and	1	then	0	0.2	1
3	If	1	and	-	and	0	then	1	0.1	1
4	If	-	and	0	and	-	then	0	0.2	1

Table 3.6.1: Rules Used to Predict Unseen Data in Table 3.6.2

Objects	c_1	c_2	c_3	Actual (d_1)	Predicted (d_1)	Predicted by rule:	Rule 1 (0) Distance	Rule 2 (0) Distance	Rule 3 (1) Distance	Rule 4 (0) Distance
o_1	1	1	1	0	0	1	0.0	0.0	0.5	1.0
o_2	0	1	0	0	0	2*	0.7	0.5	0.5	1.0
o_3	0	1	1	0	0	2	0.5	0.0	0.7	1.0
o_4	1	1	0	1	1	3	0.5	0.5	0.0	1.0
o_5	0	1	0	1	0	2*	0.7	0.5	0.5	1.0
o_6	0	0	1	1	0	4	0.5	0.5	0.7	0.0
o_7	0	1	0	1	0	2*	0.7	0.5	0.5	1.0

**Predicted by nearest rule, with highest rule strength*

Table 3.6.2: Objects Predicted by Rules from Table 3.6.1

Within Table 3.6.2, the 'Actual' column denotes the actual decision value of the object and the 'Predicted' column denotes the classification given either by, a rule that has matching condition attribute values, or the nearest rule in terms of distance (*dist*). The 'Predicted by rule' column displays which rule was used to predict an individual object. The last four columns give the distance of each rule to the individual objects (associated rule decision outcome d_1 within the parenthesis). Note that rules classifying objects that have matching condition attribute values, consequently have a distance of zero.

Where two or more rules have the same distance to an object, the object is classified by the rule with the highest strength; if the strengths are equal then the certainty is used to distinguish between the competing rules. The classification of the objects within Table 3.6.2, can be divided into four categories:

1. Those predicted correctly by a rule with matching condition values, objects o_1, o_3 and o_4.
2. Those predicted incorrectly by a rule with matching condition values, object o_6.
3. Those predicted correctly by the nearest rule, were there were no rules with matching condition values, object o_2.
4. Those predicted incorrectly by the nearest rule, were there were no rules with matching condition values, objects o_5 and o_7.

Tables similar to Table 3.6.2, are implemented in both the VPRS vein graph and re-sampling software. It provides the analyst with information about the classification results, such as, which rules are predicting correctly and which are predicting incorrectly.

3.7 Summary

This chapter has presented a broad overview of the methods utilised within data mining to evaluate the possible future performance of a given classifier. With particular attention given to estimating the true predictive accuracy through, the apparent predictive accuracy, the train and test methodology and re-sampling methods. Three sub-sampling methods were described, namely, random sub-sampling, stratified sampling and a novel statistical sampling method. The concepts of bias and variance have been discussed as means to distinguish between the different estimation methods.

Three re-sampling methods were described, namely, leave-one-out, *k*-fold cross-validation and bootstrapping. The bias and variance associated with each re-sampling method was discussed, and a method for adjusting the e0 bootstrap predictive accuracy estimate, to account for it being biased pessimistically, was described, namely, the 0.632B bootstrap.

Ensembles, as an approach to improve classifier performance, in terms of predictive accuracy and stability, were described, with particular attention given to bagging. A new, novel approach to β-reduct aggregation has been outlined which adapts the bagging method; but also allows aggregation of β-reducts through the two other described re-sampling methods, leave-one-out and *k*-fold cross-validation. A number of aggregated rule metrics were introduced, namely, Occurrence, Occurrence Support, Occurrence Correct, Occurrence Strength and Occurrence Certainty.

Finally, a demonstration of classification, using a decision table, applied to a validation set was given, and the classification of objects using the nearest rule method to classify objects, where there was no rule within a given rule set, that had matching condition attribute values, was illustrated.

Chapter 4

Data Pre-Processing and Feature Selection

This chapter outlines the methods implemented within the developed software, to facilitate the pre-processing and feature selection stages of the KDD process (as described in Chapter 1). Within this dissertation, the purpose of data pre-processing and feature selection, in the context of the VPRS analyses, is to improve the predictive performance and interpretability of the rules induced from the selected β-reducts.

It should be noted that pre-processing and feature selection, have been extensively studied within the extant literature (Liu et al., 2002b; Han and Kamber, 2006; Jensen and Shen, 2008;). The sections presented within this chapter focus on the four main areas as outlined below:

- Section 4.1. **Data Discretisation**. This section describes the process of discretising continuous valued data. It discusses the issues surrounding different discretisation approaches and describes four methods, which are implemented within the developed VPRS software.
- Section 4.2. **Feature Selection**. This section considers the issues surrounding feature selection, and describes two methods implemented within the developed pre-processing software.
- Section 4.3 **Balancing**. This section considers the problem of imbalanced data, and describes three basic methods for tackling an imbalanced data set, which are implemented within the

developed pre-processing software.

- Section 4.4. **Missing Data.** This section highlights the problem of missing data, and describes two methods of missing value imputation, which are implemented within the developed pre-processing software.
- Section 4.5. **Summary.** This section summarises the main methods described throughout the chapter, in particular those implemented in the VPRS pre-processing software.

Within this dissertation, when compared to the work presented on discretisation and feature selection, less emphasis has been placed on balancing and missing value imputation. Firstly because, feature selection and discretisation were more pressing pre-processing issues, that needed more attention to facilitate the data mining (VPRS) analyses, and secondly, because within the extant literature, less emphasises has been placed on solving the issues surrounding balancing and missing data (Weiss and Indurkhya, 1998).

4.1 Data Discretisation

Within data mining, some classification methods can only be constructed from, a training set based on discrete data (categorical, nominal and symbolic data), or require continuous valued attributes to be discretised into a finite number of intervals (Boullé, 2004). For example, decision tree induction methods, such as: ID3, C4.5 and C5/See5 (Quinlan, 1986, 1993, 2007) and Breiman et al.'s (1984) Classification and Regression Trees (CART), intervalise or discretise the data during the process. There are also a number of rule induction methods, that require the data to be discretised, such as: Cohen's (1995) RIPPER, Weiss and Indurkhya's (1998) Swap-1, and more relevant to this dissertation, rule induction based on RST (VPRS). Note though, with regards to RST, recent developments based on Fuzzy sets and Dominance Based Rough Set Theory have been developed that can facilitate continuous valued attributes (Jensen, 2004; Greco et al., 2005).

The tree induction methods, and the non-tree induction methods mentioned above, differ distinctly in the way they implement discretisation. That is, the tree induction methods such as ID3 and CART perform discretisation dynamically during the algorithm's process (Kerber, 1992). That is, discretisation is integral to the algorithms; whereas, methods such as RST require, as a prerequisite, for the data to be discretised before the analysis and construction of the classifier can be performed.

Despite the fact that the methods of discretisation described within this chapter may be relevant to both the dynamic and non-dynamic methods, because of the focus on VPRS within this dissertation, they will be described in the non-dynamic sense as part of the pre-processing phase. Hence, within the context of pre-processing presented in this chapter, discretisation is the process of dividing a continuous valued attribute into finite intervals, and recoding the data within those intervals into categoric integer values (e.g. 0, 1, 2 and so forth), prior to the VPRS analysis.[7]

4.1.1 Discretisation Basic Concepts

Manually dividing the continuous data into intervals is the simplest method of discretisation. For example, when discretising the attribute age, the discrete intervals may be set as, young (0, 30], middle aged (30, 60] and old aged (60, ∞). The advantage of manually setting the intervals, is that, where the analyst is knowledgeable of the attribute in question, they can set more intuitive intervals (for further information on the related subject of intuitive partitioning, see Han and Kamber, 2006).

Manually setting the discrete intervals for a large data set may be a slow task. Furthermore, it may not be immediately apparent, where the cut-points (values that separate the intervals) should be placed within the attribute's data range (although separate analysis of the data may aid the analyst's decision). Kerber (1992, pp. 123) notes that:

[7] Any data input into the developed VPRS analysis software, of a nominal or symbolic nature, will also be represented by numeric values.

> "While the extra effort of manual discretisation is a hardship, of much greater importance is that the classification algorithm might not be able to overcome the handicap of poorly chosen intervals."

However, there are numerous automated discretisation methods, which are faster and potentially better than the manual process, at identifying important cut-points within the data. There are simple automated methods, such as equal-width and equal-frequency (described next), which do not considered the distribution of a decision attribute's values, and there are, more advanced methods, which typically provide better intervalisation, that do consider the distribution of the decision class values. The more advanced methods are better, in the sense that, the resultant classifiers have an improved predictive performance. The following subsections describe some of the concepts relating to these methods.

4.1.1.1 Supervised and Unsupervised Discretisation

One of the simplest, and perhaps most naïve methods of discretisation,[8] namely equal-width, takes the minimum and maximum values from the continuous valued attribute and divides the range in between, into *k* intervals of equal-width (the number of required intervals is supplied by the analyst). As an example, taking the attribute salaries ranging from £20,000 to £100,000, and discretising it into four intervals of equal-width, the resultant intervals would be [£20,000, £40,000], (£40,000, £60,000], (£60,000, £80,000], (£80,000, £100,000]. A similar method, known as equal-frequency, seeks to divide the data between the minimum and maximum, into *k* intervals containing approximately the same amount of objects, for example if $k = 10$, then each interval would contain approximately 10% of the data.

The methods of equal-width and equal-frequency can perform poorly, because they do not consider the distribution of the decision attribute in relation to the continuous valued attribute being discretised. Kerber (1992, pp. 123) when discussing these methods, stated the following reason for their poor performance:

[8] When referring to discretisation, we are now referring to automated discretisation methods.

> "The primary reason that these methods fail is that they ignore the class of the training examples, making it very unlikely that the interval boundaries will just happen to occur in the places that best facilitate accurate classification."

To reiterate, Kerber is stating that the discretisation method should take into account the distribution of the decision attribute associated with the continuous value attribute being discretised, what is generally known as supervised discretisation, as opposed to unsupervised discretisation (Dougherty et al., 1995).

To illustrate, Figures 4.1.1.1.1 and 4.1.1.1.2 present, a visualisation of the equal-width discretisation method applied to a continuous valued attribute *X*(.) associated with a dichotomous decision attribute represented by the classes 'x' and 'o'.

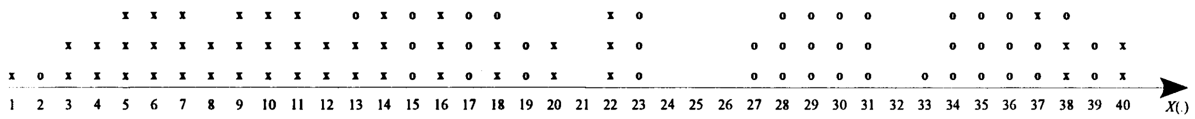

x x x x x x o x o x o o x o o o o o o o o o x o
x x x x x x x x x x x x o x o x o x x o o o o o o o o o o x o x
x o x x x x x x x x x x x x o x o x o x x o o o o o o o o o o o o x o x
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
X(.)

Figure 4.1.1.1.1: Distribution of the Decision Attribute Sorted into Ascending Order Based on the Continuous Value Attribute *X*(.)

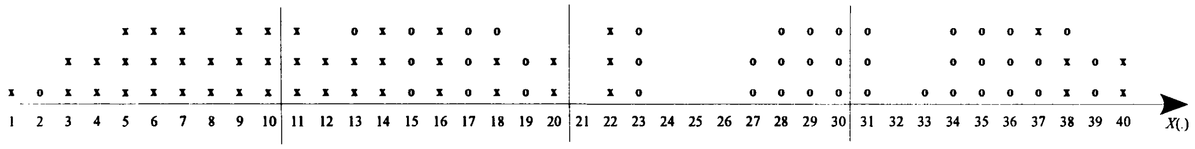

x x x x x x o x o x o o x o o o o o o o o o x o
x x x x x x x x x x x x o x o x o x x o o o o o o o o o o x o x
x o x x x x x x x x x x x x o x o x o x x o o o o o o o o o o o o x o x
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
X(.)

Figure 4.1.1.1.2: Illustrating the Equal-Width Discretisation of the Continuous Value Attribute *X*(.)

The top diagram in Figure 4.1.1.1.1, shows the distribution of the decision values associated with a sorted continuous valued attribute *X*(.), where the continuous valued attribute's range is from 1 to 40. The lower diagram illustrates the intervalisation of *X*(.) into four intervals of equal-width, namely, [1, 10], (11, 20], (21, 30], (31, 40] (as per the equal-width methodology). It is clear in Figure 4.1.1.1.2, that the intervals are not ideal, as they do not appear to split the distribution at the most "natural" positions (see for example either side of cut-point 30 and 31). That is, splitting the intervals so that they contain values from only one of the decision classes.

The question here, with regards to discretisation is, can having knowledge of the decision attributes distribution, be used to improve the interval discretisation? and indeed supervised methods have been developed that utilise the decision attribute distribution, some of which, are described in more depth in subsection 4.1.2.

4.1.1.2 Top-down and Bottom-up Discretisation

Most supervised discretisation methods can be said to be either a top-down or bottom-up process. If the process involves initially splitting a continuous value attribute into two or more intervals around a cut-point, and then recursively splitting those intervals into smaller intervals, the discretisation method is described as a top-down process. There are a number of methods for choosing the cut-points, including measures of entropy (randomness) and statistical measures (described in section 4.1.2). There are also a number of methods for terminating the recursive process, again entropic and statistical measures, or the analyst may predefine the maximum number of intervals they require. Examples of top-down methods include, Zeta (Ho and Scott, 1997, 1998), Adaptive Quantizer (Chan et al., 1991) and Minimum Class Entropy (MCE) (Fayyad and Irani, 1992).

Conversely, bottom-up processes, initially split the continuous value attribute's data into as many intervals as can be identified,[9] then merges the smaller intervals together, until a stopping criteria based on the optimisation of some measure has been satisfied. The stopping criteria may be a pre-defined minimal number of intervals, or based on entropic or statistical measures. In addition, the process will stop if the intervals are merged into a single interval. A single all inclusive interval would indicate that the attribute is of no information value to the analyst or the analytical process (Zighed et al., 1998). Examples of bottom-up merging methods include, the ChiMerge (Kerber, 1992), Chi2 (Liu and Setiono, 1997), ConMerge (Wang and Liu, 1998) and the FUSINTER method (Zighed et al., 1998).

[9] The method used for identifying the initial intervals is dependant on the discretisation method employed.

4.1.1.3 Global and Local Discretisation

The ChiMerge (Kerber, 1992) is an example of what is described as a local discretisation method. It is a local method, because during the interval merging process, the χ^2 Chi statistic is used to determine the quality of the discretisation taken across only two of the intervals being merged. Conversely, global discretion methods, select the intervals to be merged based on optimising a measure across the whole distribution (all intervals). Examples of global discretisation include FUSINTER (Zighed et al., 1998) and Zeta (Ho and Scott, 1997, 1998). The potential advantage of global discretion is its ability to avoid thin partitioning. Global methods typically identify less partitions than the ChiMerge or Minimum Class Entropy (described later), which are both local methods. Recently, Boullé (2004) suggested a global discretisation adaptation of the ChiMerge known as Khiops, which seeks to optimise the χ^2 Chi statistic across all intervals of an attribute being discretised. Top-down and bottom-up discretisation methods, can either be local or global distretisation methods, dependant on the particular method in question.

It is important to draw a distinction here, between the global methods suggested above, which are global at the attribute level, and methods such as that suggested by Chemielewski and Grzymala-Busse (1995, 1996), which are global on the data set level (all attributes). They present a wrapper method which utilises elements of RST, that can "globalise" local discretion methods. In a sense, their method, is much more of a global method, as they suggest using the Quality of Classification (they term it dependency or consistency) to optimise a local discretion method across all attributes simultaneously (the other methods described here, discretise attributes independently of each other). Integral to their method was the basic principle that the attributes' domains after discretisation should be as simple as possible (few intervals), because simpler rules can be induced from the discretised data, and that these rules encompass more general trends. It should be noted though, that their method inherently assumes that all attributes, not a feature subset of those attributes, will be used in any subsequent analysis (feature subsets are described in section 4.2).

4.1.2 Supervised Discretisation Methods

Equal-width and equal-frequency are examples of unsupervised "binning" discretisation methods. Whereby, the analyst pre-selects the number of intervals required and the continuous values lying within those intervals are simply recoded to discrete values.

Not only are equal-width and equal-frequency limited by being unsupervised methods, but by being binning methods, as they are sometimes referred to (Kotsianti and Kanellopoulos, 2006), they are sensitive to the number of intervals (bins) the analyst chooses. That is, the number of intervals may effect the performance of, any subsequently constructed classifier.[10] They are also sensitive to outlier values, which are attribute values, that do not comply with the general behaviour of the attribute in question (Kantardzic, 2003; Han and Kamber, 2006).

There are however, a number of supervised discretisation methods which seek to optimise the intervalisation of the continuous value attribute through, the optimisation of an entropic or statistical measure, calculated on the distribution of the associated decision attribute values. Here, for brevity, only the more popular methods are highlighted.

Perhaps one of the most influential supervised methods, and now popular amongst the related literature as a bench mark method, is the ubiquitous ChiMerge (Kerber, 1992). Kerber highlighted the short comings of the available "binning" methods, and that, the more advanced discretisation methods at that time, were integral to the specific classifier methods (integral, as in dynamic discretisation described in the introduction of section 4.1). Hence, Kerber presented the ChiMerge as a supervised bottom-up (merging) method; proposed for use as a pre-processing step in machine learning. Kerber (1992, pp. 126) discussed the robustness of the ChiMerge over that of the other available discretisation methods and stated:

> "...ChiMerge will seldom miss important intervals or choose an interval boundary when there is obviously a better choice. In contrast, the equal-width-intervals and equal-frequency-intervals methods can produce extremely poor discretisation for certain

[10] 1R (Holte, 1993) and maximal marginal entropy (Dougherty et al., 1995), are both supervised adaptations of the equal-width and equal-frequency methods respectively. That seek to improve the intervalisation by utilising the decision class information.

> attributes..."

Conversely, Boullé (2004) mentions the ChiSplit as a top-down alternative to the ChiMerge, which splits intervals recursively whilst trying to optimise the Chi statistic.

There are a number of other methods similar to the ChiMerge, that either merge or split intervals based on the optimisation of a measure based on the distribution of the associated decision attribute values. Examples include, discretisation methods based on the Gini index (Zhang et al., 2007), Akaike Information Criterion (AIC) and Baysian Information Criterion (BIC) (Hand et al., 2001). Interestingly, Jin et al. (2007, pp. 183) claim to prove analytically, that:

> "...discretization methods based on informational theoretical complexity and the methods based on statistical measures of data dependency are asymptotically equivalent."

They devise a generalised function and showed that discretisation methods involving the Gini index, AIC, BIC, and the X^2 Chi statistic, are all derivable from this generalised function. Furthermore, they propose (pp. 183) a, "...dynamic programming algorithm that guarantees the best discretization...", based on the utilisation of their generalised function.

Here, the supervised discretisation methods selected for implementation seek to optimise a measure of entropy associated with the distribution of the decision attribute values. The MCE method (Fayyad and Irani, 1992), was selected as a representative of the top-down and local methods. It was one of the earliest (if not the earliest) examples of top-down entropic discretisation and is based on Shannon's entropy. FUSINTER (Zighed et al., 1998) was selected to represent the bottom-up and global (attribute level) discretisation methods, and is based on quadratic entropy. In addition, equal-width and equal-frequency are implemented as representatives of the unsupervised methods.

4.1.3 Implementing MCE and FUSINTER

The following two subsections describe the MCE and FUSINTER algorithms in detail. The descriptions presented here are based on those given in Fayyad and Irani (1992) and Zighed et al. (1998). To describe both methods, firstly some general notation needs formalising. Thus, where U is the universe of objects, let $X(.)$ be the continuous value attribute to be discretised, where for any object w, $X(w)$ represents a specific value. $Y(.)$ represents the decision class values associated with the object set, where $Y(w)$ represents the class value associated with a specific object w.

Therefore, let D_X be the distribution of $X(.)$ (also called the definition field). Where discretising the attribute $X(.)$ is to intervalise D_X with a set of interval cut-point (threshold) values d_j. Hence, we obtain k intervals I_j, $j = (1,\ldots, k;\ k \geqslant 2)$ such that:

$$I_1 = [d_0,\ d_1],\ldots,\ I_j = [d_{j-1},\ d_j),\ \ldots,\ I_k = [d_{k-1},\ d_k].$$

Once the interval threshold values have been identified, the continuous attribute $X(.)$ is replaced by a categorical attribute $\tilde{X}(.)$, which takes its values in the set $\{1,\ldots, k\}$. Thus, $\forall\, w \in U$ if $d_{j-1} \leqslant X(w) < d_j$ then $\tilde{X}(w) = j$.

Let n_{ij} be the number of training objects which are in the interval I_j and which belong to the decision class y_i, hence $n_{ij} = \text{Card}\{w \in U\colon\ X(w) \in I_j,\ Y(w) = y_i\}$.

Let $n_{.j}$ be the number of objects which are in the interval I_j, $n_{.j} = \sum_{i=1}^{m} n_{ij}$.

Let $n_{i.}$ be the number of examples of the decision class y_i, $n_{i.} = \sum_{j=1}^{k} n_{ij}$.

Let n be the number of objects in the sample, $n = \sum_{j=1}^{k} n_{.j}$, or $n = \sum_{j=1}^{k} n_{i.}$.

Supervised discretisation methods, such as MCE and FUSINTER, attempt to optimise the discretisation of the attribute $X(.)$ based on the decision attribute values of $Y(.)$. Here, we are attempting to discretise the data in such a way that it improves the predictability of the decision class $Y(.)$ based on $X(.)$, hence it is desirable to identify discretisation cut-points where the intervals contain exclusively objects associated with the same decision class value (see later in Figure 4.1.3.1.2).

4.1.3.1 Minimum Class Entropy

Minimum Class Entropy (MCE) was proposed by Fayyad and Irani (1992), it is actually an extraction from the ID3 decision tree induction method (as stated previously, some discretisation methods were originally integral to the classifier method involved). MCE is a top-down approach, based on Shannon's entropy measure (Shannon, 1948), as shown in Equation 4.1.3.1.1:

$$Ent(S)=-\sum_{i=1}^{m}\frac{n_{ij}}{n_{.j}}log_2\frac{n_{ij}}{n_{.j}}, \qquad (4.1.3.1.1)$$

where S is a subset of objects in U. Within information theory, Shannon's entropy or information entropy is a measure of the uncertainty associated with a random variable. Here, with regards to discretisation, uncertainty is considered in terms of the impurity or randomness of the decision attribute's distribution over two intervals.

The algorithm below outlines a summary of the MCE process. MCE can be applied to a multiclass problem, but here, we continue to consider, only the simpler dichotomous decision class example as shown in Figure 4.1.3.1.1. The method initially identifies all possible interval cut-points (values $X(w)$) (steps 1 to 3). The process then recursively splits intervals, identifying each cut-point by selecting the cut-point which shows the lowest entropy value over the interval being split (steps 4 and 5). The process is terminated at such a point that inserting more cut-points cannot achieve a lower entropy value over any of the intervals, or the number of intervals has reached the amount specified by the analyst, prior to the discretisation (step 5).

MCE Algorithm Pseudo Code

1. The distribution D_x, is formed by sorting all objects into ascending order according to the increasing values of $X(.)$, making runs of points identified by their decision class x or o (see Figure 4.1.3.1.1).
2. Each run of points of the same decision class forms an interval.
3. If several decision classes are superposed on the same value of $X(.)$ (such as the value 13 shown in Figure 4.1.3.1.1, where there are three associated decision values, two x's and one o), then the associated interval will be reduced to this unique value and unlike other

intervals, this one will contain a mixing of classes. The set *K*, of *k* possible intervals is recorded.

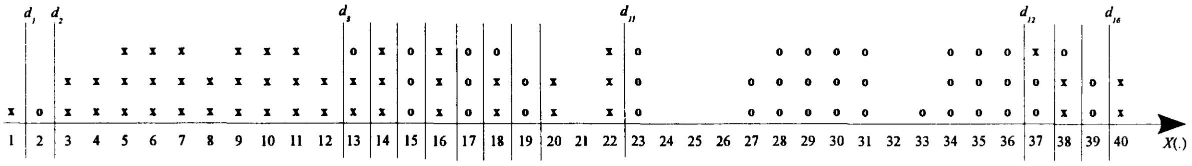


Figure 4.1.3.1.1: Initial Identification of Interval Cut-points within the Distribution D_x.

4. For an attribute *X*(.), where *S* is a subset of objects in *U*, and d_j a cut-point value from *K*, let $S_1 \subset S$ be the subset of examples in *S* with attribute values in *X*(.), not exceeding the cut-point value d_j and $S_2 = S - S_1$. The class information entropy of the partition induced by d_j, denoted $E(X(.),\ d_j,\ S)$, is defined by Equation 4.1.3.1.2:

$$E(X(.),\ d_j,\ S) = \frac{|S_1|}{|S|} Ent(S_1) + \frac{|S_2|}{|S|} Ent(S_2). \qquad (4.1.3.1.2)$$

An interval cut-point is selected from *K*, by selecting the cut-point that optimises the 'bi-partitioning' of an interval. That is, the cut-point that minimises the value $E(X(.),\ d_j,\ S)$ but where $E(X(.),\ d_j,\ S) < Ent(S)$.

5. The fifth step is repeated on each of the sub-divisions recursively, until no improvement is possible (i.e. $E(X(.),\ d_j,\ S) \geqslant Ent(S) \forall d_j \in K$) or a maximum number of intervals specified by the analyst has been reached (for MCE and other entropic feature selection stopping criteria, see Liu et al., 2002a). The resulting intervalisation of the distribution shown in Figure 4.1.3.1.1 is presented in in Figure 4.1.3.1.2.

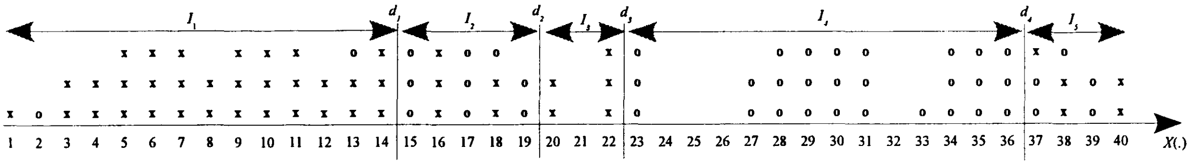


Figure 4.1.3.1.2: Illustrating the MCE Discretisation of the Continuous Value Attribute *X*(.)

Within Figure 4.1.3.1.2 five intervals based on the optimisation of an entropic value have been identified. Note that, the intervals are not necessarily the same width apart, as would be the case with the equal-width method (see Figure 4.1.1.1.2), and that the intervals do not necessarily contain the same amount of objects.

4.1.3.2 FUSINTER

FUSINTER is a bottom-up discretisation method introduced by Zighed et al. (1998). To describe the FUSINTER method, firstly some further notation must be introduced. Each discretisation into k intervals I_j, $j = (1,\ldots,k;\ k \geqslant 2)$ (including the initial discretisation), can be associated with a matrix T of m rows and k columns. The rows correspond to the classes and the columns the intervals, as shown below:

$$T = \begin{pmatrix} n_{11} & n_{12} & \cdots & n_{1k} \\ n_{21} & n_{22} & \cdots & n_{2k} \\ \vdots & \vdots & \vdots & \vdots \\ n_{m1} & n_{m2} & \cdots & n_{mk} \end{pmatrix}.$$

Also note that a single column (interval) can be referred to as $T_j = \begin{pmatrix} n_{1j} \\ n_{2j} \\ \vdots \\ n_{mj} \end{pmatrix}$, and as such T may be referred to as $T = (T_1,\ldots,\ T_j,\ldots,\ T_k)$.

Within the following algorithm, the discretisation seeks to minimise a criterion $\varphi(T)$, where $\varphi(T)$ represents quadratic entropy, as defined in Equation 4.1.3.2.1 (for further explanation, see Zighed et al., 1992, 1998):

$$\begin{aligned} \varphi(T) &= \sum_{j=1}^{k} \alpha \frac{n_{.j}}{n} \left(\sum_{i=1}^{m} \frac{n_{ij}+\lambda}{n_{.j}+m\lambda} \left(1 - \frac{n_{ij}+\lambda}{n_{.j}+m\lambda} \right) \right) + (1-\alpha) \frac{m\lambda}{n_{.j}}, \\ &= \sum_{j=1}^{k} \alpha H_j(h,\ \lambda) + (1-\alpha) \frac{m\lambda}{n_{.j}}. \end{aligned} \qquad (4.1.3.2.1)$$

Where α and λ are variables that control the performance of the discretisation, Zighed et al. (1998) state that the $(1-\alpha)\frac{m\lambda}{n_{.j}}$ term penalises for over-splitting, that is, the term penalises for discretising into too many intervals, containing potentially fewer objects. The criterion $\varphi(T)$ is essentially a compromise between the purity measure $H_j(h,\lambda)$ and the splitting measure $\frac{m\lambda}{n_{.j}}$. Informally, α can be considered the degree of emphasis on purity over, over splitting, whereas λ is the actual degree to which we want to penalise for over splitting.

Zighed et al. (1998) suggests both a cross-validation approach and a more involved analytical approach, for finding suitable values of α and λ. In the extreme case, and based on an analytical example, they set $\alpha=0.95$ and $\lambda=0.61$, but state that, based on an experimental approach (we assume they are referring to cross-validation), they found that $\alpha=0.975$ and $\lambda=1.0$ were a good compromise between purity and interval size.

Here, also through experimentation, we found that setting $\alpha=0.97$ and $\lambda=0.9$, puts slightly less emphasis on impurity and hence emphasises more on over-splitting, but penalises less for it. These values appear to influence the FUSINTER algorithm into discretising attributes into less intervals, allowing for more general trends to be established within the discretisation (for reasons similar to, and stated earlier, given by Chemielewski and Grzymala-Busse, 1996).

The points of the FUSINTER method are described in the pseudo code below. FUSINTER, similarly to MCE, firstly identifies all possible cut-points (steps 1 to 3), but in contrast to MCE, FUSINTER initially inserts all cut-points and iteratively removes them until the $\varphi(T)$ value is optimised (steps 4 to 6). The essential difference between MCE and FUSINTER (besides FUSINTER being bottom-up and MCE being top-down) is that FUSINTER takes the value $\varphi(T)$ across all intervals (hence global at the attribute level), whereas MCE calculates Shannon's entropy across two intervals (hence a local method).

FUSINTER Algorithm Pseudo Code

1. The distribution D_x, is formed by sorting all objects into ascending order according to the increasing values of $X(.)$, making runs of points identified by their decision class x or o (see Figure 4.1.3.1.1).
2. Each run of points of the same decision class forms an interval.
3. If several decision classes are superposed on a same value of $X(.)$, then the associated

interval will be reduced to this unique value and unlike other intervals, this one will contain a mixing of classes. The set *K*, of *k* possible intervals are recorded.

4. Let us suppose that the initial discretisation provides k intervals, and the matrix T is deduced of m rows and k columns that allows for the calculation of the criterion $\varphi(T)$, $T = \left(T_1, \ldots, T_{(j-1)}, T_j, \ldots, T_k\right)$.
5. Iteratively, search for the two adjacent intervals whose merging would improve the value of the criterion, that is *j* such as:
$$\varphi(T) - \varphi(\ldots, \{T_j + T_{(j+1)}\}, \ldots) = Max_{i=1}^{k-1}(\varphi(T) - \varphi(\ldots, T_i + T_{(i+1)}, \ldots))$$
6. If:
$$\varphi(T) - \varphi(T_i, \ldots, T_j + T_{(j+1)}, \ldots, T_k) > 0$$
then, the two intervals I_j and $I_{(j+1)}$ are merged.
7. The process is repeated from step 2, with $k-1$ intervals, until no improvement is possible or k reaches the value 1. If the process stops with $k = 1$, it indicates that the discretisation of *X*(.) is of no interest for the determination of *Y*(.).

The results of the FUSINTER algorithm described here, with regards to Figure 4.1.3.1.1, were identical to those results shown in Figure 4.1.3.1.2, originally found in Zighed et al. (1998).

4.2 Feature Selection

For the purpose of machine learning and classifier construction, it is often necessary to reduce the number of attributes associated with a given data set. Data sets may contain many more attributes than are necessary for the purpose of classification, where attributes may be irrelevant or redundant. Irrelevant attributes have no bearing on the classification, for example nationality is irrelevant to classifying a persons gender. An attribute that conveys similar, or the same information as another attribute, may be considered redundant (referred to as collinear in statistics). That is, a redundant attribute brings no additional information for the purpose of classification (Kantardzic, 2003; Liu and Motoda, 1998, 2008).

It is possible for an expert to select the set of most relevant attributes, but this can be time consuming, and the analyst must have a thorough knowledge on the subject domain. Additionally, where a good selection of attributes can lead to increased classifier accuracy, selection of a few irrelevant or redundant attributes can be detrimental. Han and Kamber (2006, pp. 75) states:

> "Leaving out relevant attributes or keeping in irrelevant attributes may be detrimental, causing confusion for the mining algorithm employed. This can result in discovered patterns of poor quality. In addition, the added volume of irrelevant or redundant attributes can slow down the mining process."

Hence, automated approaches, known as feature selection or attribute selection methods, seek to find the optimal set of attributes for use in classification. Given n attributes, there are 2^n possible subsets, hence searching for the best set of attributes can prove computationally expensive and time consuming. There are however, a number of sub-optimal feature selection methods that perform faster. Subsection 4.2.1 describes some of the issues relating to automated feature selection methods.

4.2.1 Feature Selection Basic Concepts

Feature selection methods, in general, can be ordered into a number of categories. Here, we describe the main categories pertinent to the methods implemented in this dissertation. For more thorough overviews, see Jensen (2004), Liu (2008) and Liu and Motoda (1998, 2008).

4.2.1.1 Filter Based and Wrapper Based Feature Selection

If a subset of attributes are selected prior to, and independent of any particular classifier method, then the feature selection algorithm is described as a filter method. Examples include ReliefF (Robnik-Šikonja and Kononenko, 2003), Contextual Merit (Hong, 1997; Puuronen et al., 2001) and Focus (Almuallim and Dietterich, 1991).

Feature selection that integrates the classifier construction method in question, is described as a "wrapper" approach (Hall, 1999, 2000). For example, the Las Vegas algorithm for Wrapper feature selection (LVW) was proposed by (Liu and Setiono, 1998b), and is a wrapper version of their Las Vegas algorithm for Filter feature selection (LVF) (Liu and Setiono, 1998a). Wrapper methods typically incorporate some measure of classifier performance (such as predictive accuracy) to evaluate the set of attributes. Note, the similarity in the distinction drawn between wrapper and filter methods and, dynamic and non-dynamic discretisation methods, described previously (section 4.1).

Although the wrapper approach may produce better results in terms of selection, that is, the best subset of attributes that provides for the best results in terms of classification accuracies, they can be resource intensive (computing memory and processing time), and may not be robust enough to deal with larger data sets (Liu and Motoda, 2000). An exhaustive search of reducts within RST is a good example of a feature selection method that breaks down when a larger number of attributes are involved (more will be discussed about this issue later in subsection 4.2.2). Filter based methods are less optimal, but are generally suitable for most classifier methods requiring an element of feature selection (Lui and Motoda, 2000).

4.2.1.2 Forward Selection, Backward Elimination and F/B Combination

If the feature selection method in question initially starts with an empty set as the subset of selected attributes, and iteratively adds attributes from the remaining set of attributes until a stopping criteria is met, then the method is described as forward selection. Zhou et al. (2006) describe a modern forward selection method called Streamwise feature selection, which has an advantage over other methods, in that, it does not assume that all attributes are known in advance. In their method, features can be generated dynamically to aid the feature selection search for the most promising attributes.

Conversely, if the method initiates with all possible attributes as the determined subset and iteratively removes attributes until some criteria is met, then it is described as a backward elimination method. Song et al. (2007) describe a modern backward elimination method based on the Hilbert-Schmidt Independence Criterion (HSIC) as a measure of dependence between the attributes. They also state that, a more efficient forward selection version of their algorithm is derivable, but that it yields slightly poorer results.

Additionally, some feature selection methodologies, use a combination of forward selection and backward elimination. Possibly including random selection of attributes, or combinations of the methods described here (see Liu and Motoda, 2000, pp. 78-80, for further reading).

4.2.1.3 Single Feature and Stepwise Feature Selection

Some feature selection algorithms select features on the principal that they are independent of each other (mainly earlier methods), a simple check of condition and decision attribute correlation being one example (for further reference on earlier feature selection methods see, Weiss and Kulikowski, 1991).

Clearly, attributes are not independent in all examples, hence, stepwise feature selection methods identify dependencies by sequentially determining the 'next' attribute to select, based on a criteria involving the previously selected attributes. In most circumstances, stepwise feature selection is considered better than independently selected features, but because backtracking is not used, stepwise feature selection is still a suboptimal solution (Weiss and Kulikowski, 1991),[11] a combination of forward and backward stepwise selection, may be one solution in facilitating backtracking. Note that by their nature, forward and backward selection/elimination are stepwise feature selection methods. Indeed, Han and Kamber (2006) only refer to the distinctions as 'stepwise forward' selection and 'stepwise backward' elimination.

[11] The only true guarantee of an optimal solution being an exhaustive search through the combinatoric domain of all attributes.

4.2.2 Reducts and β-reducts as a Method of Feature Selection

Reducts or β-reducts constitute the feature selection elements of RST and VPRS respectively, but as discussed in Chapter 2, finding all possible reducts/β-reducts would require searching the entire search space of the full set of attributes, that is, searching the full 2^n possible combinations. Theoretically it is possible, but in reality the analyst is limited by computing power and time constraints. Note, that the addition of an extra attribute to the full set of attributes affectively doubles the search space and hence the search time. That is, the search time grows exponentially as additional attributes are considered. Time efficient heuristics for the discovery of reducts/β-reducts are available, but as stated within Chapter 2 they are not appropriate within the VPRS analysis framework presented here (see Chapter 2 section 2.1.1 for further explanation).

Experimentation on the developed VPRS analysis software,[12] indicated that a maximum of twelve attributes appeared to be the reasonable limit on the developed software's capabilities. Although, this is dependant on a number of issues, including: levels of discretisation, size of the training set, computing power and the analyst's own time constraints. Thus, it may be plausible to analyse more attributes.

Hence here, we suggest a compromise, that employs feature selection methods implemented within the pre-processing software phase of the developed software, that pre-determines important attributes for selection prior to the VPRS analysis. In the role as pre-selection methods, the feature selection methods implemented and described next, may go some way to identifying irrelevant or redundant attributes, hence reducing the number of attributes that need consideration by the analyst and passing into the subsequent VPRS analysis (note the analyst has the final decision, on what attributes to pass into any subsequent VPRS analyses).

[12] Tested on a 1.6GHz Intel Centrino laptop with 512MB of RAM

4.2.3 Feature Selection Methods Selected for Implementation

Two feature selection methods have been implemented within the developed VPRS pre-processing software. Namely, ReliefF (Kononenko, 1994) and a novel feature selection method suggested by Beynon (2004) which utilises the concepts of RST and VPRS (for brevity it will be refereed to as RST_FS).

ReliefF, a filter based method proposed by Kononenko (1994), has been selected here, because of its efficiency (in terms of processing speed) and its ability to recognise dependencies between condition attributes. ReleifF is an unusual feature selection method (Liu and Motoda, 2000), as it does not seek to find a subset of attributes, but rather ranks all attributes based on their ability to distinguish between decision classes. ReleifF's unique methodology (described next) excludes its categorisation into any of the conventional feature selection categories of forward selection, backwards elimination, or independent and stepwise feature selection. ReliefF is based on the *k*-nearest neighbour classification approach and works well on similar classification algorithms (Kohavi and John, 1997). Although it is an efficient algorithm and good at distinguishing relevant features, unfortunately it is less adept at removing (giving a lower rank to) redundant (correlated) features (Kohavi and John, 1997; Liu and Yu, 2002)

RST_FS was selected as an alternative method to ReliefF, as it is a more conventional, filter based, sequential forward approach, that considers the dependency between sets of selected condition attributes and the decision attribute. Unlike ReliefF, it only selects a subset of attributes and does not attempt to rank all attributes. In some respects this is an advantage. That is, it indicates to the analyst, the number of attributes required within the determined subset (the subset having comparable classification performance to the full set of attributes, based on QoC, described next). Additionally, it was felt that RST_FS may be more pertinent as a feature selection method in respect to the VPRS model, as its underlying methodology is based on principles similar in concept to β-reduct selection.

4.2.3.1 ReliefF

Kira and Rendell (1992a, 1992b) on the realisation that the majority of the heuristic feature selection approaches of the time, assumed independence of condition attributes, proposed Relief as a new filter approach that recognised possible dependencies between attributes. The central idea of the Relief algorithm was based around assessing the suitability of an attribute by how well it distinguished between objects within different decision classes. Relief is generally accepted as a successful feature selection algorithm (Deitterich, 1997).

Relief was later expanded by Kononenko (1994), from a feature selection method that could only cope with dichotomous decision classes, to a methodology that could handle multiple decision class problems, and hence, renamed as ReliefF. A further adaptation, RReliefF, was also proposed by (Robnik-Šikonja and Kononenko, 1997) to accommodate the field of statistical regression, where the condition and decision attributes are represented by continuous (non-discrete) data. More recently, Liu et al. (2002b) proposed an alternative approach to ReleifF for the handling of multi decision class problems using selective sampling in what they called ReliefS. Here, we will be focusing on the original ReliefF.

Essentially, ReliefF works on the same principle as Relief. With reference to the algorithm shown below (originally from Robnik-Šikonja and Kononenko, 2003). The analyst defines the number of iterations m (line 2). For each iteration l in m, an object o_r where $r = (1,\ldots, n)$, is randomly selected from the training set of n possible objects $\{o_1,\ldots, o_n\}$ (line 2.1). ReliefF searches for the object's k nearest neighbours from the same decision class $d_j \in D$ where $D = \{d_1,\ldots, d_j\}$, called nearest hits H; and k nearest neighbours from each of the other decision classes, called nearest misses M (lines 2.2 to 2.3.1.1). The search criterion is based on the Manhattan distance between the objects, derived from their associated set of condition attribute values $\{value(a_1),\ldots, value(a_i)\}$, equated by Equations 4.2.3.1.2 and 4.2.3.1.3. $W = \{wa_1,\ldots, wa_i\}$ is the quality estimation (weight) associated with each condition attribute, updated for every iteration *l* in *m,* using Equation 4.2.3.1.1 (line 2.3.2.1). The quality estimation for each attribute indicates its ability to distinguish between objects from the same decision class (nearest hits) and objects from the other decision classes (nearest misses).

When searching for a suitable attribute, if the distance is large between the random object o_r and any of the *k* objects from the same decision class (nearest hits), this is an undesirable characteristic. Hence, the term α in Equation 4.2.3.1.1 reduces the weight wa_i associated with any given condition attribute a_i, penalising proportionally for larger distances. Where the distance is large between the random object o_r and any of the *k* objects from the other decision classes (nearest misses), this is a desirable characteristic. Hence, the term β in Equation 4.2.3.1.1 increases the weight wa_i associated with any given condition attribute a_i, rewarding proportionally for larger distances (lines 2.3.2 to 2.3.3).

This process is repeated *m* times, selecting random objects o_r and adjusting the quality estimation values in *W.*

<u>ReliefF Algorithm Pseudo Code</u>

```
1. Set all weights $W$ := 0.0;
2. For $l$ := 1 to $m$ do
   2.1. Randomly select an instance $o_r$;
   2.2. Find $k$ nearest hits $H$;
   2.3. For $j := 1 \text{ to} |D|$ do
      2.3.1. If $d_j \neq decionclass(o_r)$ do
         2.3.1.1. From class $d_j$ find $k$ nearest misses $M(d_j)$;
      2.3.2. For $i := 1 \text{ to} |A|$ do
         2.3.2.1. $a_i := a_i - \alpha + \beta$, where:
              $\alpha = \sum_{j=1}^{k} \left\{ \frac{\text{diff}(a_i, o_r, H)}{mk} \right\},$
              $\beta = \sum_{d_j \neq class(o_r)}^{k} \left\{ \frac{P(d_j)}{1 - P(class(o_r))} \left\{ \sum_{j=1}^{k} \frac{\text{diff}(a_i, o_r, H)}{mk} \right\} \right\}.$          (4.2.3.1.1)
      2.3.3. End For;
   2.4. End For;
3. End For;
4. Output $W$;
```

Where, if the current attribute values associated with two objects are discrete:

$$\text{diff}(a_i, o_1, o_2) = \left\{ \begin{array}{l} 0: value(a_i, o_1) = value(a_i, o_2) \\ 1: \text{otherwise} \end{array} \right\}, \qquad (4.2.3.1.2)$$

and if the current attribute values associated with two objects are continuous:

$$\text{diff}(a_i, o_1, o_2) = \frac{|value(a_i, o_1) - value(a_i, o_2)|}{MaxValue(a_i) - MinValue(a_i)}. \qquad (4.2.3.1.3)$$

With reference to the setting of the parameters, *m* and *k*. In Robnik-Šikonja and Kononenko (2003), based on empirical evidence they found that twenty to fifty iterations (*m*) were necessary. They also demonstrate an example that requires 300 iterations. From our experimentations we found these recommended values too low and had to set a much higher value of *m*. This issue is better discussed within the context of the pre-processing results, hence it is described in more detail in Chapter 6 section 6.3.1.

With regards to setting the *k* value, Robnik-Šikonja and Kononenko (2003) reported that, setting a value of *k* too high, can be detrimental to the identification of dependencies between condition attributes. Conversely, they also stated that, setting *k* too low may be detrimental, because the method would not be robust enough to handle noisy or complex data. Hence, they propose a default value of $k = 10$. This value appeared satisfactory for our analyses and is supported by other studies (Hall, 2000).

Here, the implemented ReliefF algorithm within the developed software will be applied to both the continuous pre-discretised training data, and the post-discretised training data, with both sets of results being presented to the analyst. As such, to distinguish between both sets of the algorithm's results, they were designated RefliefFC and ReliefFD, relating to the continuous and discrete versions of the training data, respectively. Furthermore, the pre-processing software will present three graphs. Firstly a graph that shows, the weight values associated with each condition attribute over a range of values of *m* up to, $m = n$. A second graph that shows the difference in weight values between each consecutively calculated weight value, associated with each condition attribute over a range of values of *m* up to $m = n$. Then a third graph that displays the variance in rank positions for each of the condition attributes based on their associated weights over a range of values of m up to $m = n$ (see later Chapter 6 subsection 6.3.1).

4.2.3.2 RST_FS, a Feature Selection Algorithm Based on Rough Set Theory

The feature selection algorithm referred to here as RST_FS is an implementation of the original work proposed by Beynon (2004), who described an iterative procedure for β-reduct selection within the VPRS model. The method is similar to the QuickReduct (Chouchoulas and Shen, 2001) and the ReverseReduct (Chouchoulas et al., 2002), and is a suboptimal approach to finding β-reducts. Here, the procedure is re-considered as a feature selection method.

As described previously, the method was selected for implementation because of its stepwise approach, which considers dependencies between the condition attributes and the decision attribute. It has also been selected for its association with the β-reduct identification method integral to the developed VPRS software (vein graph or re-sampling).

To describe the RST_FS method, the notation and concepts relating to VPRS described in Chapter 2 are required here. Hence, describing the RST_FS method with reference to the algorithm outlined below. Individual QoCs can be calculated for each β-interval (intervals described on line 2 of the pseudo code) associated with the full set of condition attributes C and the decision attribute set D; where $\gamma^{\frac{1}{2}(\beta^{C}_{i,1}+\beta^{C}_{i,2})}(\mathrm{C},\mathrm{D})$ $i = 1,\ldots,|\beta^{\mathrm{C}}|$ defines the QoC associated with each individual β-interval. RST_FS seeks to find a subset of condition attributes P that has, as close to as is possible, the same QoC over each subdomain of β, as the full set of attributes C.

An attribute set A, is initially set equal to the full set of attributes C (line 3). Attribute a_j is selected individually from A, and its suitability as a possible augmentation to the determined set of attributes P (initially empty) is tested, by setting a test set T, equal to P for each iteration and augmenting T with the current selected attribute a_j. A distance measure shown at Equation 4.2.3.2.1

is then calculated on the set T (lines 4.1 to 4.2). The attribute a_j that offers the best result in terms of minimising the distance measure between the QoCs associated with the full set of attributes C and the test set T is recorded, removed from A and augmented to P. The process is repeated until either the measure equals zero, in which case the subset P of attributes is an exactitude of C, in terms of β domains and associated QoC; or the set of attributes A is exhausted and no attributes remain to be augmented to P (lines 4, 4.5. and 5).

<u>RST_FS Algorithm Pseudo Code</u>

```
1. Let $P$ equal the initially empty set of selected attributes;
2. Identify the $\beta$-intervals associated with $C$, defined by:
   $\beta^C:(\beta^C_{1,1}, \beta^C_{1,2}], (\beta^C_{2,1}, \beta^C_{2,2}], \ldots, (\beta^C_{i,1}, \beta^C_{i,2}], \ldots, (\beta^C_{|\beta^C|,1}, \beta^C_{|\beta^C|,2}];$
3. Let the set $A := C$        //$A$ is the set of potentially selected attributes $\{a_1, \ldots, a_j\}$;
4. While $A$ not empty do
   4.1. For $j := 1$ to $|A|$ do
        4.1.1. Let the test set $T := P$;
        4.1.2. $T := T \cup \{a_j\}$;
        4.1.3. Identify the $\beta$-intervals associated with $T$, defined by:
               $\beta^T:(\beta^T_{1,1}, \beta^T_{1,2}], (\beta^T_{2,1}, \beta^T_{2,2}], \ldots, (\beta^T_{i,1}, \beta^T_{i,2}], \ldots, (\beta^T_{|\beta^T|,1}, \beta^T_{|\beta^T|,2}];$
        4.1.4. Merge the $\beta$-intervals associated with $C$ and $T$ to obtain the combined intervals
               defined by:
               $\beta^{T\cup C}:(\beta^{T\cup C}_{1,1}, \beta^{T\cup C}_{1,2}], (\beta^{T\cup C}_{2,1}, \beta^{T\cup C}_{2,2}], \ldots, (\beta^{T\cup C}_{i,1}, \beta^{T\cup C}_{i,2}], \ldots, (\beta^{T\cup C}_{|\beta^{T\cup C}|,1}, \beta^{T\cup C}_{|\beta^{T\cup C}|,2}];;$
        4.1.5. Calculate the associated distance measure over the combined intervals, defined
               by $\Delta\gamma^{\beta}_{T\cup C}(T, D)$ and given by Equation 4.2.3.2.1:
   $\Delta\gamma^{\beta}_{T,C}(T, D) = 2\sum_{i=1}^{|\beta^{T\cup C}|}(\beta^{T\cup C}_{i,2} - \beta^{T\cup C}_{i,1}) \times |\gamma^{\frac{1}{2}(\beta^{T\cup C}_{i,1}+\beta^{T\cup C}_{i,2})}(T, D) - \gamma^{\frac{1}{2}(\beta^{T\cup C}_{i,1}+\beta^{T\cup C}_{i,2})}(C, D)|;$     (4.2.3.2.1)
        4.1.6. If $j = 1$, set $pv := \Delta\gamma^{\beta}_{T\cup C}(T, D)$ and the recorded attribute index value $r = j$,
               Else if $\Delta\gamma^{\beta}_{T\cup C}(T, D)$ is less than the previously recorded value $pv$, let
               $pv := \Delta\gamma^{\beta}_{T\cup C}(T, D)$ and let $r = j$;
   4.2. End For;
   4.3. Let $P := P \cup \{a_r\}$;
   4.4. Let $A := A - \{a_r\}$;
```

```
    4.5. If $pv = 0.0$ break while loop.          //no further augmentation necessary
5. End While;
6. Output $P$;
```

To demonstrate the RST_FS algorithm, an example is described with the aid of the graphical visualization shown below in Figure 4.2.3.2.1. This example was first presented in Beynon (2004). It is based on the wine data set (http://archive.ics.uci.edu/ml/datasets/Wine), which categorizes bottles of wine to three wine cultivators. The data set contains 178 objects described by 13 condition attributes $\{c_1,\ldots,c_{13}\}$ and classified by a three class decision attribute values (three cultivators). The condition attributes, were appropriately discretised using a dichotomous discretisation (discretisation method not given in Beynon, 2004). Table 4.2.3.2.1 summarised the results of the augmentation procedure, based on the wine data set.

nth Iteration	Attribute Augmented	P	$\Delta\gamma^{\beta}_{P\cup C}(P, D)$
1	c_1	$\{c_1\}$	0.6142
2	c_{12}	$\{c_1, c_{12}\}$	0.2966
3	c_{11}	$\{c_1, c_{11}, c_{12}\}$	0.2107
4	c_3	$\{c_1, c_3, c_{11}, c_{12}\}$	0.1605
5	c_4	$\{c_1, c_3, c_4, c_{11}, c_{12}\}$	0.1164
6	c_{10}	$\{c_1, c_3, c_4, c_{10}, c_{11}, c_{12}\}$	0.0861
7	c_2	$\{c_1, c_2, c_3, c_4, c_{10}, c_{11}, c_{12}\}$	0.0674
8	c_6	$\{c_1, c_2, c_3, c_4, c_6, c_{10}, c_{11}, c_{12}\}$	0.0449
9	c_{13}	$\{c_1, c_2, c_3, c_4, c_6, c_{10}, c_{11}, c_{12}, c_{13}\}$	0.0337
10	c_8	$\{c_1, c_2, c_3, c_4, c_6, c_8, c_{10}, c_{11}, c_{12}, c_{13}\}$	0.0225
11	c_5	$\{c_1, c_2, c_3, c_4, c_5, c_6, c_8, c_{10}, c_{11}, c_{12}, c_{13}\}$	0.0000

Table 4.2.3.2.1: Summary of the RST_FS Augmentation Process, on Successive Sets of P

Within Table 4.2.3.2.1, it shows for the first iteration, that the attribute c_1, where $\Delta\gamma^{\beta}_{P\cup C}(\{c_1\}, D) = 0.6142$, is the closest single attribute to the full set of attributes C (no information was given on the nearness of the other individual attributes). The remaining attributes were successively augmented to the set of previously selected attributes, based on which attribute resulted in the greatest reduction in distance through augmentation. With their concomitant distance measures $\Delta \gamma^{\beta}_{P \cup C}(P, D)$ shown in the fourth column of Table 4.2.3.2.1. Note that, for the eleventh iteration (which selected attribute c_5), the distance measure reaches zero, hence the process is terminated as no further selection of attributes is necessary. That is, a subset of eleven condition attributes have been identified, that are an exactitude, to the full set of the thirteen condition attributes C, in terms of β-interval domains and QoC. Figure 4.2.3.2.1 presents a visual representation of the augmentation process (taken from Beynon, 2004).

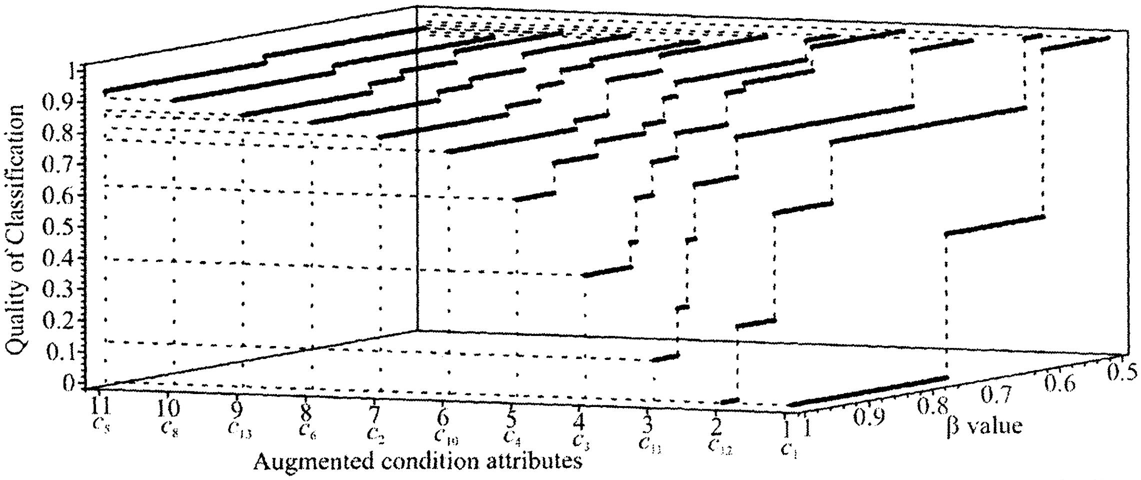


Figure 4.2.3.2.1: Graphical Representation of the Augmentation Process (Beynon, 2004)

The three dimensional graph in Figure 4.2.3.2.1, illustrates how the augmentation process converges on the β-intervals, and QoCs, associated with the full set of condition attributes C. A line for the full set of condition attributes C cannot be visualised on this particular graph, as it would occupy the same space as the subset P of the eleven identified attributes (to the far left of the graph). With regards to the developed software, a two dimensional version of the graph shown above was developed, that uses colour coding to represent the condition attributes augmented at each stage of the process (shown later in Chapter 6 section 6.3.2).

Finally, as a by-product of RST_FS, a more rudimentary feature selection method based on the individual distance measures $\Delta \gamma^{\beta}_{P \cup C}(P, D)$ for each condition attribute was developed. It was recognised that a ranking could be assigned to each attribute based on their individual distance measure. The first iteration of RST_FS calculates these individual distance measures, hence they were recorded along with the associated attribute, sorted, and a ranking assigned. The closest attribute (in terms of distance) is given the highest ranking. This simple feature selection method was named RST phase one or RST_PH1 for brevity, as it is based on the first phase of the RST_FS algorithm (finding the single nearest condition attributes).

The initial motivation for RST_PH1 was based on the realisation that RST_FS does not necessarily offer a complete ranking of all the condition attributes. However, it was promptly realised that the rankings given by RST_PH1 should be considered as a separate assessment to RST_FS, as RST_PH1 does not recognise dependencies between attributes, whereas RST_FS does, hence, the rankings given by RST_PH1 have no bearing on the rankings given by RST_FS. RST_PH1 supplies the analyst with a separate, alternative feature selection analysis, and should be regarded as such. Interestingly, based on correlation assessment, RST_PH1 produces comparable results to ReliefF (ReliefFC and ReleifFD), this will be shown later in Chapter 6 section 6.3.3. Additionally, because RST_PH1 does not recognise dependencies between attributes, it will not recognise redundant attributes, but it does recognise irrelevant attributes and assigns them lower rankings (comparable to ReliefF).

4.3 Imbalanced Data

Decision classes within data sets often have an uneven distribution (Maloof, 2003). That is to say, there may be more or less objects of a particular decision class when compared to another. An imbalanced data set occurs where there is a significant disparity between the number of objects belonging to the different decision classes (Guo and Viktor, 2004; Estabrooks et al., 2004; Grzymala-Busse et al., 2005). In extreme circumstances, there may be a small number of decision classes that dominate the data set (majority classes), leaving other decision classes critically under represented (minority classes) for the purpose of classification (An et al., 2001).

Imbalanced data sets can be detrimental to classifier performance. Training and testing classifiers on relatively balanced data sets does not reveal their potential vulnerabilities to the affects of imbalanced data (An et al., 2001). Japkowicz (2000) suggests that the underlying problem with many of the classifiers proposed in the extant literature, is that they distinguish a classifier's performance based solely on a measure of predictive accuracy taken across all decision classes, Japkowicz states (pp. 18):

> "Such a situation poses challenges for typical classifiers such as decision tree induction systems or multilayer perceptrons that are designed to optimize overall accuracy without taking into account the relative distribution of each class. As a result, these classifiers tend to ignore small classes while concentrating on classifying the large ones accurately. Unfortunately, this problem is quite pervasive as many domains are cursed with class imbalance."

In addition, Provost (2000) stated, that not only is there an assumption within many classifier designs that optimising overall predictive accuracy is the goal, but that the constructed classifier will also operate on data drawn from the same distribution as the training set.

There are two approaches to handling imbalanced data (An et al., 2001). Much in the same way as discretisation and feature selection methods can be either integral (dynamic, wrapper) to a classifier or an external (non-dynamic, filter) pre-processing step, balancing methods can also be divided by that distinction. Here, as before, we will be concentrating on external pre-processing approaches to data balancing.

The most common strategy, is to rebalance the data set artificially by taking samples from the objects belonging to each decision class. The three methodologies most frequently cited (Japkowicz, 2000), and hence implemented here, within the pre-processing software, are described below:

Under-sampling

Under-sampling, samples from the decision classes, where the sample size is equal to the number of objects within the minority class (Kubat and Matwin, 1997). The sample process is based on random sampling without replacement, although more sophisticated approaches do exist (Liu, 2004).

Under-sampling has the advantage that, by using less data, classifier construction time and computer resources are greatly reduced. However, by removing data, it is possible that valuable information may be lost that could be useful to the constructed classifier (*ibid*). Here, not to confuse with the issues of sub-sampling, under-sampling is referred to as down-balancing.

Over-sampling

Over-sampling increases the number of objects within the minority classes, up to the number of objects within the majority class, by randomly sampling with replacement each of the minority classes (i.e. the classes that are not the majority class) (Ling and Li, 1998).

Over-sampling has the advantage that it does not lose any data. Although, the increase in data can cause an increase in processing time and computer memory usage, Liu (2004) suggests a number of alternative over-sampling methods that may address these issues. Here, over-sampling is referred to as up-balancing.

Combined Over and Under-sampling

Chawla et al. (2002) implemented a method that over-sampled the minority class and under-sampled the majority class. Eastbrooks et al. (2004), also indicated that based on their results, it may be useful to combine the ideas of over-sampling and under-sampling by selecting a sample rate in-between the minority and majority class sizes, they state (pp. 27):

> "...the combination of the oversampling and undersampling strategies may be useful given the fact that the two approaches are both useful in the presence of imbalanced data sets and appear to learn concepts in different ways..."

Here, a simple combined approach is implemented that uses the average decision class size, and randomly over-samples the minority classes up, whilst under-sampling the majority class down. Here, combined sampling is referred to as average-balancing.

4.4 Missing Values

Real world data sets are often incomplete or 'contain' missing data. That is, attribute values may not have been recorded for every attribute associated with a subset of the available objects. These missing data values can occur for a number of reasons (Kantardzic, 2003). Issues surrounding missing data have been documented in: Friedman (1977), Breiman et al. (1984), Quinlan (1989), and Han and Kamber (2006).

Some machine learning processes can cope with missing data, whereas others require that data sets to be complete (Kantardzic, 2003). The simplest solution to the problem of incomplete data is to remove all objects containing missing data, but with some incomplete data sets, this approach may be too drastic, leaving the analyst with little or no data to train their classifier. A second solution would be to input values for the missing values during the pre-processing stage, and there are established approaches for doing this, namely: impute a global constant, impute the attribute mean, and impute the attribute mean value for the associated object's given decision class.

The three methodologies mentioned above, are widely used. With regards to the last two, imputing a calculated mean value has the potential to bias an attribute towards a certain value (Kantardzic, 2003; Han and Kamber, 2006). Widely accepted, robust approaches, to dealing with missing data have yet to be recognised, Weiss and Indurkhya (1998, pp. 61) suggest that the current approaches to dealing with missing data are, "...weak...", and that the problem of imputing a surrogate value is a whole prediction sub problem of its own, but logic based methods may be more robust to missing data (note though, they were quoting with regards to predictive, not classification methods[13]).

Although these methods for handling missing data are not ideal, it is clear that dealing with missing data is a matter of facilitating a balance between leaving out potentially useful data, and biasing a particular attribute towards a certain value. With regards to the developed software, here, a

[13] The difference between predictive and classification problems was stated in Chapter 1 section 1.1

compromise is implemented. Such that, only objects from the data set that are associated with less than 10% missing data will be used[14]. Where, missing values will be imputed using a method similar to imputing the decision class mean, except here, we will be calculating the average value based on the *k*-nearest neighbours in the decision class (*k*-nearest objects, based on Manhattan distance). As a default *k* was set equal to 10 (for similar methods see Jönsson and Wohlin, 2006). By using this combination of approaches it is hoped to lessen the impact of attribute value bias.

4.5 Data Transition Through the Data Pre-processing and Feature Selection Stages

Considering the scope of the pre-processing and feature selection methods that may be applied to the selected data set, before it is passed onto the data mining stage of the KDD process, it is pertinent to consider the correct order to which these methods are applied. Thus, to finalise this chapter, a flow chart is presented in Figure 4.5.1, which illustrates the order of events, charting the transition of the original data set from its raw unprocessed state, through its separation into training and validation sets, then the application of the pre-processing and feature selection methods before being passed into the data mining stage.

As can be seen from Figure 4.5.1, initially, the process of missing value imputation is performed on the original data set (see section 4.4), after which the data set is separated into a training set and a validation set (if a validation set is required by the analyst).

[14] The values of 10% and 5% are commonly used within statistics. A value of 5% was given consideration, but represented an appreciable loss of data, for further reference see later in section 5.5.

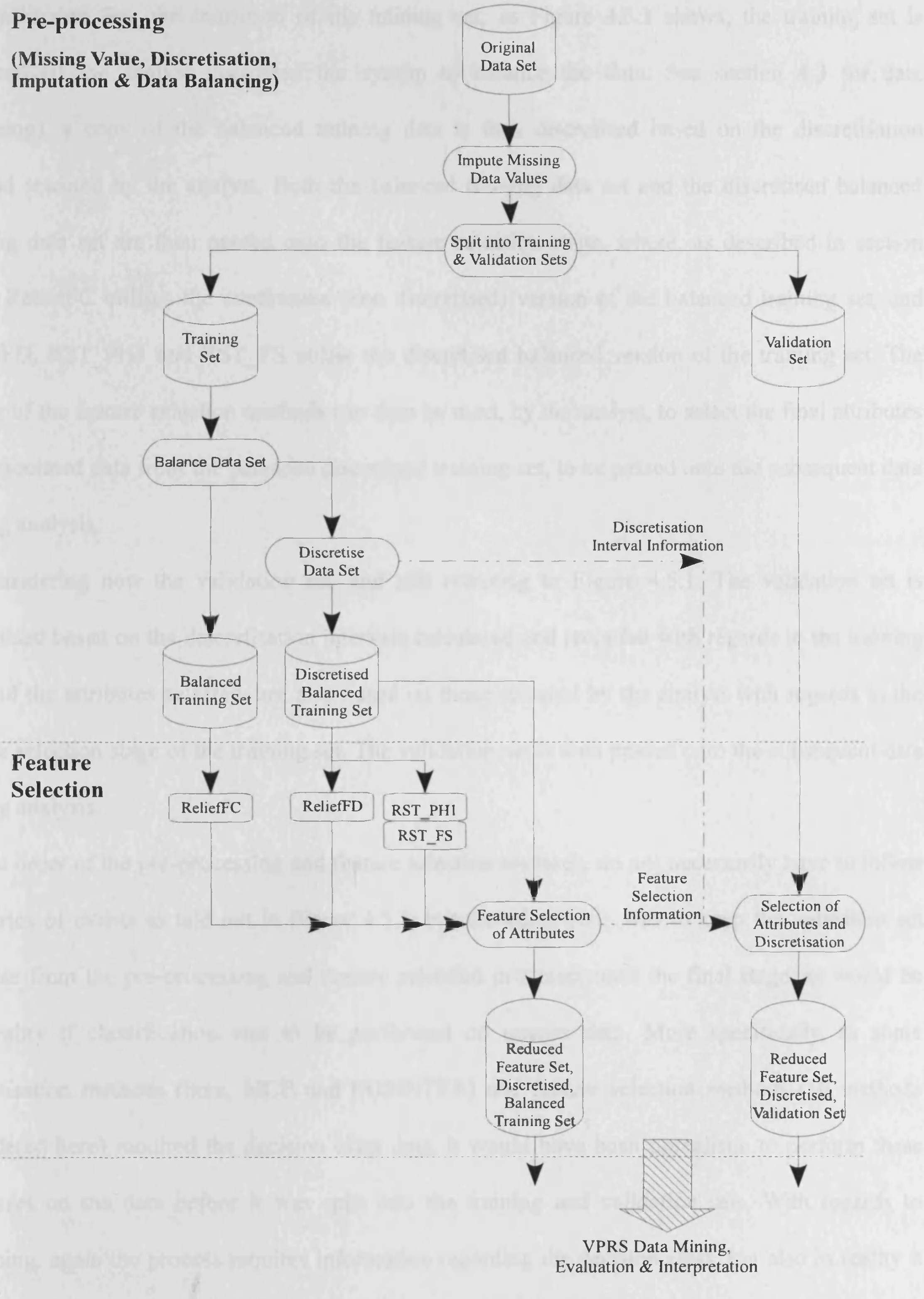

Pre-processing
(Missing Value, Discretisation, Imputation & Data Balancing)
Original Data Set
Impute Missing Data Values
Split into Training & Validation Sets
Training Set
Validation Set
Balance Data Set
Discretise Data Set
Discretisation Interval Information
Balanced Training Set
Discretised Balanced Training Set
Feature Selection
ReliefFC
ReliefFD
RST_PH1
RST_FS
Feature Selection of Attributes
Feature Selection Information
Selection of Attributes and Discretisation
Reduced Feature Set, Discretised, Balanced Training Set
Reduced Feature Set, Discretised, Validation Set
VPRS Data Mining, Evaluation & Interpretation

Figure 4.5.1: Transition of the Original Data Set through the Pre-processing and Feature Selection Stages

Considering first the transition of the training set, as Figure 4.5.1 shows, the training set is balanced (if the analyst instructed the system to balance the data. See section 4.3 for data balancing), a copy of the balanced training data is then discretised based on the discretisation method selected by the analyst. Both the balanced training data set and the discretised balanced training data set are then passed onto the feature selection stage, where, as described in section 4.2.3, ReliefFC utilises the continuous (non discretised) version of the balanced training set, and ReliefFD, RST_PH1 and RST_FS utilise the discretised balanced version of the training set. The results of the feature selection methods can then be used, by the analyst, to select the final attributes and associated data from the balanced discretised training set, to be passed onto the subsequent data mining analysis.

Considering now the validation set, and still referring to Figure 4.5.1. The validation set is discretised based on the discretisation intervals calculated and recorded with regards to the training set, and the attributes selected, are also based on those selected by the analyst with regards to the feature selection stage of the training set. The validation set is then passed onto the subsequent data mining analysis.

The order of the pre-processing and feature selection methods, do not necessarily have to follow the series of events as laid out in Figure 4.5.1, but the ethos here, was to keep the validation set separate from the pre-processing and feature selection processes until the final stage, as would be the reality if classification was to be performed on unseen data. More specifically, as some discretisation methods (here, MCE and FUSINTER) and feature selection methods (all methods considered here) required the decision class data, it would have been unrealistic to perform these processes on the data before it was split into the training and validation sets. With regards to balancing, again the process requires information regarding the decision class, but also in reality it would serve no purpose to balance unseen data. Finally, with regards to missing value imputation, here this was applied before the data set was split, but this was based on a practical consideration, as the theory relating to RST and VPRS described in Chapter 2 does not consider the problem of handling missing data with regards to classifying unseen objects. This problem is referred back to in Chapter 9, as work for future consideration.

4.6 Summary

This chapter has presented the three aspects of pre-processing most pertinent to the developed software, namely, discretisation, missing values, and imbalanced data. It has also presented a number of feature selection methods. Although, pre-processing and feature selection have not been presented as extensively, as perhaps is warranted, we have been constrained both by development time and limits to the size of this dissertation.

Four methods of discretisation were chosen for implementation, namely equal-width, equal-frequency, Minimum Class Entropy (MCE) and FUSINTER. These methods were discussed in detail, but the wider issues surrounding discretisation, including other established methodologies were also highlighted.

Technically, two methods for "pre-feature" selection were chosen for implementation, namely ReliefF and a method based on RST, designated RST_FS (though four sets of results are derived from the two methods). ReliefF will be applied to the data, both pre and post discretisation. To distinguish the two variations they have been designated ReliefFC and ReliefFD. As a first phase in the RST_FS algorithm, RST_PH1 was also identified as a rudimentary feature selection algorithm based on measures of QoC. Therefore, with regards to what attributes to pass into the subsequent VPRS analysis, from the two feature selection methods implemented, four sets of results will be available to the analyst, to aid their decision making.

With regards to balancing, the issue appears to have received proportionally less attention from the data-mining/machine learning community. Here, we will be implementing the three basic re-balancing methods, down-balancing, up-balancing, and average-balancing.

The issue of missing data was highlighted. Again, some basic methodologies for handling missing data have been presented. Here, a strategy that is a compromise between excluding objects with missing attribute values, and imputing the missing values using a combined decision class and *k*-nearest neighbour approach, has been implemented within the pre-processing software.

Finally, it is worth mentioning here, that the software was designed in such a way, as to allow a "retro fit" of other pre-processing methods, at a later date (given more development time).

Chapter 5

Credit Ratings, Fitch Individual Bank Strength Ratings and Initial Data Selection

> "There are two superpowers in the world today in my opinion. There's the united States and there's Moody's Bond Rating Service. The United States can destroy you by dropping bombs, and Moody's can destroy you by downgrading your bonds. And believe me, it's not clear sometimes who's the more powerful."
>
> (Friedman, 1996)

This chapter introduces the credit rating problem, and in particular, the prediction of bank credit ratings, as the area of application, for which the developed VPRS software will be applied. It offers a justification of why an analyst may be interested in predicting a credit rating. As the opening quote suggests, credit rating agencies such as Moody's, Fitch and S&P (Standard and Poor), have within recent years, become increasingly important within the fixed income financial markets (Sylla, 2002; Partnoy, 2006). It is hypothesised by many scholars that this increase, is by and large, a result of tighter financial regulations imposed by domicile authorities such as the United States Securities and Exchange Commmision (SEC) (Levich et al., 2002; Partnoy, 2002; SEC, 2003) or, the international implementation of the Basel Committee's, Basel one (1988) and Basel two (2007) accords on banking laws, recommendations, regulations and codes of best practice (Basel, 2007).

Currently, there are only a handful of globally recognised rating agencies, predominantly those mentioned above, that appear to dominate the market (Cantor and Packer, 1994, 1995; Partnoy, 2002). Although, there are a considerable number of less recognised global firms and agencies that operate within smaller economic regions (typically within the country in which they are registered). For a comprehensive overview of global and regional rating agencies, see Smith and Walter (2002).

Many financial institutions allocate considerable resources to the measurement and management of credit risk, that is, the assessment of the possibility, of debt issuers defaulting. Credit rating agencies are in the business of providing credit ratings, which indicate their assessment of the issuers credit worthiness (Hull, 2006). Originally, the subscribers (the lenders) would pay the rating agencies a fee for the information, but in recent times the issuers have had to pay a fee to be rated, which has raised issues over conflicts of interest within the ratings industry (Smith and Walter, 2002; Poon and Firth, 2005; Van Roy, 2006).

A credit rating typically takes the form of a discrete alpha numeric value. Taking Moody's corporate bond rating system as an example, the highest Aaa rating is considered as having almost no chance of defaulting, with the next safest level being Aa and the following ratings are, A, Baa, Ba, B, Caa, with any rating over Baa considered investment grade (Hull, 2006). To further refine such a system, numerical values may be introduced (e.g. Moody's A1, A2, A3, Aa1 etc.) or positive and negative signs may be appended to indicate higher or lower risk (e.g. S&P's AA+, AA, AA–, A+ etc.), there are a number of other appendages used by other rating agencies.

There are, for a variety of reasons, a number of different organisations who are interested in the prediction of credit ratings. For example, governmental organisations such as the Federal Reserve in the U.S. or the U.K.'s Bank of England, are interested in implementing off-site early warning systems to identify struggling banks (increased likelihood of banking collapse) (Sahajwala and Van den Bergh, 2000). Off-site monitoring systems imply that there has been no direct consultation with the bank in question, identification of possible struggling banks may involve more costly on site inspections (Jagtiani et al., 2003)

In terms of the wider credit rating industry, investors are also interested in predicting ratings, as Peavy (1984, pp. 46) noted:

> "The ability to successfully predict industrial bond rating changes would be most useful in formulating profitable bond portfolio strategies...if bond yields (and therefore prices) do indeed adjust immediately after, but not before, a rating change announcement, then the ability to predict these reclassifications should translate into a profitable trading strategy."

Another direction relating to credit rating prediction, is with regards to predicting the credit risk of companies, that have not been issued a rating. Companies may wish not to be rated for a number of reasons, Huang et al. (2004b) suggest that the time and human resources needed by the credit rating companies, to do a thorough on-site inspection, prove too expensive for some smaller companies. Poon (2003) state a number of other reasons why a company may not wish to be rated:

1. Middle eastern bankers prefer not to raise debt or equity from oversees investors as they do not wish foreign share holders to take control of their countries banks, and as net lenders on the global inter-banking system they do not need to obtain a rating (Harington, 1997).
2. China prefers local rating agencies to rate their financial institutions. As they claim that their regulatory procedures are different, and external companies, such as the U.S. based Moody's, are not knowledgeable enough about the current situation in China to make a well informed decision.
3. Issuers may fear that their rating would not be up to investment grade quality, so do not solicit a rating in case an unfavourable rating would damage their company's reputation.

Interestingly, in reference to the second point on China, Poon and Chan (2008) highlight that China's rating agencies exaggerate the credit worthiness of the companies that they rate, and indeed a disproportionate number of companies are rated as investment grade or above, whilst external global ratings agencies which assign them unsolicited ratings[15] (an off-site assessment based on publicly available data, typically balance sheet data), find that the companies are in fact only speculative grade. Kennedy (2003) suggests that, investors within the Chinese market attach little creditability to these domestic ratings. So there is clearly an opportunity for off-site models to provide impartial risk assessments to investors.

Poon's third point, in some respects relates to the rating transition process, which if understood, could provide some incite into the rating process used by the rating agencies (Kim and Sohn, 2008). Which in turn, would enable companies to improve specific areas of operation, to encourage a more favourable credit rating (with the intention of attaining and retaining an investment grade rating). It should be noted, that larger credit rating agencies, such as Moody's, do not explicitly disclose their rating rationale and claim that quantitative models cannot capture the qualitative aspects included in their model (Shin and Han, 2001). Conversely, smaller agencies which are less dependant on subscriber revenue are more transparent and make their ratings and rating rationale publicly available (SEC, 2003).

The remainder of this chapter describes the attribute model, and the data selected within this dissertation, for the prediction of bank ratings. The sections are outlined below:

- Section 5.1 **Historic Overview**. This section provides a concise historic overview, of credit ratings, and the credit ratings industry.
- Section 5.2 **Bank Ratings and Their Prediction**. This section describes the problem of bank credit rating prediction, and investigates a number of studies related to that topic.
- Section 5.3 **CAMELS Model**. This section describes the CAMELS model, a generally accepted model within the extant literature relating to bank credit rating prediction.
- Section 5.4 **Fitch Individual Bank Ratings**. This section describes Fitch's Individual Bank Strength Ratings, and their prediction. Which is the focus application, of the developed software

[15] Also an area of some contention within the ratings industry (Smith and Walter, 2002; Poon and Firth, 2005; Van Roy, 2006).

within this dissertation.

- Section 5.5 **Data Selection and Attribute Model.** Based on the CAMELS model, this section details the final attribute and data selection process.
- Section 5.6 **Summary**. This section briefly summaries the main points of this chapter, relevant to the subsequent results chapters.

5.1 Historic Overview

The origin of the credit rating agency is generally attributed to John Moody, who in 1909 set up the first recognised rating agency, to rate the bonded debts of the burgeoning U.S. railroad market (Moody's, 2007). Although, the practice of bond ratings can in some respects be traced to the earlier 1857 company founded by John Bradstreet, who published what appeared to be the first commercially available ratings book. The Bradstreet company, later went on to merge with the American credit-reporting agency R. G. Dun and Co. in 1933, to form Dun & Bradstreet, who in 1962 went on to acquire Moody's Investor services (The bond rating agency originally set up by John Moody in 1909), but they continued to operate as independent entities (Cantor and Packer, 1994, 1995).

Since their initial introduction by Moody's, the importance of credit ratings has fluctuated, in the eyes of both the issuers and the lenders. Indeed, the bonded markets of America had existed for about 300 years without the necessity of credit ratings (Levich et al., 2002; Sylla, 2002). Moreover, the railroad markets had been issuing corporate bonds for at least 60 years before John Moody's entrepreneurial inception of the credit rating. For Moody, the pivotal point came with, the critical mass of the expanding investment class within the American society, hungry for financial information; and railroad companies endeavouring to raise capital on an almost continental scale. It is also, in part, attributed to the decline of the type of investment bankers operating during that period, because of the combined disdain for them, both by issuers and lenders (Sylla, 2002). The relevance of the credit rating was boosted in the period between the 1914 and 1940, due to America emerging out of World War 1 as the worlds new financial superpower, but more so because of the role rating agencies were beginning to play within the tighter regulations introduced by the U.S. at Sate and Federal levels (White, 2002).

The period between 1940 and 1970 saw a stagnation of the rating agencies importance (Partnoy, 2002). Many issuers considering credit ratings, perhaps more of a necessity than of any practical real value (Partnoy, 1999), and the lenders saw that ratings carried little more information than could actually be inferred from the market place (Pinches and Mingo, 1973; Reilly and Joehnk, 1976; Pinches and Singleton, 1978). The information value of credit ratings is a topic of some contention within the extant literature, with arguments both for and against the perceived value of the information provided by a rating (for an overview see Gonzalez et al., 2004).

From 1970 onwards, history in some respects repeated itself, almost in an ironic sense. As credit ratings originally emerged from the need to rate prosperous railroad companies in 1909, by 1970 it was the default and inevitable bankruptcy of Pennsylvania and New York Central Transportation, which ignited a chain reaction amongst the other railroad companies, that caused investors to demand better research into companies' credit worthiness (Partnoy, 2002; Daughen and Binzen, 1999). Additionally, and probably a factor that caused the financial problems within the railroad companies, was the global credit crises of the 1970s which led the governments of the time to impose further regulation on the financial sector and to introduce stricter fiscal policies (Partnoy, 2002). Notable, was the implementation of a regulation imposed by the U.S. SEC in 1975 which incorporated the use of credit ratings into the regulations, but only by Nationally Recognised Statistical Rating Organisations (NRSROs).

The use of NRSROs gave recognition to the then established rating agencies Moody's, S&P and Fitch, and affectively froze out any other competition, because of a paradox, that is, being able to raise what is termed reputation capital before being given the accolade of NRSRO (Cantor and Packer, 1995). To explain further, a rating agency would need to prove its competence and gain reputation before being accepted as an NRSRO; which would prove difficult, because by not initially being an NSRRO, a new ratings company would find it difficult to enter the market and raise the reputation capital needed. It is to this, that some studies attribute the reason behind, why there are so few globally recognised ratings agencies (White, 2002).

As a consequence of the tighter regulations imposed throughout the period of the 1970s, the 1980s saw a phenomenal rise in the importance of rating agencies (including their staffing levels and profits), again boosted by the 1988 Basel accord on banking regulation (Partnoy, 2002; Sylla, 2002) By the late 1990s, the requirement for a credit rating had almost become a expected component of many large reputable companies' profiles, and an antecedent part of a new company seeking credit approval (Partnoy, 1999). With the 2007 implementation of the much awaited Basel two accord, it is likely that the future position of the credit rating agencies is only going to become stronger. Furthermore, with the recent credit crises of the 2007 and 2008, where the rating agencies failed to anticipate the initial downgrading of Collateralised Debt Obligations (CDOs) caused by problems within the U.S. sub prime mortgage market (Stempel, 2007), investors will inevitably echo the calls of the 1970s, for better research into the financial worthiness of the institutions and even stricter financial regulations, indeed a quick search for 'Basel III' on the internet delivers a plethora of speculative articles.

5.2 Bank Ratings and their Prediction

It is evident from the historic overview of credit ratings that the American financial market has been a leading force within the credit rating industry. Interestingly, the U.K. which accounts for 20% of the worlds cross boarder lending and is the biggest financial centre in the E.U. (Kosmidou et al., 2006), as of 2002, had no rating agency solely headquartered in that country (White, 2002). However, the trend towards globalisation has seen the credit rating firms rise to prominence and providing services on a global scale, Poon and Firth (2005, pp. 1742) noted:

> "While once confined to the U.S. and other industrialised nations, credit rating agencies now evaluate firms in virtually all countries with organised securities markets and even in some that don't."

International financial regulations such as the Basel accords and the relaxation of individual countries' rules on cross country banking activity, have, no doubt been the catalyst towards this globalisation of the banking industry (Kosmidou et al., 2006).

Banking, particularly banking external to the U.S., is one area of the financial industry that has felt the impact of this globalisation, and has had to implement the American practice of ratings to aid in the regulation of the banking sector. White (2002, pp. 58) summed up the differences between the manner in which U.S. banks supply capital and how banking systems external to the U.S. encourage capital raising:

> "These countries have tended to stress bank-supplied loans as their sources of finance for companies; and, since the countries tend to be more geographically compact than is the United States and they encourage nationwide branching, the banks themselves could be effective information gatherers."

However, given the effects of globalisation, banks now need to be competitive on a global scale, which has led to mergers and a concentration of the banks into larger institutions operating across global markets (Kosmidou et al., 2006). As a response, in 1995, Moody's ushered in a new type of rating, namely the Bank Financial Strength Rating (BFSR), with other ratings companies following suit and offering similar rating systems, such as Fitch's Individual Bank Strength Rating (FIBR) and Bank support ratings.

The U.S. banking regulators, have had, for many years, early warning models in place to identify struggling banks, such as the CAEL system implemented in the early 1980s, used by the Federal Deposit Insurance Company (FDIC) and the Federal Reserve. However, the general field of rating prediction (particularly bond ratings), was pioneered earlier than the introduction of the CAEL model, with perhaps some of earliest work having been undertaken by Horrigan (1966). Horrigan used a multiple regression model with very limited success, predicting 58% of Moody's bond ratings and 52% of S&P's. West's (1970) model, achieved slightly more success, using the attributes proposed by Fisher's (1959) work on determining risk premiums on corporate bonds; West achieved a 62% predictive accuracy on Moody's ratings. Motivated by the observation of Foster (1978) who noted that previous models lacked an 'economic rationale', Belkaoui (1980) implemented a model identifying eight financial attributes, which they believed captured the essence of three factors, which they felt, determined the investment quality of a bond (i.e. their economic rationale). Although the predictive model still only achieved a 65.9% predictive accuracy on S&P's ratings, on the breakdown of the different bond rating grades, they found a range of predictive accuracies, with a then impressive 75% of B ratings being predicted correctly (but only 36% accuracy for the BBB rating). Though not conclusive proof of their method, it did set a precedent in that the models that followed attempted to apply a rationale to their attribute selection. The internal[16] BOPEC and CAMELS bank ratings systems in use by the U.S. Federal Reserve are indicative of the principle of implementing an economic rationale (the CAEL and CAMELS models are described in more depth within the next section).

Over the following thirty years, the statistical techniques and the quality of the data improved (Poon et al., 1999), leading to better rates of predictive accuracy (for a comparison, see overviews in, Altman et al., 1981; Huang et al., 2004b). Returning to the prediction of bank ratings specifically, there appears to be, only a limited number of studies concentrating on predicting ratings issued by commercial rating agencies (since as previously stated, bank ratings were only introduced in the late 1990s). There have however, been numerous studies looking at the prediction of the U.S. banks' internal BOPEC and CAMELS ratings (for examples, see Gilbert et al., 2000; Krainer and Lopez, 2003).

Poon and Firth (1999) appear to have conducted the first study investigating the prediction of

[16] Internal, in the sense that, the ratings are not disclosed to the general public.

commercially available bank ratings. They utilised six various logistic regression models based on a selection of different attributes to predict ten categories of the Moody's BFSR's (A+, A, B+, B, C+, C, D+, D, E+ and E). They achieved accuracies of between 21.1% for the poorest model and 71.1% for the best model. Although, it does appear these accuracies are based on the training sample (i.e. the apparent predictive accuracy as described in Chapter 4 section 4.1, which is typically over optimistic). More recently, Poon and Firth (2005) investigated whether unsolicited credit ratings (also called shadow ratings) were lower than solicited ratings. They also investigated the overall ability of their model to predict Fitch's FIBRs, with some impressive results. On a validation sample, they achieved a predictive accuracy of 85%. Other notable studies include, Kosmidou et al. (2006), who analysed the financial characteristics (attributes) important to foreign and domestic banks operating within the U.K.; and Pasiouras et al. (2006), who investigated the importance of banking regulation, supervision and market structure with regards to characterising bank ratings. They found that different combinations and enforcement of these factors (such as amount of regulation, deposit insurance schemes etc.), did have some impact on the prediction of FIBRs.

The aforementioned CAMELS model has become core to many of the studies into rating prediction, particularly within bank rating prediction, such as, those studies mentioned above. As previously stated, the CAEL model (a derivative of the CAMELS model), was the model implemented by the FDIC and the Federal Reserve as an early warning system to identify failing banks. Here, we will be utilising the CAMELS economic rationale which is described in the following section.

5.3 CAMELS Model

During the early 1980s, the U.S. supervisory authorities such as the Federal Reserve and FDIC, to assess the likelihood of a bank failing, introduced the CAMEL model as an on-site assessment system. The system assigns a grade between, one (best) and five (worst), reflecting the supervisors assessment of the banks condition. CAMEL is an acronym for the five elements deemed to be component factors for the successful operation of a banking institution, namely Capital, Asset Quality, Management, Earnings and Liquidity. A sixth 'S' component was added to the model in 1997, in an effort to capture the banks Sensitivity to market risk, hence the modern acronym is referred to as the CAMELS model (Feldman et al., 2003; Derviz and Podpiera, 2004).

Prior to the introduction of the CAMEL model, and since the early 1970s, the U.S. authorities had used off-site computer systems to identify possible problem banks, but these initial off-site monitoring systems performed poorly (Sahajwala and Van den Bergh, 2000). Subsequently, and in parallel with the introduction of the CAMEL rating, the FDIC implemented the CAEL model which took the four components of the original CAMEL model that could be used to assess a quarterly off-site assessment (i.e. Capital, Asset Quality, Earnings and Liquidity). This CAEL rating would be compared to the most recent CAMEL(S) rating, and if there was a change, the FDIC would then identify the bank for further investigation, and a possible full on-site inspection. Essentially, the CAEL system was seen as an early warning system, however, since 1999 it has been withdrawn and superseded by the SCOR system (Statistical **C**AMELS **O**ff-site **R**ating). The Federal Reserve have also implemented their own early warning system independently of the other regulatory authorities. Known as SEER (System for Estimating Exam Ratings), it involves two models, one for the predicting of bank failure over a two year horizon, and the second for predicting CAMELS ratings. More recently, European financial authorities have also introduced early warning systems, such as U.K.'s financial services authorities RATE system (Risk Assessment, Tools of Supervision and Evaluation) (see Sahajwala and Van den Bergh, 2000, for an overview of some of the most prominent early warning systems used globally).

Today, the CAMELS model is widely accepted as capturing the elements that underpin a financial institution's level of risk, and has become widely used as the economic rationale of choice in many studies (e.g. Pasiouras et al., 2006). For off-site systems, four of the six elements of the CAMELS model, namely 'C', 'A', 'E', 'L' are captured using balance sheet data, and often captured in the form of financial ratios (described in section 5.5). The management and sensitivity elements of the CAMELS model, are more difficult to capture, but a number of studies have used "proxy" attributes in an attempt to capture elements of management style and sensitivity to market risk (Krainer and Lopez, 2003; Pasiouras et al., 2006; Van Roy, 2006).

5.4 Fitch Individual Bank Ratings

As was stated in the introduction to this chapter, the developed VPRS software will be applied to the prediction of bank ratings, specifically, Fitch 's Individual Bank Strength ratings (FIBRs) (Fitch, 2007). There is only a limited amount of research literature specific to FIBR prediction. Both Poon and Firth (2005) and Van Roy (2006) investigated whether there was evidence that unsolicited FIBRs were lower than solicited ratings; and Pasiouras et al. (2006) tested whether attributes capturing the environment aspects in which the bank operated, such as banking regulations, would have an impact on FIBR prediction models. Fitch (2007) describe FIBRs as:

> "...ratings, which are internationally comparable, attempt to assess how a bank would be viewed if it were entirely independent and could not rely on external support. These ratings are designed to assess a bank's exposure to, appetite for, and management of risk, and thus represent our view on the likelihood that it would run into significant difficulties such that it would require support."

FIBRs are divided into six categories, representing Fitch's opinion on the likelihood that a bank will get into difficulties (or is in difficulty), and in such an event would require external support. External support can be in the form of state assistance (e.g. The Bank of England, Federal Reserve etc.), deposit insurance funds (e.g. the U.S.'s Federal Deposit Insurance Corporation FDIC); acquisition by some other corporate entity or an injection of new funds from its shareholders or equivalent.

The five main FIBR's categories are described below in Table 5.4.1, as given by Fitch (2007).

Fitch Individual Bank Rating	Recoded to	Description
A	0	A very strong bank. Characteristics may include outstanding profitability and balance sheet integrity, franchise, management, operating environment or prospects.
B	1	A strong bank. There are no major concerns regarding the bank. Characteristics may include strong profitability and balance sheet integrity, franchise, management, operating environment or prospects.
C	2	An adequate bank, which, however, possesses one or more troublesome aspects. There may be some concerns regarding its profitability and balance sheet integrity, franchise, management, operating environment or prospects.
D	3	A bank, which has weaknesses of internal and/or external origin. There are concerns regarding its profitability and balance sheet integrity, franchise, management, operating environment or prospects. Banks in emerging markets are necessarily faced with a greater number of potential deficiencies of external origin.
E	4	A bank with very serious problems, which either requires or is likely to require external support.

Table 5.4.1: Fitch's Definition of its Individual Bank Rating Categories

In addition, Fitch also provide an F category that represents a bank, that has either defaulted or, in Fitch's opinion, would have defaulted if it had not received external support. The F category is not considered within this dissertation as it is not available within our database (described in the next section). The five main 'A' to 'E' categories will be recoded to numerical values as shown on the second column of Table 5.4.1 (the developed VPRS software operates on numeric data values).

Furthermore, Fitch describes four intermediate categories known as graduations, they are A/B, B/ C, C/D, and D/E. Banks are assigned these graduation ratings if Fitch deems them to be in between two of the five main 'A' to 'E' ratings. Banks within these categories are left out at the data selection stage (described next), as initial studies suggested that the results were improved by increasing the notional boundary between rating categories. That is, where banks may have been border line between ratings, it caused the resultant VPRS classifier (rules) from the developed software to perform poorly on the validation sample, perhaps because of the weakness of these "border line rules". This issue of weak boundary area rule reduction is considered in Ziarko (2003), who suggests three approaches for the reduction of the boundary area. The concept of these border line objects (resulting in weak rules) and the impact they have within the training phase (rule creation) of VPRS is a topic which may need some attention in the future.

5.5 Data Selection and Attribute Model

The target data used within this dissertation, has been taken from Bureau van Dijk's Bankscope Database (2007). This database, provides information on banks and financial institutions world wide, with up to 16 years of detailed accounts, on ratios, ratings and rating reports, ownership, country risk and country finance reports.

The previously discussed CAMELS model has been utilised as a basis for attribute selection. Within the related literature, the elements of the CAMELS model are typically captured using balance sheet data in the form of financial ratios (Sahajwala and Van Den, 2000). Financial ratios are used to evaluate the overall financial condition of a company. They are expressed as decimal values and are used by company managers, shareholders and financial analysts. The practice of using financial ratios has been around since the late 1890s, for a historical analysis see Horrigan (1968).

Table 5.5.1 lists the number of occurrences of financial ratios that occurred twice or more, across twelve recent studies related to bank rating predictions (Poon et al., 1999; Gilbert et al., 2000;

Raveh, 2000; DeYoung et al., 2001; Feldman et al., 2003; Krainer and Lopez, 2003; Derviz and Podpiera, 2004; Poon, 2003; Poon and Firth, 2005; Kosmidou et al., 2006; Pasiouras et al., 2006; Van Roy, 2006). These ratios have been separated into the appropriate CAMELS categories for which they are considered representative. Ratio explanations can be found on Bureau van Dijk's Bankscope (2007) product support manual.

CAMELS Category	**Attribute Occurrence**	**CAMELS Category**	**Attribute Occurrence**
CAPITAL Adequacy(C)		**EARNINGS (E)**	
Tier 1 Ratio	2	Net Interest Margin	3
Total Capital Ratio	4	Net Int. Inc./ Aveg Assest	3
Equity / Total Assets	9	Non Int Exp / Avg Assets	3
Equity / Net Loans	2	Return on Average Assets	11
Cap Funds / Tot Assets	2	Return on Average Equity	5
		Cost to Income Ratio	4
ASSET QUALITY (A)			
Loan Loss Reserve / Gross Loans	2	**LIQUIDITY (L)**	
Loan Loss Prov / Net Int Rev	4	Net Loans / Total Assets	5
Loan Loss Res / Impaired Loans	2	Net Loans / Customer & ST Funding	3
Impaired Loans / Gross Loans	3	Liquid Assets / Cust & ST Funding	5
		Sensitivity to Market Risk (S)	
		Number of subsidiaries	3

Table 5.5.1: Occurrence of Attributes from 12 Recent Bank Rating Studies

The management element 'M' of the CAMELS model is absent from Table 5.5.1. Many studies ignore this category, due to its qualitative nature and the subjective analysis required, it is deemed difficult to capture and quantify (Pasiouras et al., 2006; Sahajwala and Van den Bergh, 2000). The sensitivity to market risk category 'S', which is a later addition to the CAMELS model, has received less attention than the other categories, mainly because the literature concentrates on the pre 'S' category period, where there is more data available (Gilbert et al., 2000; Feldman et al., 2003).

Derviz and Podpiera (2004), suggest using the attribute Total Assets Value at Risk (see McNeil et al., 2005) to model the 'S' category, as it is commonly used by financial institutions to measure the market risk of their portfolio (but it is not available within Bankscope). Pasiouras et al. (2006), presented a comprehensive study, suggesting a number of attributes that could be used as proxies for non-quantitative elements of bank ratings, particularly with relation to banking regulations, supervision and market structure. Interestingly, based on Falkenstein et al.'s (2000) hypothesis that smaller companies have less depth in management and are more susceptible to idiosyncratic shocks within the market, Pasiouras et al. (2006) suggest using the number of subsidiaries as a measure of business size and diversification. Supporting Pasiouras et al.'s argument, Fitch (2007) state that the banks diversification in terms of involvement in a variety of activities in different economic and geographical sectors, is, an important factor in their model.

Based on the attributes in Table 5.5.1, Table 5.5.2 presents the final selection of attributes taken form Bankscope, with the addition of some attributes that were available, and that were deemed appropriate to the model. Note that, Net Int. Inc./Aveg Assets (net interest income to average assets) as shown in Table 5.5.1, was not available within Bankscope, and hence does not appear in Table 5.5.2. The second column displays the amount of missing data associated with each attribute (discussed later in this section).

CAMELS Category	**Missing Data (%)**	**CAMELS Category**	**Missing Data (%)**
CAPITAL (C)		**EARNINGS (E)**	
Tier 1 Ratio	8.055	Net Interest Margin	0.000
Total Capital Ratio	0.323	Non Int Exp / Avg Assets	0.000
Equity / Total Assets	0.000	Return on Average Assets	0.000
Equity / Net Loans	0.323	Return on Average Equity	0.000
Cap Funds / Tot Assets	0.000	Cost to Income Ratio	0.161
Subord Debt / Cap Funds	1.936		
		LIQUIDITY (L)	
ASSET QUALITY (A)		Net Loans / Total Assets	0.000
Loan Loss Reserve / Gross Loans	0.000	Net Loans / Customer & ST Funding	0.161
Loan Loss Prov / Net Int Rev	0.323	Liquid Assets / Cust & ST Funding	0.806
Loan Loss Res / Impaired Loans	7.258		
Impaired Loans / Gross Loans	0.000	**Sensitivity to Market Risk (S)**	
		EIU Overall Country Risk	0.000
		EIU Banking Sector Risk	0.000
		EIU Banking Sector Risk Outlook	3.065
		Number of recorded subsidiaries	0.000
		GDP/head	0.000

Table 5.5.2: Final Attribute Selection

The ratio Subord Debt/Cap Funds (subordinated debt to capital funds) has been included under the Capital Adequacy category 'C', as a number of papers referred to the possible importance of subordinated debt in the overall risk of a institution's portfolio (e.g Krainer and Lopez, 2003; Derviz and Podpiera, 2004). Subordinated debt refers to debt that is, repayable only after a borrower's other debts or financial obligations have been settled, in the event of foreclosure subordinated debt has the lowest priority. Thus, it is deemed to be a more risky option for the lender. According to Bankscope, the lower the ratio of subordinated debt to capital funds the better, that is, less risk associated with the institution.

Within Table 5.5.2, perhaps the most liberty in terms of attribute selection has been taken with regards to the Sensitivity to Market Risk category 'S', in that, five additional attributes that were available in Bankscope have been included, because it was deemed that they may perform as good proxies for important factors within our model. To explain our rationale, Le Bras and Andrews (2004) suggest that Fitch consider a number of factors which relate to a banks operating environment including, a countries political situation and banking regulatory system. Additionally, Fitch (2007) state themselves that FIBRs are internationally comparable and that operating environment is important. Hence, the three EIU[17] attributes Overall Country Risk, Banking Sector Risk and Banking Sector Risk Outlook, have been included in our attribute selection. Poon and Firth (1999) appear to support the addition of banking environment attributes, as they include a similar country risk attribute CRISK which they obtained from the International Country Risk Guide (Sealy, 1997). They state that this attribute is a composite measure of three factors, namely, a countries political, financial and economic risk. In addition to the banking environment attributes, GDP/head (Gross Domestic Product per head) was included in our model, as it may reflect, a candid measure of the internal economic situation within a country.

Having identified the final set of attributes (Table 5.5.2), and selecting only banks (objects) from Bureau van Dijk's Bankscope Database (2007), associated with less than 5% missing data (See Chapter 4 section 4.4 for description of missing data), the target data set contained 620 banks. With regards to the percentage of missing data associated to each attribute, as shown in Table 5.5.2, only two attributes (Tier 1 Ratio and Loan Loss Res/Impaired Loans) have more than 5% missing data, but still less than 10%. Table 5.5.3 presents the distribution of the number of banks associated with the five FIBR rating levels .

[17] Economist Intelligence Unit (EIU, 2007) attributes were available directly through Bankscope.

Rating	**A**	**B**	**C**	**D**	**E**	**Total**
Grade	**A**	**B**	**C**	**D**	**E**	
Number of Banks	16	319	163	107	15	620
Percentage of Total	2.5811	51.452	26.290	17.258	2.419	100

Table 5.5.3: Distribution of FIBR Ratings

It can be seen from Table 5.5.3, that the data within the five rating classes is quite imbalanced, with the 'B' rated banks being the majority class (See Chapter 4 section 4.3 for description of imbalanced data). The 'A' grade and 'E' grade banks being the under represented classes.

5.6 Summary

This Chapter has presented an overview of credit ratings and the credit rating industry. The main emphasis has been on bank ratings, and the prediction of bank ratings. The CAMELS model has been described as a general rationale for attribute selection.

The last section of this chapter described the final attribute and data selection, concerned with those attributes that where assessed to be relevant to the prediction of Fitch's Individual Bank strength Ratings (FIBRs). The data has been obtained from Bureau van Dijk's Bankscope Database, and is used in the following three chapters.

Chapter 6

Introduction to Software, Pre-processing and Feature Selection Results

This chapter introduces and describes the developed software, in particular, the results of the pre-processing element and feature selection elements of the software, applied to the FIBR data set as discussed in Chapter 5. The results from this chapter are used in the subsequent analyses of the FIBR data, presented in Chapters 7 and 8 (which further describe the developed software).

This chapter contains three sections, as outlined below:

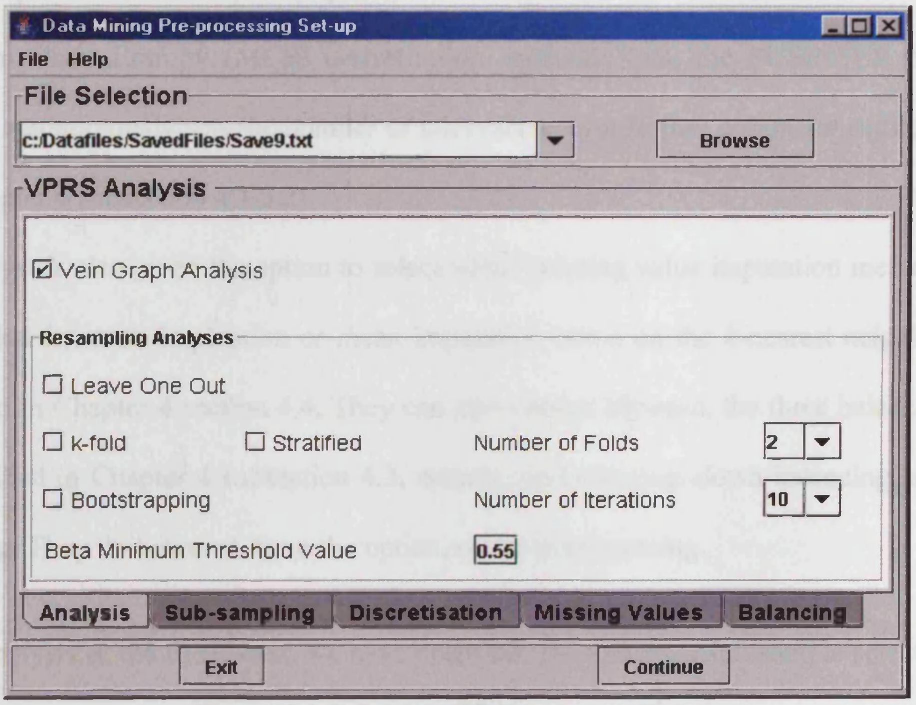
- Section 6.1. **Parameter Setting**. This section presents and describes, the opening window of the developed software, which allows the analyst to set parameters relating to the subsequent, pre-processing, feature selection, data mining and evaluation (re-sampling) stages of the KDD process, as outlined in Chapter 1.
- Section 6.2. **Data Discretisation Results**. This section describes the results of the data discretisation and briefly discusses the issues encountered with regards to data balancing and missing value imputation.
- Section 6.3. **Feature Selection**. This section describes the results of the feature selection algorithms employed, including a number of graphs that were developed to assist the analyst during the final attribute selection. Following on, the final attribute selection is also described,

where final selection, takes the form of a point and click table, allowing the user to have the final decision on, what attributes to pass forward into any subsequent VPRS analyses.

- Section 6.4 **Summary**. This section summarises the results of the discretisation, feature selection and final attribute selection. Where those attributes selected, are passed into the subsequent software analyses described in Chapters 7 (vein graph) and 8 (re-sampling).

6.1 Parameter Settings

At the initial stage of the developed VPRS software, the analyst is presented with a window that allows them to set a number of parameters and options affecting, as stated, the subsequent pre-processing, feature selection, data mining and evaluation methods employed within the analyses. Figure 6.1.1 displays the screenshot of this set-up window.



File Selection

c:/Datafiles/SavedFiles/Save9.txt

VPRS Analysis

☑ Vein Graph Analysis

Resampling Analyses

☐ Leave One Out

☐ k-fold ☐ Stratified Number of Folds 2

☐ Bootstrapping Number of Iterations 10

Beta Minimum Threshold Value 0.55

Figure 6.1.1: Initial VPRS Software Set-up

From the set-up window shown in Figure 6.1.1, the analyst has the following options:

1. The ability to browse their file system, for the file they wish to analyse, or to re-open a file from

a previously saved analysis.

2. The option to select the basic VPRS vein graph analysis (as will be described in Chapter 7), or selection of the VPRS re-sampling analyses, including, leave-one-out, *k*-fold cross-validation and bootstrapping (as will be described in Chapter 8). On selection of, *k*-fold cross-validation or bootstrapping, the analyst will also have the option to select the number of *k*-folds or bootstrap samples (repetitions), respectively.
3. The analyst has the option to select from two sampling methods. That is, the stratified sub sampling method, which takes a percentage of the data for the validation set, whilst maintaining the decision class distribution, or the statistical sub-sampling method based on Equation 3.1.2.1 described in Chapter 3 section 3.1.2.
4. The analyst is given the option to select which discretisation method they wish to use. Additionally, the analyst may set the number of discretisation intervals they feel is appropriate for the equal-width and equal-frequency methods, or maximum number of intervals for the Minimum Class Entropy (MCE) discretisation method. Note, the FUSINTER discretisation method, autonomously sets the number of intervals and no further parameter setting is required (see Chapter 4 subsection 4.1.3.2).
5. The analyst is also given the option to select which missing value imputation method they wish to use, that is, mean imputation or mean imputation based on the *k*-nearest neighbour method described in Chapter 4 section 4.4. They can also choose between, the three balancing methods as described in Chapter 4 subsection 4.3, namely, up-balancing, down-balancing, and average-balancing. They do however, have the option, not to use balancing.

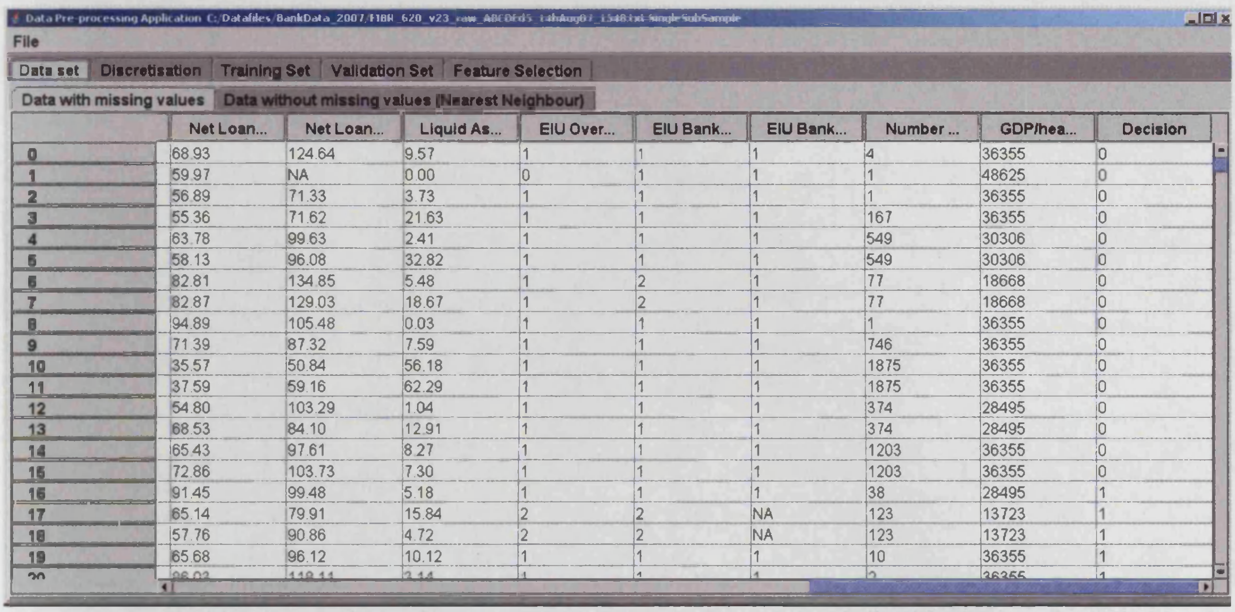
For the analysis of the FIBR data, we have opted for, the statistical sub-sample selection method, and the *k*-nearest neighbour missing value imputation method. After extensive testing, it was found that for the FIBR data, the choice of missing value imputation method had no noticeable impact on the final predictive accuracies.

The developed statistical sub-sampling method was chosen for the reasons given in Chapter 3, in brief because it recognises if a decision class is under represented and takes less objects for the validation set. With regards to balancing (up, down and average-balancing), again after extensive testing it was found that, although balancing improved the predictive accuracies on the under represented classes (mainly 'A' and 'E' grade banks), it proved quite detrimental to the predictive accuracies of well represented classes, and to the overall predictive accuracy. Hence, it was felt that the loss in overall predictive accuracy could not justify the use of balancing in the case of the FIBR data. Perhaps employing more involved balancing methods, to tackle such highly imbalanced data as the FIBR data, may be required. Estabrooks and Japkowocz (2004), describe a multiple re-sampling method, that involves a more complex approach to re-balancing, Grzymala-Busse et al. (2005), suggest simply changing the rule strengths, specifically targeting the rules based on the under represented condition classes.

With regards to selection of discretisation method, it was found that the best predictive accuracies (in the subsequent VPRS analyses), were based on the data discretised using the FUSINTER method. In addition, the FUSINTER method removes the requirement of the analyst to make a possibly subjective decision on how many intervals to use for the undertaken discretisation. Boullé (2004), which includes a survey of recent discretisation methods, also found that compared to the alternative methods (particularly the methods implemented with the developed software), FUSINTER performed better, in terms of the constructed classifiers' predictive accuracies, on both the training and validation sets. Once the analyst is satisfied with the parameter settings, they can proceed to the pre-processing stage, which is described in the following two sections.

6.2 Data Discretisation Results

This section presents the FIBR data set at different stages of the developed pre-processing software, from the initial data set through to the discretised training and validation sets. Figure 6.2.1 displays the opening screenshot of the pre-processing application, specific to the FIBR data set.

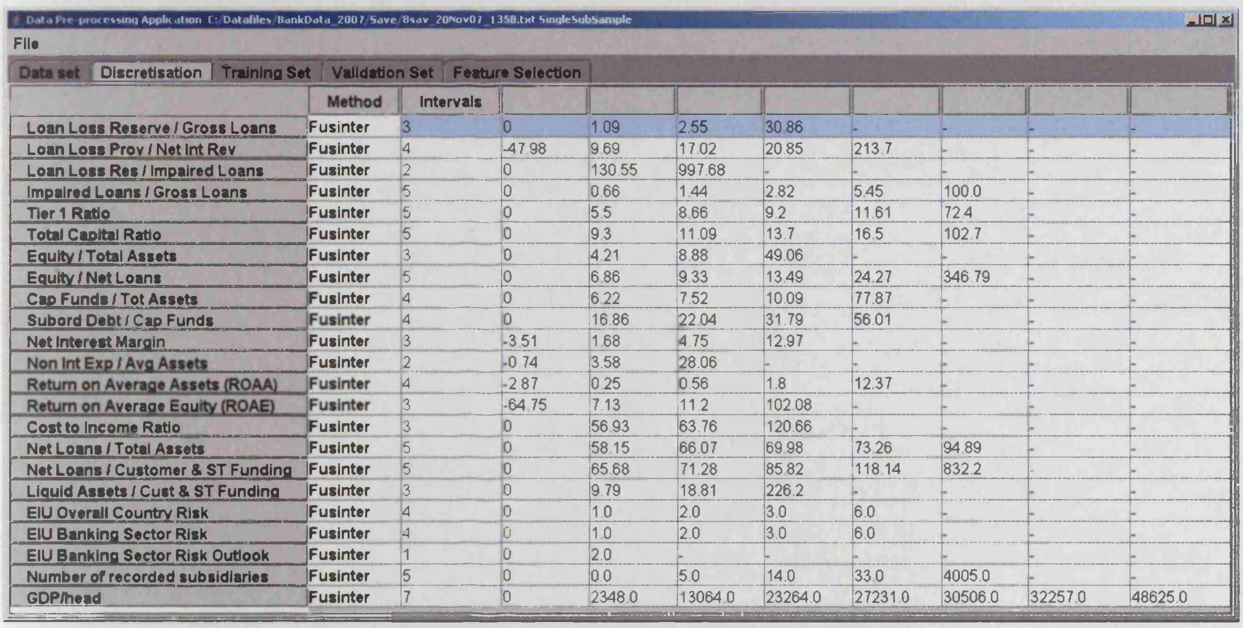

Data Pre-processing Application C:/Datafiles/BankData_2007/FIBR_620_v23_raw_ABCDEd5_14thAug07_1548.txt-SingleSubSample
File
Data set
Discretisation
Training Set
Validation Set
Feature Selection
Data with missing values
Data without missing values (Nearest Neighbour)

	Net Loan...	Net Loan...	Liquid As...	EIU Over...	EIU Bank...	EIU Bank...	Number ...	GDP/hea...	Decision
0	68.93	124.64	9.57	1	1	1	4	36355	0
1	59.97	NA	0.00	0	1	1	1	48625	0
2	56.89	71.33	3.73	1	1	1	1	36355	0
3	55.36	71.62	21.63	1	1	1	167	36355	0
4	63.78	99.63	2.41	1	1	1	549	30306	0
5	58.13	96.08	32.82	1	1	1	549	30306	0
6	82.81	134.85	5.48	1	2	1	77	18668	0
7	82.87	129.03	18.67	1	2	1	77	18668	0
8	94.89	105.48	0.03	1	1	1	1	36355	0
9	71.39	87.32	7.59	1	1	1	746	36355	0
10	35.57	50.84	56.18	1	1	1	1875	36355	0
11	37.59	59.16	62.29	1	1	1	1875	36355	0
12	54.80	103.29	1.04	1	1	1	374	28495	0
13	68.53	84.10	12.91	1	1	1	374	28495	0
14	65.43	97.61	8.27	1	1	1	1203	36355	0
15	72.86	103.73	7.30	1	1	1	1203	36355	0
16	91.45	99.48	5.18	1	1	1	38	28495	1
17	65.14	79.91	15.84	2	2	NA	123	13723	1
18	57.76	90.86	4.72	2	2	NA	123	13723	1
19	65.68	96.12	10.12	1	1	1	10	36355	1
20	96.03	118.11	3.14	1	1	1	2	36355	1

Figure 6.2.1: Initial Screenshot of the Pre-processing Software, Partially Displaying the FIBR Data Set

The left most column of the table exhibited in Figure 6.2.1, displays the index of the objects from zero, as they were input from the data file. The column headings show the attributes names, and the final column to the right (which has been scrolled to, using the horizontal scroll bar), indicates the decision class of each object. Note that, some of the data cells within the table contain the letters 'NA', this indicates a missing value. A separate table (not shown here), selectable on the adjacent tab (labelled 'Data without missing values (Nearest Neighbour)'), displays the same data set but with values evaluated and imputed for the missing values. The full data set is separated into the training set and validation sets (shown later in this section).

Here, the FUSINTER discretisation algorithm has been applied to the training set, and the intervals identified for each attribute are displayed in Figure 6.2.2.


Data Pre-processing Application C:/Datafiles/BankData_2007/Save/8sav_20Nov07_1358.txt SingleSubSample
File
Data set
Discretisation
Training Set
Validation Set
Feature Selection

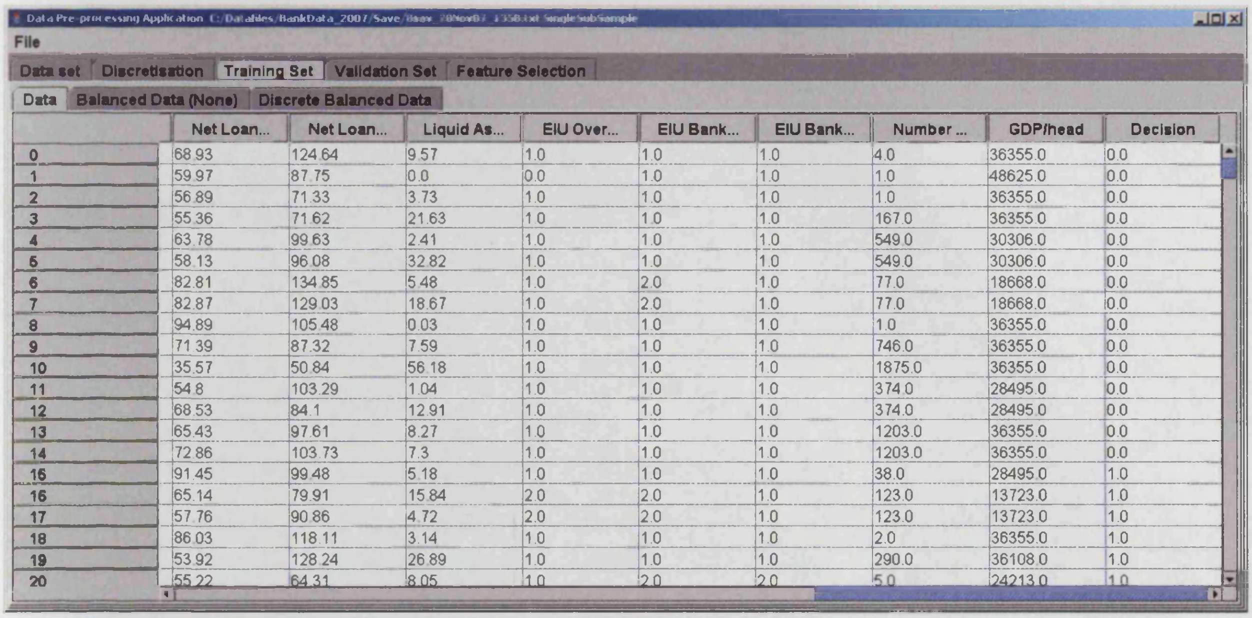
	Method	Intervals								
Loan Loss Reserve / Gross Loans	Fusinter	3	0	1.09	2.55	30.86	-	-	-	-
Loan Loss Prov / Net Int Rev	Fusinter	4	-47.98	9.69	17.02	20.85	213.7	-	-	-
Loan Loss Res / Impaired Loans	Fusinter	2	0	130.55	997.68	-	-	-	-	-
Impaired Loans / Gross Loans	Fusinter	5	0	0.66	1.44	2.82	5.45	100.0	-	-
Tier 1 Ratio	Fusinter	5	0	5.5	8.66	9.2	11.61	72.4	-	-
Total Capital Ratio	Fusinter	5	0	9.3	11.09	13.7	16.5	102.7	-	-
Equity / Total Assets	Fusinter	3	0	4.21	8.88	49.06	-	-	-	-
Equity / Net Loans	Fusinter	5	0	6.86	9.33	13.49	24.27	346.79	-	-
Cap Funds / Tot Assets	Fusinter	4	0	6.22	7.52	10.09	77.87	-	-	-
Subord Debt / Cap Funds	Fusinter	4	0	16.86	22.04	31.79	56.01	-	-	-
Net Interest Margin	Fusinter	3	-3.51	1.68	4.75	12.97	-	-	-	-
Non Int Exp / Avg Assets	Fusinter	2	-0.74	3.58	28.06	-	-	-	-	-
Return on Average Assets (ROAA)	Fusinter	4	-2.87	0.25	0.56	1.8	12.37	-	-	-
Return on Average Equity (ROAE)	Fusinter	3	-64.75	7.13	11.2	102.08	-	-	-	-
Cost to Income Ratio	Fusinter	3	0	56.93	63.76	120.66	-	-	-	-
Net Loans / Total Assets	Fusinter	5	0	58.15	66.07	69.98	73.26	94.89	-	-
Net Loans / Customer & ST Funding	Fusinter	5	0	65.68	71.28	85.82	118.14	832.2	-	-
Liquid Assets / Cust & ST Funding	Fusinter	3	0	9.79	18.81	226.2	-	-	-	-
EIU Overall Country Risk	Fusinter	4	0	1.0	2.0	3.0	6.0	-	-	-
EIU Banking Sector Risk	Fusinter	4	0	1.0	2.0	3.0	6.0	-	-	-
EIU Banking Sector Risk Outlook	Fusinter	1	0	2.0	-	-	-	-	-	-
Number of recorded subsidiaries	Fusinter	5	0	0.0	5.0	14.0	33.0	4005.0	-	-
GDP/head	Fusinter	7	0	2348.0	13064.0	23264.0	27231.0	30506.0	32257.0	48625.0

Figure 6.2.2: Data Discretisation Table

The table row headings displayed in Figure 6.2.2, indicate the attributes input from the data file. The first column displays the choice of discretisation method used to discretise the data (here, FUSINTER was used for all attributes), the choice of discretisation was made in the initial set-up (see section 6.1). The second column displays the number of intervals associated with each attribute, and the interval ranges themselves are displayed in the proceeding columns. Note that the default lowest initial interval value, associated with any attribute is zero, except where the attribute in question, contained negative values. For example, Loan Loss Prov/Net Int Rev's lowest value was –47.98 and hence its first interval is the range [–47.98, 9.96].

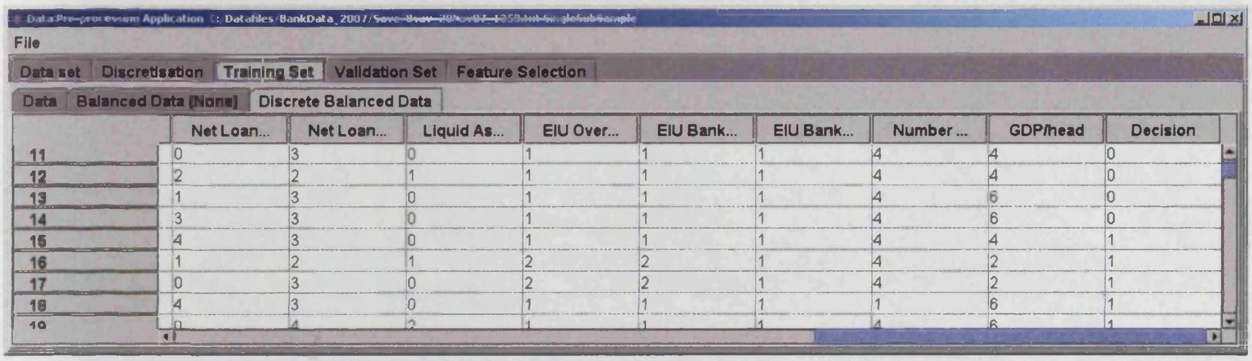
Looking at the first row, and taking the attribute Loan Loss Reserve/Gross Loans as an example (highlighted in Figure 6.2.2), it has been discretised into three intervals [0.0, 1.09], (1.09, 2.55] and (2.55, 30.86]. Hence, the data associated with this attribute, is recoded into the discrete values '0', '1' and '2', respectively (shown next in Figure 6.2.4). Note, that when these intervals are used to discretise the validation set, there is the possibility that a value within the validation set, may lie outside the lower (0.0) and upper (30.86) bounds. In this circumstance the value will be discretised to its nearest associated interval, for example, a value less than 0.0 will be set to 0 and any value greater than 30.86 would be set to 2.

The analyst is provided with separate tables containing both the training and validation sets. Figures 6.2.3 and 6.2.4 display the training data at the pre-discretised (labelled 'Data' under the 'Training Set' tab) and post-discretised (labelled 'Discrete Balanced Data' under the 'Training Set' tab) stages, respectively. There is a further table (not shown here), which if balancing had been used, allows the analyst to inspect the balanced version of the training data (labelled 'Balanced Data (None)' under the 'Training Set' tab).



	Net Loan...	Net Loan...	Liquid As...	EIU Over...	EIU Bank...	EIU Bank...	Number ...	GDP/head	Decision
0	68.93	124.64	9.57	1.0	1.0	1.0	4.0	36355.0	0.0
1	59.97	87.75	0.0	0.0	1.0	1.0	1.0	48625.0	0.0
2	56.89	71.33	3.73	1.0	1.0	1.0	1.0	36355.0	0.0
3	55.36	71.62	21.63	1.0	1.0	1.0	167.0	36355.0	0.0
4	63.78	99.63	2.41	1.0	1.0	1.0	549.0	30306.0	0.0
5	58.13	96.08	32.82	1.0	1.0	1.0	549.0	30306.0	0.0
6	82.81	134.85	5.48	1.0	2.0	1.0	77.0	18668.0	0.0
7	82.87	129.03	18.67	1.0	2.0	1.0	77.0	18668.0	0.0
8	94.89	105.48	0.03	1.0	1.0	1.0	1.0	36355.0	0.0
9	71.39	87.32	7.59	1.0	1.0	1.0	746.0	36355.0	0.0
10	35.57	50.84	56.18	1.0	1.0	1.0	1875.0	36355.0	0.0
11	54.8	103.29	1.04	1.0	1.0	1.0	374.0	28495.0	0.0
12	68.53	84.1	12.91	1.0	1.0	1.0	374.0	28495.0	0.0
13	65.43	97.61	8.27	1.0	1.0	1.0	1203.0	36355.0	0.0
14	72.86	103.73	7.3	1.0	1.0	1.0	1203.0	36355.0	0.0
15	91.45	99.48	5.18	1.0	1.0	1.0	38.0	28495.0	1.0
16	65.14	79.91	15.84	2.0	2.0	1.0	123.0	13723.0	1.0
17	57.76	90.86	4.72	2.0	2.0	1.0	123.0	13723.0	1.0
18	86.03	118.11	3.14	1.0	1.0	1.0	2.0	36355.0	1.0
19	53.92	128.24	26.89	1.0	1.0	1.0	290.0	36108.0	1.0
20	55.22	64.31	8.05	1.0	2.0	2.0	5.0	24213.0	1.0

Figure 6.2.3: Training Set, Pre-discretisation

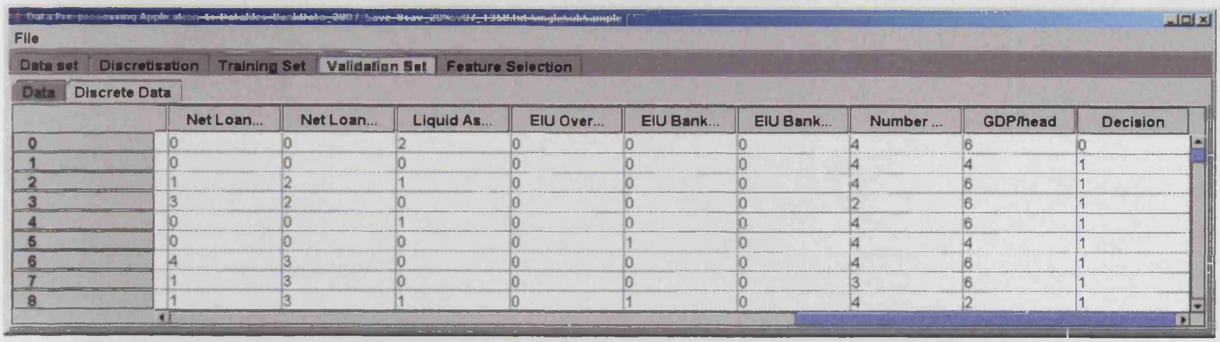


	Net Loan...	Net Loan...	Liquid As...	EIU Over...	EIU Bank...	EIU Bank...	Number ...	GDP/head	Decision
11	0	3	0	1	1	1	4	4	0
12	2	2	1	1	1	1	4	4	0
13	1	3	0	1	1	1	4	6	0
14	3	3	0	1	1	1	4	6	0
15	4	3	0	1	1	1	4	4	1
16	1	2	1	2	2	1	4	2	1
17	0	3	0	2	2	1	4	2	1
18	4	3	0	1	1	1	1	6	1
19	0	4	2	1	1	1	4	6	1

Figure 6.2.4: Training Set, Post-discretisation

Note that the missing values denoted previously by the letters 'NA', have been replaced within the table displayed in Figure 6.2.3, with values evaluated using the *k*-nearest neighbour method. With regards to the number of objects taken for the training set, 405 objects were taken, this value was obtainable by scrolling the vertical scroll bar to the bottom of the table in Figure 6.2.3 (the breakdown of training and validation set objects is described next).

The validation set is discretised using the intervals calculated from the training set, shown previously. The results of which are shown below in Figure 6.2.5.



	Net Loan...	Net Loan...	Liquid As...	EIU Over...	EIU Bank...	EIU Bank...	Number ...	GDP/head	Decision
0	0	0	2	0	0	0	4	6	0
1	0	0	0	0	0	0	4	4	1
2	1	2	1	0	0	0	4	6	1
3	3	2	0	0	0	0	2	6	1
4	0	0	1	0	0	0	4	6	1
5	0	0	0	0	1	0	4	4	1
6	4	3	0	0	0	0	4	6	1
7	1	3	0	0	0	0	3	6	1
8	1	3	1	0	1	0	4	2	1

Figure 6.2.5: Discretised Validation Set

Note, from inspecting the table displayed in Figure 6.2.5, that only one object (bank) from the 'A' grade banks (decision class '0'), has been included in the validation set. This is a result of the statistical sub-sampling method, recognising that the class is under represented and that the majority of the 'A' grade banks are required for training purposes. With regards to the number of objects selected for validation, 215 objects were selected, again this value was obtainable by scrolling the vertical scroll bar to the bottom of the table shown in Figure 6.2.5.

The following Table 6.1.1, displays the breakdown of the number of objects sampled from each class for use in the training set, with the remaining objects used in the validation set (displayed previously in Chapter 3 section 3.1.2).

Bank Grade (Decision Class)	Number of Objects	Training (Taken)	Validation (Remaining)
A (0)	16	15	1
B (1)	319	177	142
C (2)	163	115	48
D (3)	107	84	23
E (4)	15	14	1
Totals	620	405	215

Table 6.1.1: Training Set and Validation Set Sample Sizes

With regards to the under represented classes within Table 6.1.1, as expected, less objects were left for the purpose of validation. The proportional split between the training and validation sets is around two thirds, one third respectively, which is in line with the recommendations quoted in the associated literature (Weiss and Kulikowski, 1991; Han and Kamber, 2006).

6.3 Feature Selection

This section is split into three subsections which present the results of the two implemented feature selection algorithms, namely ReliefF and the more novel algorithm proposed by Beynon (2004) based on RST, referred to here as the RST_FS method (RST Feature Selection). The first and second subsections describe graphs developed in association with the ReliefF algorithm and the RST_FS algorithm, respectively. The final subsection describes how the software collates the results of the feature selection methods into a table, to be used by the analyst to select the attributes to be passed forward into any subsequent VPRS analysis.

6.3.1 Results of ReliefF

Within the developed software, the ReliefF algorithm, described in Chapter 4, is applied to both the pre-discretised training set and the post-discretised data set. The results for both variations of the application of the ReliefF algorithm are presented under two separate sub-categories within the feature selection element of the software, namely ReliefFC (C signifying continuous data) and ReliefFD (D signifying discrete data). Here, only the results of ReliefFD are discussed because the VPRS analysis is based on discretised data, hence, it was thought more pertinent to select attributes identified by ReliefFD at the final attribute selection stage (described in subsection 6.2.3).

Three graphs were developed to elucidate the ReliefF process, and to aid the analyst in their attribute selection decisions. The graphs have also, to some extent, assited the development of the ReliefF algorithm (explained next).

The first of these three graphs is shown in Figure 6.3.1.1, it represents the evaluated quality estimation[18] value for each attribute, over values of *m*, where the value of *m* is increased in increments of five, between five and the number of objects in the training data set, thus the highest value of *m* shown in being Figure 6.3.1.1 405.[19] To recap, the *m* value essentially represents the number of randomly chosen objects from the training set. The lowest value of *m* chosen here (for the graph) being five, that is five objects selected at random (without replacement).

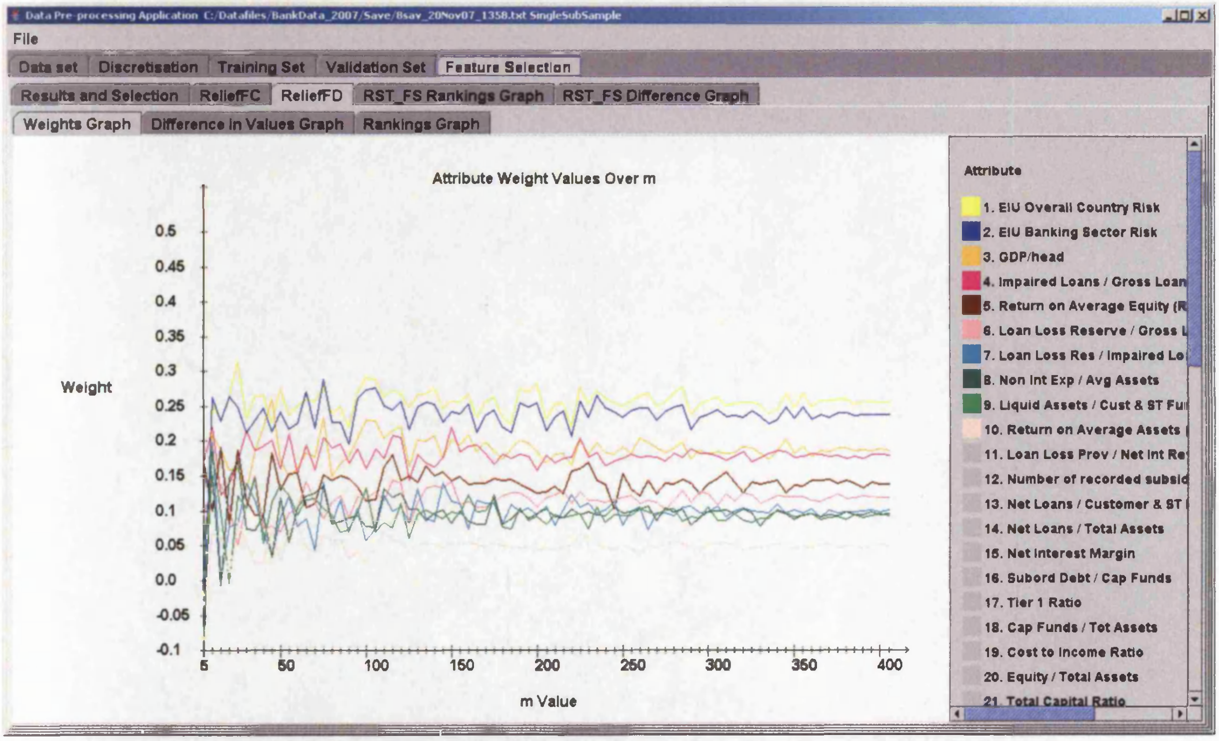


Figure 6.3.1.1: ReliefFD Weights Graph Over the Range of *m* Values

The colour coded legend to the right of the graph in Figure 6.3.1.1, indicates the final ranked positions of all the attributes based on the highest value of *m*. Only the top ten ranked attributes are plotted on the graph, and are coloured other than grey in the legend (described in more detail later in this section).

It is clear from the graph in Figure 6.3.1.1, that the quality estimation associated with each

[18] Robnik-Šikonja and Kononenko (2003), describe an attribute's quality estimation, as a measure of an attributes ability to distinguish between (dis)similar objects. With a larger quality estimation indicating an attribute has a good level of distinguishing capability.

[19] The software implementation of the ReliefF algorithm, by default, sets the highest *m* value on the graph, to the size of the training set. That is, if the training set had not been a multiple of five, e.g. 403, the final *m* value would have been set to 403.

attribute are volatile (inconsistent) for lower values of *m*, but become more consistent as *m* increases and more objects are utilised during the ReliefF algorithm. Moreover, the final evaluated quality estimates, which in theory should be the most accurate estimation of the quality associated with each attribute (Robnik-Šikonja and Kononenko, 2003), appear to converge to specific values. To further demonstrate this convergence, the graph presented in Figure 6.3.1.2 displays the difference between each consecutive quality estimate over consecutive values of *m*, associated with each attribute.

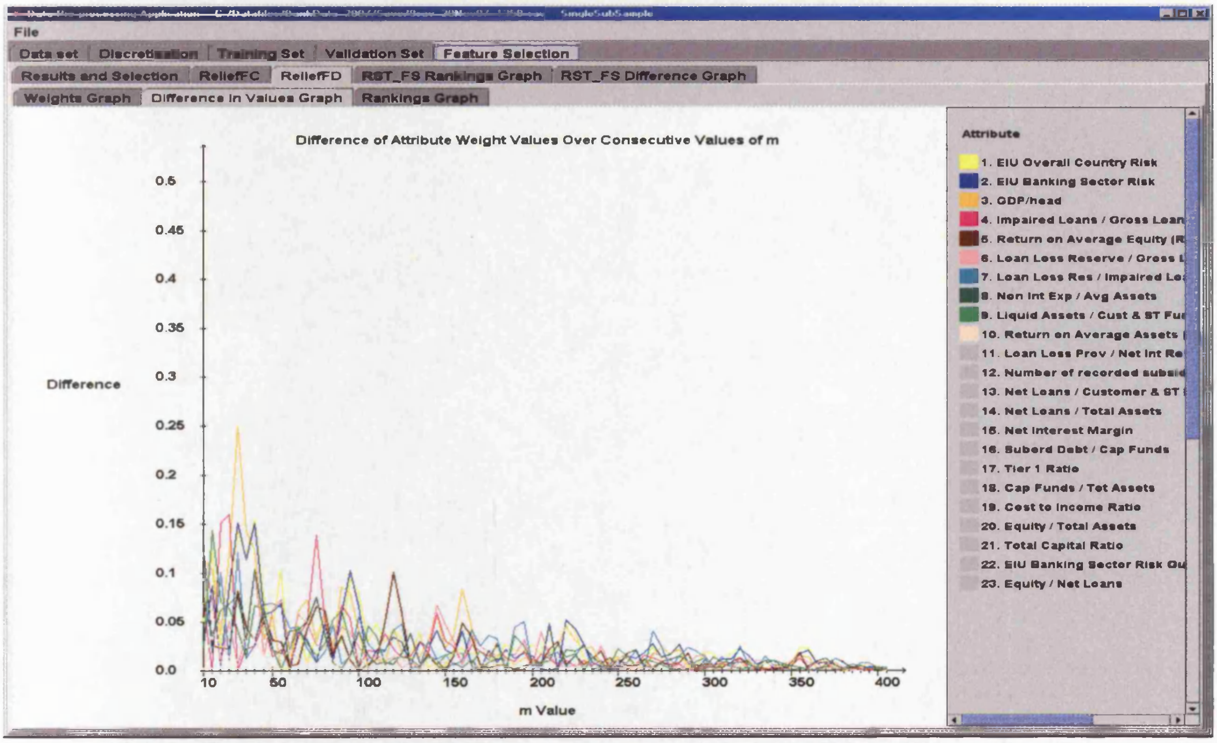

SingleSubSample
File
Data set
Discretisation
Training Set
Validation Set
Feature Selection
Results and Selection
ReliefFC
ReliefFD
RST_FS Rankings Graph
RST_FS Difference Graph
Weights Graph
Difference in Values Graph
Rankings Graph
Difference of Attribute Weight Values Over Consecutive Values of m
Difference
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0.0
10
50
100
150
200
250
300
350
400
m Value
Attribute
🟨 1. EIU Overall Country Risk
🟦 2. EIU Banking Sector Risk
🟧 3. GDP/head
🟥 4. Impaired Loans / Gross Loan
🟫 5. Return on Average Equity (R
6. Loan Loss Reserve / Gross L
7. Loan Loss Res / Impaired Lo
🟩 8. Non Int Exp / Avg Assets
🟩 9. Liquid Assets / Cust & ST Fu
10. Return on Average Assets
11. Loan Loss Prov / Net Int Re
12. Number of recorded subsid
13. Net Loans / Customer & ST
14. Net Loans / Total Assets
15. Net Interest Margin
16. Suberd Debt / Cap Funds
17. Tier 1 Ratio
18. Cap Funds / Tot Assets
19. Cost to Income Ratio
20. Equity / Total Assets
21. Total Capital Ratio
22. EIU Banking Sector Risk Ou
23. Equity / Net Loans

Figure 6.3.1.2: Convergence of the Difference Between ReliefFD Weight Values Over Consecutive Values of *m*

Although the graph presented in Figure 6.3.1.2, does not provide any extra information to aid the analyst in their final attribute selection decision, it does further emphasize, the initial volatility in the evaluated quality estimates, and the convergence of the values to a specific stable value, for higher values of *m*.

With regards to both the graphs presented in this subsection so far, it is difficult to asses the rank positions (highest to lowest) associated with each attribute for the whole range of *m* values. Hence, the third and final graph presented in this subsection, presented in Figure 6.3.1.3, shows the rank positions associated with each attribute over the range of *m* values. Where for each value of *m*, the attribute with the highest associated quality estimate is ranked as first, and the attribute with the lowest associated quality estimate, is ranked as last. There are 23 attributes associated with this pre-processing analysis, hence the lowest possible position an attribute can take is 23rd. The graph only displays the top 10 ranked attributes, based on their final ranking position (i.e. for *m* = 405).

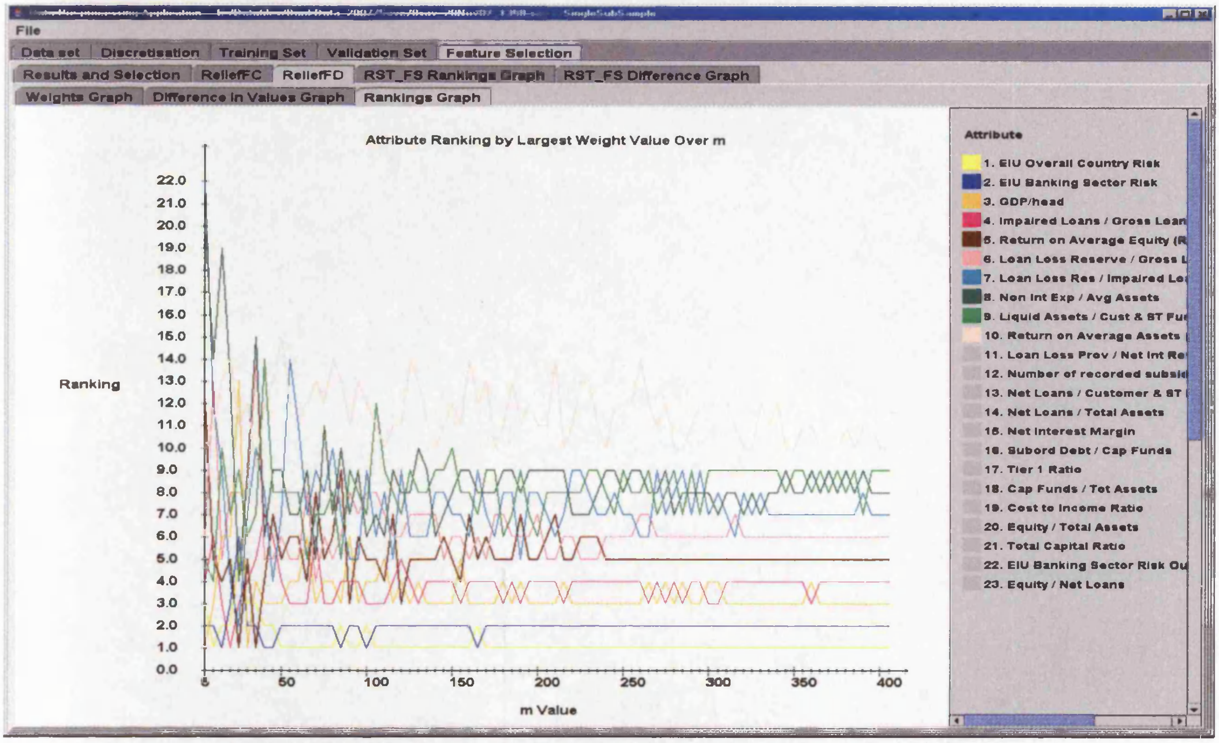

File
Data set
Discretisation
Training Set
Validation Set
Feature Selection
Results and Selection
ReliefFC
ReliefFD
RST_FS Rankings Graph
RST_FS Difference Graph
Weights Graph
Difference in Values Graph
Rankings Graph
Attribute Ranking by Largest Weight Value Over m
Ranking
22.0
21.0
20.0
19.0
18.0
17.0
16.0
15.0
14.0
13.0
12.0
11.0
10.0
9.0
8.0
7.0
6.0
5.0
4.0
3.0
2.0
1.0
0.0
5
50
100
150
200
250
300
350
400
m Value
Attribute
1. EIU Overall Country Risk
2. EIU Banking Sector Risk
3. GDP/head
4. Impaired Loans / Gross Loan
5. Return on Average Equity (R
6. Loan Loss Reserve / Gross L
7. Loan Loss Res / Impaired Lo
8. Non Int Exp / Avg Assets
9. Liquid Assets / Cust & ST Fu
10. Return on Average Assets
11. Loan Loss Prov / Net Int Re
12. Number of recorded subsid
13. Net Loans / Customer & ST
14. Net Loans / Total Assets
15. Net Interest Margin
16. Subord Debt / Cap Funds
17. Tier 1 Ratio
18. Cap Funds / Tot Assets
19. Cost to Income Ratio
20. Equity / Total Assets
21. Total Capital Ratio
22. EIU Banking Sector Risk Ou
23. Equity / Net Loans

Figure 6.3.1.3: Attribute Rankings Over the Range of *m* Values

The graph presented in Figure 6.3.1.3, does not convey the convergence effect of the ReliefF algorithm over the range of *m* values, as clearly as the previous two graphs. It does however, indicate the inconsistent nature of the rank positions associated with each attribute for lower values of *m*, which in general, becomes more consistent for higher values of *m*.

For the top ranked attributes 1st (yellow) to 6th (pink), there is consistency for the larger values of *m*, and total consistency from *m* greater than, or equal to 380. The attributes ranked 7th (turquoise) to 9th (light green), are consistently ranked for values of *m* over 265, between the 7th and 9th positions. The analyst cannot take their final rankings as being totally conclusive, but can say with confidence, that their rankings are in the range seven to nine. Similarly for the 10th, 11th, and 12th ranked attributes (11 and 12 not shown on graph) their final rank position may be inconclusive (in the range 10 to 12), but the analyst can say with some confidence that the tenth ranked attribute is likely to have a ranking in the range ten to twelve for values of *m* greater than 360.

The graphs described within this subsection, provide the analyst with further insight into the final rank positions that are indicated by the ReliefF algorithm. Interestingly, and of particular relevance to this analysis, they clearly indicate that selection of the largest *m* value is of importance. It should be noted here though, that a deterministic approach to ReliefF was implemented (as opposed to a random sampling approach), for reasons described next. Finally, the graphs indicate that the analyst cannot take the final rank positions as conclusive. This was demonstrated within the final graph shown in Figure 6.3.1.3. Hence, the analyst should only use the results to aid their final decision, where they may consider other factors, such as, the results from alternative feature selection algorithms.

There were a number of issues relating to ReliefF, that require further consideration. Firstly, relating to the stability of the results. The number of iterations *m*, affected the precision of the attributes' final rankings. Increasing the number of iterations *m*, led to more stable quality estimates (because of the relative increase in sample size as *m* is increased). That is, for larger values of *m*, the associated quality estimates were more consistent between two or more runs of the algorithm (as demonstrated by Figure 6.3.1.1 and Figure 6.3.1.2).

It was not specifically stated by Robnik-Šikonja and Kononenko (2003), whether ReliefF was based on sampling with or without replacement (sampling with replacement seemed inferred, see Chapter 4 subsection 4.2.3.1 for algorithm description). In Robnik-Šikonja and Kononenko (2003), based on empirical evidence they suggested that, they typically observed stable results within twenty to fifty iterations. They also demonstrate an example that requires 300 iterations, but our results were contradictory. We required *m* to be much higher, up to twenty times the number of objects within the training set (for the FIBR data we required 10,000 iterations to achieve stable results, not shown here in software).

Interestingly, Kohavi and John (1997, pp. 7) reported a similar problem when utilising the original Relief algorithm, and stated that:

> "...we found significant variance in the relevance rankings given by Relief. Since Relief randomly samples instances and their neighbors from the training set, the answers it gives are unreliable without a very high number of samples. In our experiments, the required number of samples was of the order of two to three times the number of cases in the training set. We were worried by this variance, and implemented a deterministic version of Relief that uses all instances..."

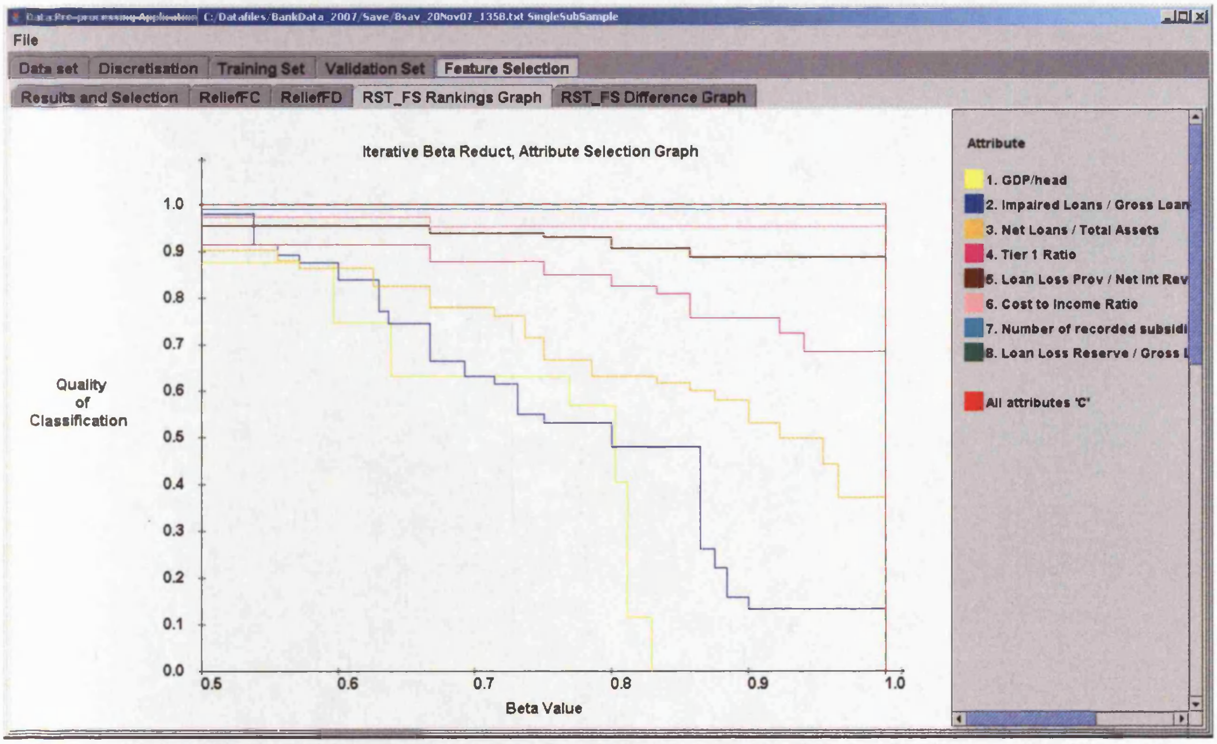
Here, we implemented a similar deterministic approach to the ReliefF algorithm, by employing sampling without replacement, and allowing for *m* to equal up to the number of objects within the training set *n*. For $m = n$, it is effectively the same as sequentially going through and selecting each object within the training set once, as opposed to randomly sampling objects. Here though, random sampling was used, as it enabled the graphical demonstration, of how, the results converged to a stable solution as *m* tended towards *n*.

Interestingly, as *m* tends to infinity, the results based on sampling 'with' replacement converge to the solution based on sampling 'without' replacement (the deterministic approach). Kohavi and John (1997) reported a similar observation with Relief. Note however, sampling with replacement greatly increases the processing time as more iterations *m* are required before stability of results is achieved, hence sampling without replacement is also a more efficient alternative.

There may be a number of reasons why Robnik-Šikonja and Kononenko (2003) suggested using such relatively low values of *m*. Their empirical results appear to be based on large simulated data sets (up to 7,000 objects), where sampling with or without replacement may not have been an issue. Whereas, our preliminary tests included relatively smaller data sets (500 objects), that were imbalanced and had multiple decision classes. From these findings, for small difficult data sets (imbalanced, missing data, multiple decision classes), it may be pertinent to prescribe implementing the deterministic approach suggest here, or more simply, implementing a version of ReliefF that utilises a sequential approach that selects all *n* objects once (possibly using stratification for larger datasets).

6.3.2 RST Feature Selection Method Results

Continuing with the results of feature selection based on the FIBR data, this subsection describes two graphs that were developed in association with the RST_FS algorithm described by Beynon (2004). The first graph to be presented, as shown in Figure 6.3.2.1, is an adaptation of the three dimensional graph system presented in Beynon (2004). That is, a colour coded system has been utilised to distinguish between the attributes, as opposed to using a third axis with attribute names along the scale.



C:/Datafiles/BankData_2007/Save/8sav_20Nov07_1358.txt SingleSubSample
File
Data set
Discretisation
Training Set
Validation Set
Feature Selection
Results and Selection
ReliefFC
ReliefFD
RST_FS Rankings Graph
RST_FS Difference Graph
Iterative Beta Reduct, Attribute Selection Graph
Quality of Classification
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.5
0.6
0.7
0.8
0.9
1.0
Beta Value
Attribute
1. GDP/head
2. Impaired Loans / Gross Loan
3. Net Loans / Total Assets
4. Tier 1 Ratio
5. Loan Loss Prov / Net Int Rev
6. Cost to Income Ratio
7. Number of recorded subsidi
8. Loan Loss Reserve / Gross L
All attributes 'C'

Figure 6.3.2.1: RST_FS Graph, Depicting the Difference in QoC Over the Range of β Values for each Subset of Selected Attributes

To recap, the RST_FS graph as shown in Figure 6.3.2.1, indicates the disparity, or notional distance between the Quality of Classification (QoC) over the range of β, for the selected attributes compared to the full set of attributes. The initial attribute is selected by taking the attribute whose

QoC is closest to the full set of attributes' QoC. This initial attribute is successively augmented with additional attributes, which in combination with the previous selected attributes, offer the greatest decrease in distance to the QoC of the full set of attributes. The process finishes when either the distance is zero, or all attributes have been successively selected. In the case presented in Figure 6.3.2.1, it took eight attributes to attain a QoC equal to the full set of attributes.

The legend to the right of the graph, indicates the order in which the attributes were selected and hence ranked. Here, with GDP/head ranked as first (yellow) and Loan Loss Reserve/Gross Loans ranked last (dark green).

The full set of attributes is represented by the dashed red line, because at the point that the subset of selected attributes has equal QoC as the full set, the lines representing the selected subset and the full set are identical (overlapping). Hence, the selected subset also appears as a dashed line, indicated here by the dashed dark green.

Clearly, from Figure 6.3.2.1, GDP/head appears to make the most impact in terms of eliminating the notional distance between the QoC of the full set and that of the selected subset of attributes. To elucidate how much each additional selected attribute contributes to bringing the subset of selected attributes closer to the full set of attributes' QoC, a second graph was developed and is presented in Figure 6.3.2.2.

As described with regards to the previous graph, GDP/head has the most impact on reducing the distance between the QoC of the selected subset and full set of attributes, over the domain of β. Within Figure 6.3.2.2, this "impact" (importance) is depicted by the gradient of the slope, with a steeper slope indicating more impact. The colour coded dashed lines help indicate how much "numerically" each attribute impacts on the feature selection process, for example, GDP/head reduces the notional distance from 1.0 to 0.55. The following four attributes ranked 2nd to 5th appear to be of relatively equal importance, in terms of the successive reductions in distance. The attributes ranked 6th and 7th, appear to have similar importance but less than the previous attributes, and the final attribute ranked 8th, appears to contribute the least to the selection process.

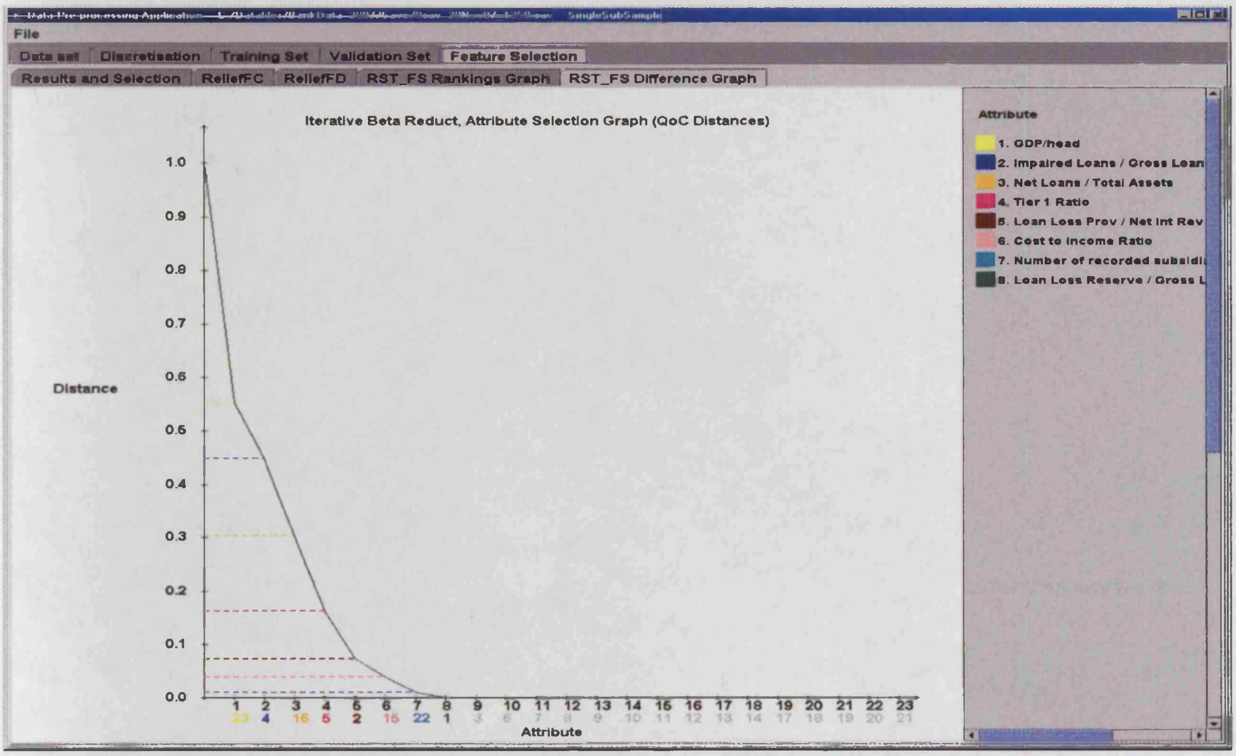
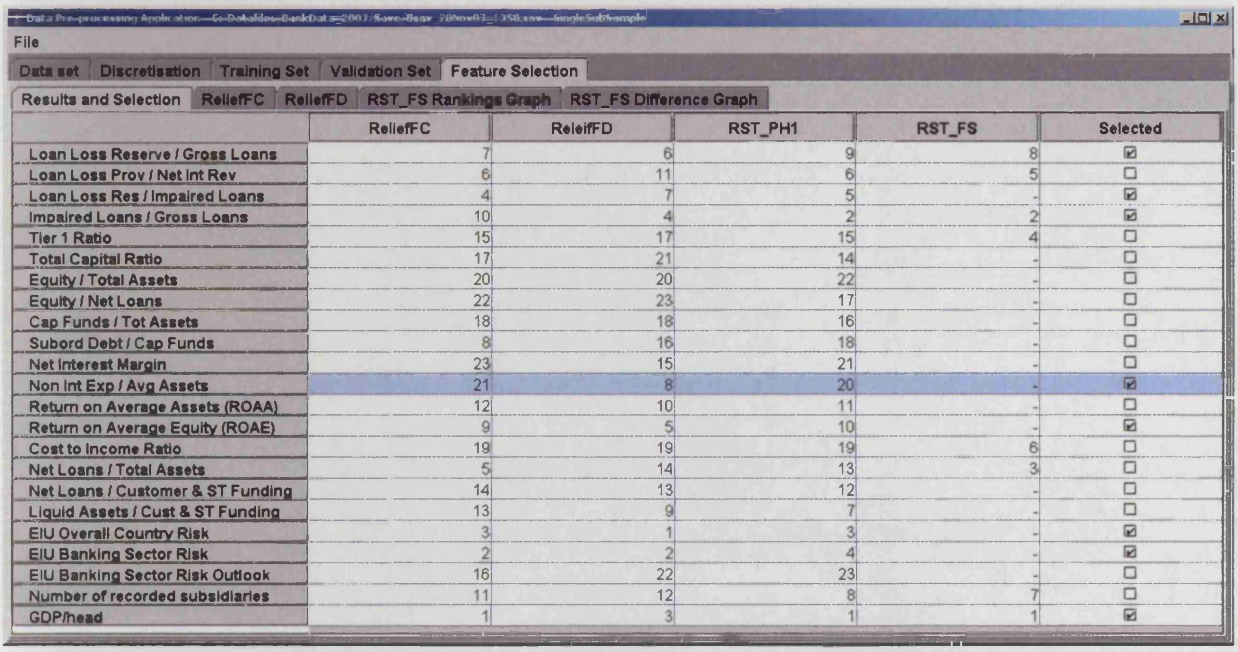


Figure 6.3.2.2: QoC 'Distance' Gain for each Additional Selected Attribute

As with the graphs described in the previous subsections, these graphs present the analyst with additional information, such as the contribution/importance of each attribute during the feature selection precess. This additional information will aid the analyst in making their final choice of attributes to pass forward into any subsequent VPRS analysis (vein graph or re-sampling). Selection of the final set of attributes is discussed in the next subsection.

6.3.3 Final Attribute Selection

This section presents the final screenshot of the pre-processing software to be discussed within this chapter. Figure 6.3.3.1 displays the final rankings from each feature selection method collated into one table.



	ReliefFC	ReleifFD	RST_PH1	RST_FS	Selected
Loan Loss Reserve / Gross Loans	7	6	9	8	☑
Loan Loss Prov / Net Int Rev	6	11	6	5	□
Loan Loss Res / Impaired Loans	4	7	5	-	☑
Impaired Loans / Gross Loans	10	4	2	2	☑
Tier 1 Ratio	15	17	15	4	□
Total Capital Ratio	17	21	14	-	□
Equity / Total Assets	20	20	22	-	□
Equity / Net Loans	22	23	17	-	□
Cap Funds / Tot Assets	18	18	16	-	□
Subord Debt / Cap Funds	8	16	18	-	□
Net Interest Margin	23	15	21	-	□
Non Int Exp / Avg Assets	21	8	20	-	☑
Return on Average Assets (ROAA)	12	10	11	-	□
Return on Average Equity (ROAE)	9	5	10	-	☑
Cost to Income Ratio	19	19	19	6	□
Net Loans / Total Assets	5	14	13	3	□
Net Loans / Customer & ST Funding	14	13	12	-	□
Liquid Assets / Cust & ST Funding	13	9	7	-	□
EIU Overall Country Risk	3	1	3	-	☑
EIU Banking Sector Risk	2	2	4	-	☑
EIU Banking Sector Risk Outlook	16	22	23	-	□
Number of recorded subsidiaries	11	12	8	7	□
GDP/head	1	3	1	1	☑

Figure 6.3.3.1: Final Attribute Ranking and Selection

The row headings within the table displayed in Figure 6.3.3.1, contain the attribute names, and the first four column headings indicate the feature selection methods used. The values within the columns represent the final rank position for each attribute associated with each feature selection method. The first two columns ReliefFC and ReliefFD's rankings range from 1st to 23rd. The RST_FS method's rankings range from 1st to the number of attributes identified by the feature selection algorithm, in this case 8, hence a lowest ranking of 8th.

During the initial development of the software, in particular regards to RST_FS, consideration was given to whether there may be circumstances where the analyst may wish to select more attributes than those identified by the RST_FS method, hence the RST phase one (RST_PH1) approach was developed (see Chapter 4 section 4.2.3.2 for more development explanation). RST_PH1 simply ranks the attributes by their notional distance to the full set of attributes (based on QoC over the range of β). Essentially, RST_PH1 is the first phase of the RST_FS algorithm (as described in Chapter 4 section 4.2.3.2), and as such, it can be clearly seen that, GDP/Head is ranked both first for RST_PH1 and RST_FS in Figure 6.3.3.1, because GDP/head is notionally the closest attribute in terms of QoC to the full set of attributes available in the data set.

The RST_PH1 method, was later dismissed as a creditable method for choosing further attributes in addition to those identified by the RST_FS method, because it was felt that the rankings associated with the RST_PH1 method had little meaning or relevance to those already ranked by the RST_FS method. Although, it is interesting to see that the rankings indicated by ReliefFC, ReliefFD and RST_PH1 do appear to have a high level of correlation. Table 6.1.2 indicates the Spearman's rank correlations between those three methods.

	ReliefFC	ReliefFD	RST_PH1
ReliefFC	-	0.727*	0.810*
ReliefFD	0.727*	-	0.810*
RST_PH1	0.810*	0.810*	-

Table 6.1.2: Spearman's Rank Correlations Between Three Feature Selection Methods
* Correlation significant at the 0.01 level (1-tailed)

Although the RST_PH1 is not creditable as a method for choosing further attributes in addition to those identified by RST_FS, it does appear, that based on the correlations in Table 6.1.2, it may still be useful as a simple feature selection method in itself.

It was found that, the RST_FS method, perhaps due to its stepwise nature (see Chapter 4 section 4.2.1.3), was too restrictive as a feature selection method. Moreover, subsequent VPRS analyses based on the RST_FS method typically only identified one reduct based on all the attributes input into the VPRS analysis. Hence, it was hypothesised that RST_FS was tending to identify a single β-reduct, rather than a set of attributes allowing for a range of β-reducts. This hypothesis is further strengthened by the fact that the RST_FS algorithm draws many similarities with the QuickReduct heuristic described by Chouchoulas and Shen (2001), which is a suboptimal method for identifying reducts.

The subsequent VPRS analyses based on attributes identified by the feature selection methods other than RST_FS, namely, ReliefFC, ReliefFD and RST_PH1, tended to, provide more of a varied range of β-reducts, in terms of the number of β-reducts, the number of attributes associated with those β-reducts, the QoC and β-ranges associated with the set of β-reducts (only the results associated with ReliefFD are shown in the following chapter, due to constraints on the size of this dissertation).

Finally, the table shown in Figure 6.3.3.1 allows the analyst to select, through the use of tick boxes, which attributes they wish to pass forward into the subsequent VPRS analysis. Here, the top eight ranked attributes identified by the ReliefFD method were chosen (for reasons explained next). As VPRS utilises the discrete training data, it was concluded that, the attributes identified by ReliefFC were less appropriate than those identified by ReliefFD, since ReliefFD was based on the discrete training data. Additionally, the developed RST_PH1 method was deemed as experimental, whereas ReliefF is an established method, hence it was decided to use ReliefFD rather than RST_PH1.

With regards to the number of attributes selected, for the subsequent VPRS analyses; initially ten attributes were considered, but it was found that the subsequent VPRS analyses, did not yield favourable results. That is, it typically appeared to default to identifying one β-reduct, based on the full set of attributes, which were associated with a large number of weak rules, with no apparent general trend. Weiss and Kulikowski (1991) appear to suggest an explanation, and state that, in certain situations, having too many attributes relative to the number objects, can result in poorer predictive performance, and overfitting. They stated further that (pp. 73):

> "While we like to think that the more information the better, one needs a corresponding increase in the number of samples to determine what information is useful."

By only taking eight attributes, this improved our range of β-reducts and identified β-reducts with more general sets of rules, with good predictive accuracies (based on a number of subsequent VPRS analyses). It was reasoned that, by including too many attributes in the analyses, the number of condition classes would increase (because of the potential extra combinations of attribute values associated with the objects), which would have the effect of creating many weak rules (say based on condition classes only containing one object). In effect, by including too many attributes, it could be considered as, saturating the VPRS analysis with data (in the attribute sense). One alternative option would be to reduce the granularity of the data, by either using an alternative discretisation method or relaxing the parameters within the FUSINTER algorithm, to allow it to reduce the number of intervals associated with each attribute (see Chapter 4 section 4.1.3.2 for FUSINTER parameter setting).

As described previously, the eight attributes identified by RST_FS were too impacting as a feature selection method, because, there was no further scope for identification of β-reducts within the subsequent VPRS analyses. However, RST_FS may be a good indicator, of how many attributes were required in the subsequent analysis, which could aid the analyst.

The eight attributes shown as having been selected in Figure 6.3.3.1, are subsequently utilised in Chapters 7 and 8. Additionally, at this stage the analyst has the opportunity to save the current pre-processing analysis by selecting the save option under the pull-down 'File' menu as seen at the top left corner of Figure 6.3.3.1. The system, saves the separate training and validation sets, and only includes data associated with the attributes selected by the analyst in Figure 6.3.3.1. To continue to the VPRS analysis, the analyst can choose the continue option under the 'File' pull-down menu.

6.4 Summary

This chapter has introduced and described the developed pre-processing software, elucidating the pre-processing and feature selection stages within the developed software, applied to the FIBR data set described in Chapter 5. The separation of the data into training and validation sets was exposited, and the results of the discretisation of the data set based on the FUSINTER algorithm was presented and described.

A number of graphs were presented, associated with the feature selection methods described in Chapter 4, and implemented within the software, to aid the analyst in their final attribute selection. The results of the feature selection algorithms were compared, most notably, RST_FS appeared too restrictive, and the results of the other three feature selection methods, namely, ReliefFC, ReliefFD and RST_PH1 appeared correlated. The final attribute selection was demonstrated, utilising the results from RelieFD, where a novel tick box selection system implemented within the developed software, enabled the analyst to select which attributes to pass forward into the subsequent VPRS analysis.

The final set of attributes selected within this chapter, are now utilised in the subsequent VPRS vein graph and re-sampling analyses, presented in Chapters 7 and 8, respectively.

Chapter 7

Vein Graph Software Analysis of the Example and FIBR Data Sets

This chapter introduces the developed VPRS vein graph software, based on the vein graph described by Beynon (2002) (discussed in Chapter 2 sections 2.3). Continuing from the pre-processing phase described in the previous chapter, this chapter elucidates a VPRS vein graph analysis of the FIBR data set based on the final attributes selected during the pre-processing. This presentation is a precursor to the larger elucidation of the developed VPRS re-sampling software presented in Chapter 8.

The vein graph analysis software allows the analyst, via a simple point and click interface, to inspect and select the β-reduct which they believe is most appropriate to their analysis. That is, selection of a β-reduct which has a low β threshold value, allowing for a greater level of misclassification, but with a relatively high proportion of the objects given a classification; or selection of a β-reduct which has a high β threshold value, allowing for a greater level of accuracy, but fewer objects given a classification.

Within the software analysis, the rules induced from the selected β-reduct are applied to both the training and the validation sets. The predictive results based on both the training and validation sets are presented separately and broken down further into those objects that are predicted by a rule which matches an object's condition values, and objects that are predicted by the nearest rule method, based on the equation introduced by Słowiński (1992) (discussed in Chapter 3 Equation 3.6.1).

The sections within this chapter are briefly described below:

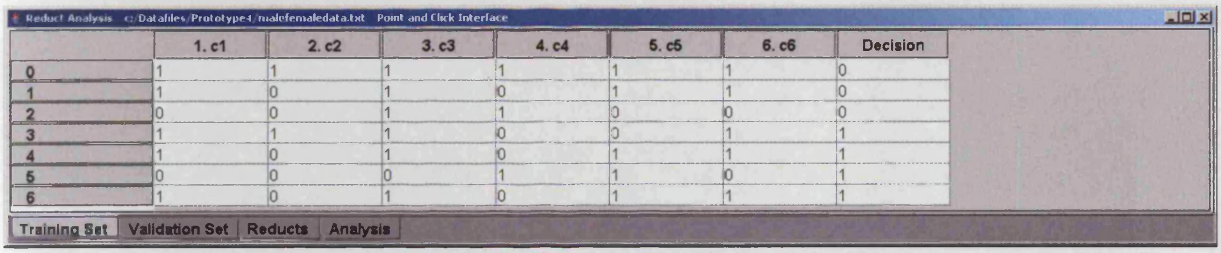
- Section 7.1 **Vein Graph Analysis of the Example Data Set**. This section demonstrates the VPRS vein graph analysis of the simple example data set introduced in Chapter 2 section 2.1. The purpose of this short analysis, is to provide some continuity with Chapter 2, and to give confidence that the developed software can produce results equivalent to those shown in Chapter 2, and are verifiable by hand, by the reader.
- Section 7.2 **Vein Graph Analysis of the FIBR Data Set**. This section presents the VPRS vein graph analysis of the FIBR data set described in Chapter 5. Including, descriptions of the different information panels incorporated within the software, and predictive performances of a selected β-reduct, on both the training and validation sets.
- Section 7.3 **Comparison of FIBR β-reducts**. This section compares the properties of the β-reducts identified within the FIBR data set, in terms of matching or nearest rule classification of objects, and the confusion matrix.
- Section 7.4 **Summary**. Summarises the results from sections 7.2 and 7.3, and draws conclusions which have implications for the following VPRS re-sampling analyses presented in Chapter 8.

7.1 Vein Graph Analysis of the Example Data Set

This section describes the VPRS vein graph software. If the analyst initially selected the vein graph analysis option during the set-up phase (see Chapter 6 section 6.1); on choosing the 'continue' option from the pull-down menu in the pre-processing software (after the analyst is satisfied with the selected attributes), the software proceeds to perform the VPRS vein graph analysis. They are initially presented with a new frame containing a tabbed panel displaying the training set, as shown in Figure 7.1.1.

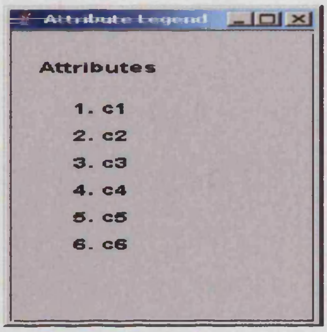
As with the pre-processing software, the tab (tab name) of the panel of information that is currently selected, is highlighted in a light shade of grey. With regards to Figure 7.1.1, the 'Training Set' panel is selected. The tab (tab name) of the panels that are not currently selected (therefore not visible) are indicated by a darker grey.

In addition to the frame shown in Figure 7.1.1, when the analyst continues from the pre-processing phase a second smaller frame also appears as shown in Figure 7.1.2. This smaller frame acts as a quick reference legend, displaying the condition attribute names (discussed in more detail next).


Reduct Analysis c:/Datafiles/Prototype4/malefemaledata.txt Point and Click Interface
Training Set
Validation Set
Reducts
Analysis

	1. c1	2. c2	3. c3	4. c4	5. c5	6. c6	Decision
0	1	1	1	1	1	1	0
1	1	0	1	0	1	1	0
2	0	0	1	1	0	0	0
3	1	1	1	0	0	1	1
4	1	0	1	0	1	1	1
5	0	0	0	1	1	0	1
6	1	0	1	0	1	1	1

Figure 7.1.1: Initial Screenshot of the VPRS Vein Graph Analysis, Displaying the Training Set



Attributes

1. c1
2. c2
3. c3
4. c4
5. c5
6. c6

Figure 7.1.2: Attribute Legend

The training set displayed in Figure 7.1.1, is that of the simple example data set, as discussed in Chapter 2 (Table 2.1.1). The column headers display the condition attribute names, here c_1 to c_6, input from the data file (more practical names can be used). In addition, the attributes are preceded by an index 1 to 6, used for reference purposes. This reference index is most useful during the β-

reduct selection phase, where describing each β-reduct by its associated condition attribute's name, would be awkward for attributes with longer names (shown later). In addition, the floating legend[20] (Figure 7.1.2), allows the analyst to quickly reference the index for the condition attributes' full names. The floating legend is also useful where the condition attribute headings are too large to fit in the column headers of tables within the information panels, such as, that shown in the next section with regards to the analysis of the FIBR data.

Within the tabbed panel displayed in Figure 7.1.1, the analyst can select the 'Validation Set' tab to view the validation set, but in this demonstration, no portion of the data was set aside for the validation set (hence no screenshot is shown here). Also note that, during the pre-processing phase, prior to this VPRS vein graph analysis, the analyst had the option under the discretisation settings to select a setting indicating that the input data was already of a discretised format, and required no further discretisation. Hence, this allowed the simple example data set to be input directly into the analysis, in a pre-discretised format.

By selecting the 'Reducts' tab the analyst is presented with the software implementation of the vein graph, as shown in Figure 7.1.3. The vein graph, has been developed to facilitate an interactive "point and click" interface. The analyst may use the mouse to select a β-reduct which is most appropriate for their analysis requirements. The selected β-reduct is highlighted in dark grey (all β sub-domains are highlighted), within Figure 7.1.3 β-reduct $\{c_4\}$ has been selected.

The position of the mouse pointer is indicated by the red cross-hairs. For the cross-hairs' current position within the β domain (0.5, 1.0], the associated β value is displayed in red (here the β value is 0.6377). The Quality of Classification (QoC) associated with the highlighted β-reduct over the β sub-domain, is displayed in blue, below the β value (here the QoC value is 1.0), and can also be found by inspection of the top line. The top line displays in blue, the differing levels of QoC associated with the full set of condition attributes C, over the associated β sub-domains. Note that

[20] Floating in that, it is not attached to any other frames, and can be minimized or moved to a different position on the screen.

the vein graph displayed in Figure 7.1.3 is equivalent to that presented in Beynon (2001) (shown in Chapter 2 section 2.3 Figure 2.3.1). Figure 7.1.3, shows the selection of β-reduct $\{c_4\}$.

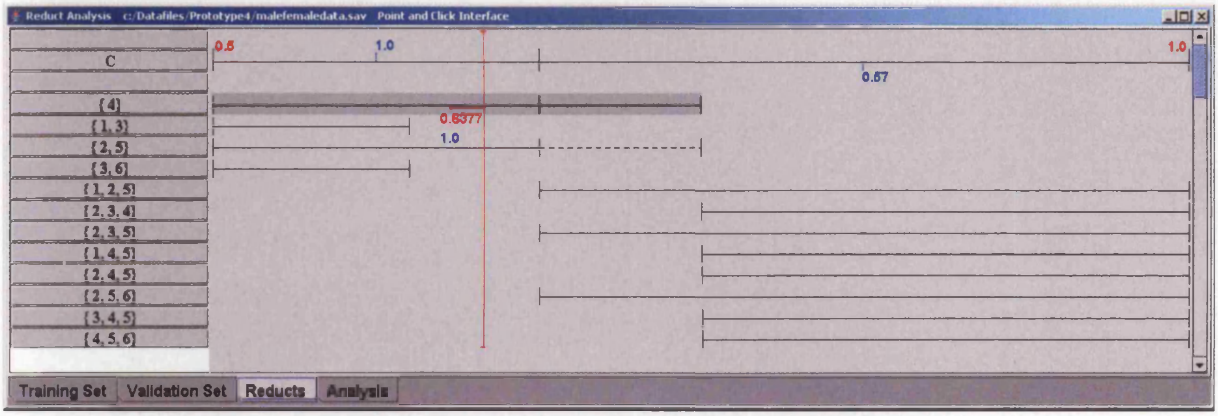
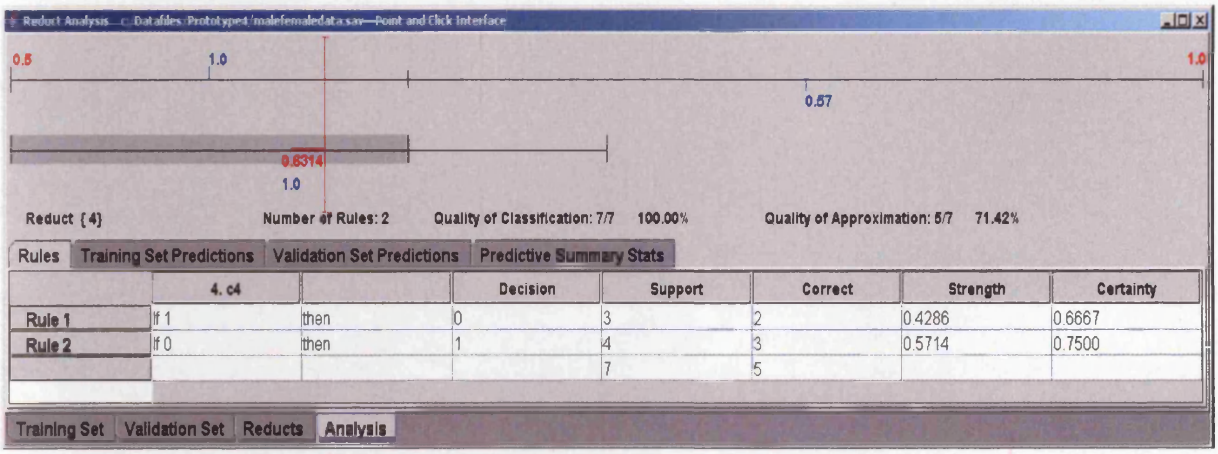


Figure 7.1.3: Vein Graph, Displaying the Selection of β-reduct $\{c_4\}$

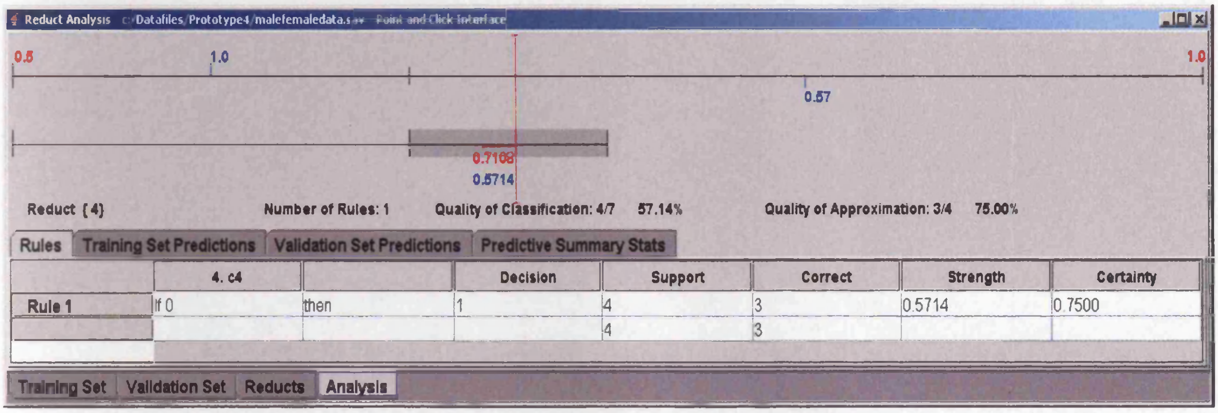
On selecting a β-reduct from the vein graph, the 'Analysis' panel, as shown in Figure 7.1.4, is updated to reflect the analysis based on the rules associated with the selected β-reduct. From here, the analyst has the further option of choosing, where applicable, the β-reduct over a different β sub-domain, using the top portion of the interface (the currently selected β sub-domain is highlighted in dark grey). Selection of a different β sub-domain for the β-reduct $\{c_4\}$ is shown in Figure 7.1.5 (as shown in Chapter 2 section 2.3, β-reducts may be associated with more than one β sub domain).



Reduct {4} Number of Rules: 2 Quality of Classification: 7/7 100.00% Quality of Approximation: 5/7 71.42%

	4. c4		Decision	Support	Correct	Strength	Certainty
Rule 1	If 1	then	0	3	2	0.4286	0.6667
Rule 2	If 0	then	1	4	3	0.5714	0.7500
				7	5		

Figure 7.1.4: The Analysis Panel, Displaying Rules Associated with the β-reduct $\{c_4\}$ Over the β Sub-domain (0.5, 0.66]



Reduct {4} Number of Rules: 1 Quality of Classification: 4/7 57.14% Quality of Approximation: 3/4 75.00%

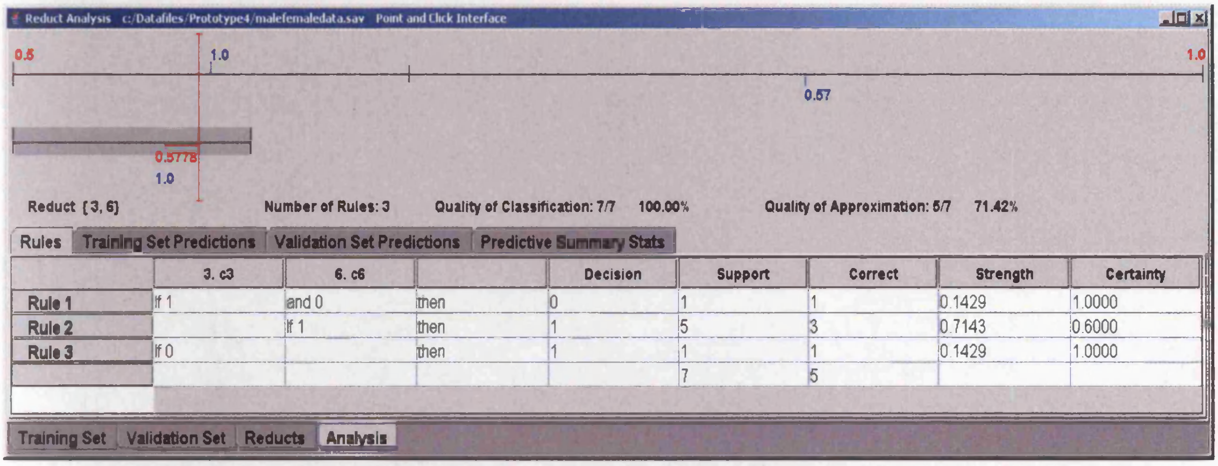
	4. c4		Decision	Support	Correct	Strength	Certainty
Rule 1	If 0	then	1	4	3	0.5714	0.7500
				4	3		

Figure 7.1.5: The Analysis Panel, Displaying the Rule Associated with the β-reduct $\{c_4\}$ Over the β Sub-domain (0.66, 0.75]

Figure 7.1.4 displays the selection of the β-reduct $\{c_4\}$ over the β sub-domain (0.5, 0.66] and its two associated rules (following the rule construction procedure outlined throughout Chapter 2). The condition attributes associated with the β-reduct $\{c_4\}$ are displayed below and to the left of the single β-reduct vein, here {4}, that is condition attribute c_4. Additionally, the number of rules (i.e. two), QoC 7/7 (or 100%) and the QoA 3/4 (or 71.4%) associated with the β-reduct $\{c_4\}$ are also displayed underneath the vein line.

Figure 7.1.5 displays the rules associated with β-reduct $\{c_4\}$, but over the β sub-domain (0.66, 0.75]. It can be clearly seen, that by selecting a higher range of β, decreases the QoC (number of objects within the training set given a classification 4/7) but increases the QoA (number of objects given a classification, that have been classified correctly 3/4). This is indicative of the inverse relationship between QoC and QoA, as highlighted in Beynon (2001).

The rules associated with the β-reduct $\{c_4\}$, are understandably going to be based on a single condition attribute. Hence for example, rule 1 in Figure 7.1.5, would be read as, "*If* $c_4 = 0$ *then* $d_1 = 1$", it is supported by four objects, three of which are classified correctly, leading to a strength of 0.5714 (4/7) and a certainty of 0.75 (3/4). To demonstrate a more interesting set of rules, and to provide some continuity with the example shown at the end of Chapter 2 section 2.2.3, Figure 7.1.6 displays the rules associated with the selected β-reduct $\{c_3, c_6\}$.



Reduct {3, 6} Number of Rules: 3 Quality of Classification: 7/7 100.00% Quality of Approximation: 5/7 71.42%

	3. c3	6. c6		Decision	Support	Correct	Strength	Certainty
Rule 1	If 1	and 0	then	0	1	1	0.1429	1.0000
Rule 2		If 1	then	1	5	3	0.7143	0.6000
Rule 3	If 0		then	1	1	1	0.1429	1.0000
					7	5		

Figure 7.1.6: The Analysis Panel, Displaying Rules Associated with the β-reduct $\{c_3, c_6\}$

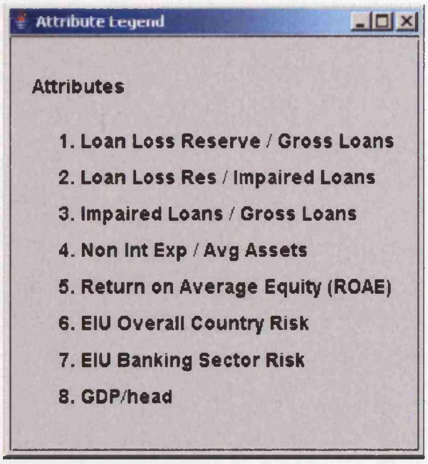
As would be expected, according to the example given in Chapter 2, the β-reduct $\{c_3, c_6\}$ is associated with three rules, a QoC of 7/7 (100%) and a QoA of 5/7 (71.42%). It is interesting to note at this stage, that two of the rules, rule 1 and rule 3, each have a certainty of 1. As a measure of possible future performance (predictive accuracy) of these rules, their certainty values taken on their own would be misleading, as the strength associated with each rule is only 0.1428. Essentially the rules are only categorising one object each. In more complex analyses, rules with a certainty of 1, that only categorise a single object may be considered spurious. The more advanced VPRS re-sampling software assists the analyst in identifying possible spurious rules (as will be shown in Chapter 8), it allows the analyst to select which rules they wish to involve in the final predictive analysis.

There are three further information panels available to the analyst under the 'Analysis' tab within this VPRS vein graphs software; namely 'Training Set Predictions', 'Validation Set Predictions' and 'Predictive Summary Stats', but as no validation set was taken for this example data set, it is more appropriate and productive to discuss these three panels, within the context of the next section, based on the more involved FIBR data set analysis.

7.2 Vein Graph Analysis of FIBR Data Set

This section applies the vein graph analysis to the FIBR data set, based on the attributes selected in the previous chapter. The screenshot of the legend shown in Figure 7.2.1, lists the attributes used within the analysis (identified in Chapter 6), this legend is available to the analyst during the analysis, for referencing the condition attribute names. With regards to the analysis presented within this section, a single β-reduct is selected for investigation and demonstration of the software based results. However, section 7.4 of this chapter will make comparisons over the full set of identified β-reducts.

The vein graph shown in Figure 7.2.2 is associated with the FIBR data set based on the condition attributes displayed in Figure 7.2.1. In Figure 7.2.1, it is shown that there are only six β-reducts identified when using this selected set of attributes.



Attributes

1. Loan Loss Reserve / Gross Loans
2. Loan Loss Res / Impaired Loans
3. Impaired Loans / Gross Loans
4. Non Int Exp / Avg Assets
5. Return on Average Equity (ROAE)
6. EIU Overall Country Risk
7. EIU Banking Sector Risk
8. GDP/head

Figure 7.2.1: Attribute Legend for the Considered FIBR Data Set

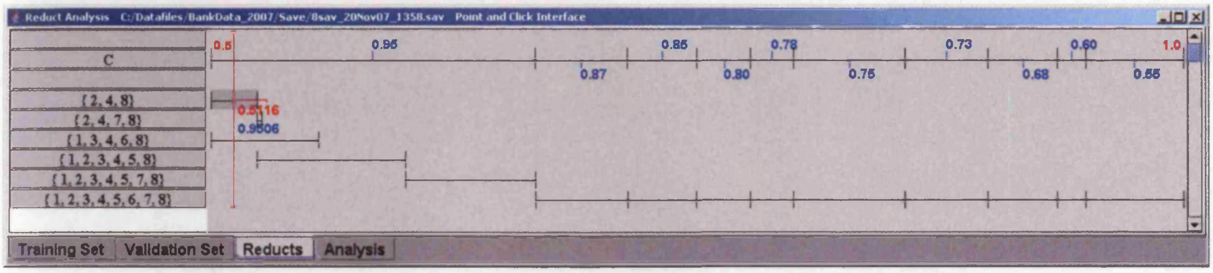


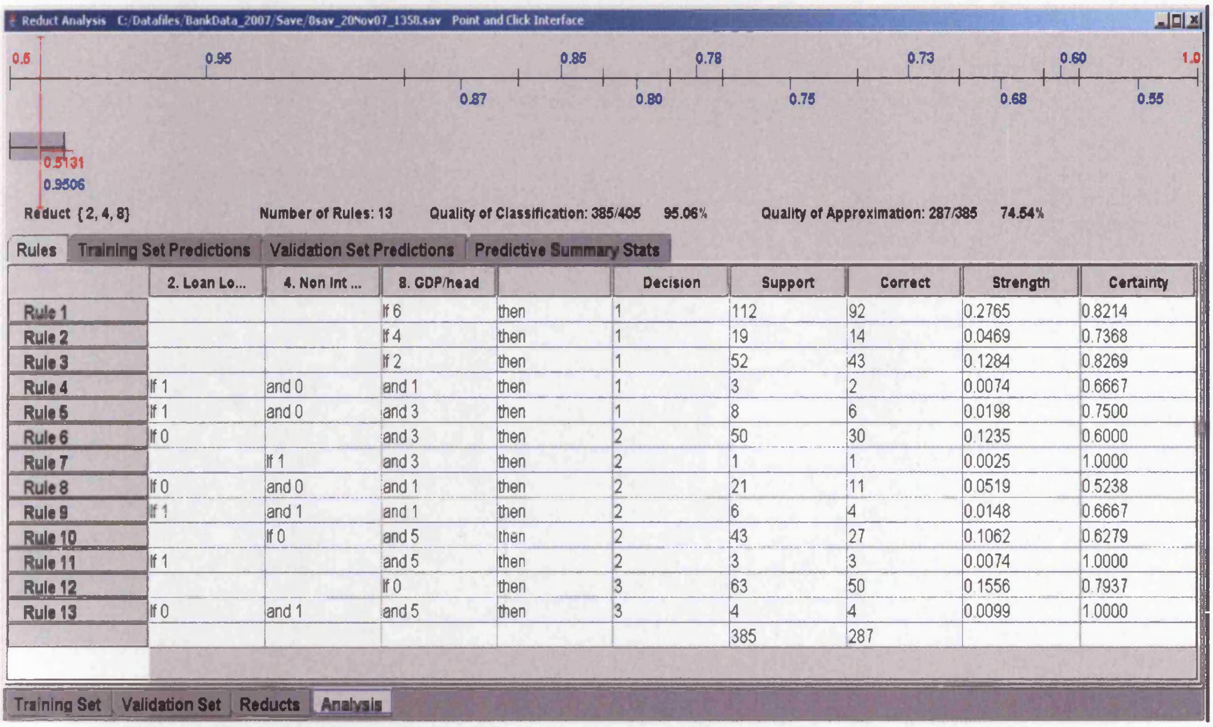
Figure 7.2.2: Vein Graph Associated with the FIBR Data Set

Five out of the six identified β-reducts shown in Figure 7.2.2, have a QoC of 0.95 (lying in the β sub-domain (0.5, 0.67]), including the selected β-reduct $\{c_2, c_4, c_8\}$. Where there is no β-reduct over a certain domain of β, the system defaults to the full set of attributes C, that is, as shown by the sixth β-reduct $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$, with nine associated β sub-domains ranging in (0.6667, 1.0].

Within Figure 7.2.2, the three condition attributes associated with the selected β-reduct $\{c_2, c_4, c_8\}$, belong to three of the five represented categories of the CAMELS model (n.b. the 'M' category was not considered, see Chapter 5 section 5.5). That is, attribute '2' (Impaired Loans / Gross Loans) belongs to the Asset quality category (A), attribute '4' (Non Int Exp / Avg Assests) belongs to the Earnings category (E), and attribute '8' (GDP/head) belongs to the Sensitivity to market risk category (S). Furthermore, it is interesting to see that both attributes '4' and '8' are prevalent in all identified β-reducts, and attribute '2' is associated with five of the six identified β-reducts.

It is hypothesised that, attribute '8' could be acting as a global discriminant (n.b. the FIBR is a rating that is applied to banks globally), which was the original intention behind its inclusion during the attribute/data selection rationale given in Chapter 5 section 5.5. With regards to attributes '2' and '4', these attributes could be acting as the discriminants, that further differentiate banks by refining the discernibility potential within the induced rule sets. As a pertinent but interesting side point, the recently collapsed IndyMac bank within the U.S.A (previously that countries seventh largest mortgage lender), reported on March 31st 2008 that its impaired loans had reached \$1.85 billion, an increase 40.56% from the previous quarter (we assume as a percentage of gross loans) which in turn caused loss of earnings, and hence liquidity and capital problems. The bank was taken into conservatorship by the FDIC on July 11th 2008, two days after Standard & Poor's downgraded its credit risk rating from a 'B' to a 'CCC' and warned its rating would most likely see another downgrade (FDIC, 2008; Reuters, 2008b).

Returning to the selection of β-reduct $\{c_2, c_4, c_8\}$, Figure 7.2.3 displays the 13 rules associated with it. The cross-hairs indicate that, the β value is set to 0.5131, which is within the β-reduct's β sub-domain of (0.5, 0.5238], and is associated with a QoC of 0.9506. Note that, the β-reduct $\{c_2, c_4, c_8\}$ has an upper β threshold value equal to the lowest certainty value belonging to rule 8, shown in Figure 7.2.3.



Reduct {2, 4, 8}   Number of Rules: 13   Quality of Classification: 385/405   95.06%   Quality of Approximation: 287/385   74.54%

	2. Loan Lo...	4. Non Int ...	8. GDP/head		Decision	Support	Correct	Strength	Certainty
Rule 1			If 6	then	1	112	92	0.2765	0.8214
Rule 2			If 4	then	1	19	14	0.0469	0.7368
Rule 3			If 2	then	1	52	43	0.1284	0.8269
Rule 4	If 1	and 0	and 1	then	1	3	2	0.0074	0.6667
Rule 5	If 1	and 0	and 3	then	1	8	6	0.0198	0.7500
Rule 6	If 0		and 3	then	2	50	30	0.1235	0.6000
Rule 7		If 1	and 3	then	2	1	1	0.0025	1.0000
Rule 8	If 0	and 0	and 1	then	2	21	11	0.0519	0.5238
Rule 9	If 1	and 1	and 1	then	2	6	4	0.0148	0.6667
Rule 10		If 0	and 5	then	2	43	27	0.1062	0.6279
Rule 11	If 1		and 5	then	2	3	3	0.0074	1.0000
Rule 12			If 0	then	3	63	50	0.1556	0.7937
Rule 13	If 0	and 1	and 5	then	3	4	4	0.0099	1.0000
						385	287		

Figure 7.2.3: Analysis of the β-reduct $\{c_2, c_4, c_8\}$, which has 13 Associated Rules

Giving some interpretation to the rules shown within Figure 7.2.3, of the 13 rules shown, rule 1 based only on GDP/head is the strongest (rule supported by 112 banks, giving a strength value of 112/405 or 0.2765) and has the second highest certainty value (correctly predicts 92 banks, giving a certainty value of 92/112 or 0.8214). It is interesting to note that almost a third of all the banks are given a classification based on just rule 1, associated with only one condition attribute, namely GDP/Head. Moreover, GDP/head appears to be a factor in all thirteen rules, strengthening the argument that GDP/head could be acting as a “global” discriminant.

There are a number of rules within the rule set, which have relatively high support values but are associated with weak certainties, for example, rules 6 and 8. This is understandable, since the β-reduct $\{c_2, c_4, c_8\}$ is associated with a low β threshold value, which therefore implies that, the analysis is including rules associated with a high degree of misclassification.

Furthermore, there are a number of rules with low support values (therefore low strength), but have high certainty values, for example, rules 7, 11 and 13. These rules tend to be associated with condition classes containing only a few banks (objects[21]), or in the extreme case, such as rule 7, a single bank. Strong conclusions, cannot be drawn on the rules associated with low strengths, and high certainty values. Hence, perhaps with regards to β-reducts associated with low β threshold values, only the more general rules, concomitant with, relatively high support (higher rule strengths) and reasonably high levels of certainty should be considered. This may be particularly true when trying to establish any general trends, relating condition attribute values to the decision classes (here financial variables to FIBR grade).

Here in terms of the FIBR, the general trend suggests that a bank domicile within a country that has a high GDP/head, is more likely to have a relatively high bank rating. Loan Loss Res/Impaired Loans (index 2) and Non Int Exp/Avg Assets (index 4), appear to be important additional factors to a bank's final rating classification (grade) (as considered with regards to Figure 7.2.2). Where Loan Loss Res/Impaired Loans, is associated with a relatively high value (discrete value of 1) and Non Int Exp/Avg Assets is associated with a relatively low value (discrete value of 0), this could imply the difference between a bank being classified with a 'B' (1) rating, or a 'C' (2) rating (see rule 5).

Rules 10, 11 and 13 appear more difficult to interpret. That is, why would banks within countries that have apparently high levels of GDP/head (discrete value 5), not be associated with higher ratings? Rules 11 and 13 could be discounted as their rule strengths are quite weak (0.0074 and 0.0099 respectively), and would give the analyst no confidence in their future performance. Rule 10

[21] As a convention with regards to the FIBR data, objects will be referred to as banks throughout the remainder of this chapter, and Chapter 8.

which has the 5th highest strength (0.1062), appears to suggest that a low value for Non Int Exp / Avg Assets would bring the rating down, but note the weak certainty of rule 10. It has the third lowest certainty value (0.6279), and perhaps on this basis, rule 10 could be discredited. Indeed Non Int Exp / Avg Assets (non interest expenses or overheads to average assets) is a measure of a banks operational costs, hence conventional wisdom would suggest that, in generally a lower value would indicate an efficiently operating bank (Bankscope, 2007).

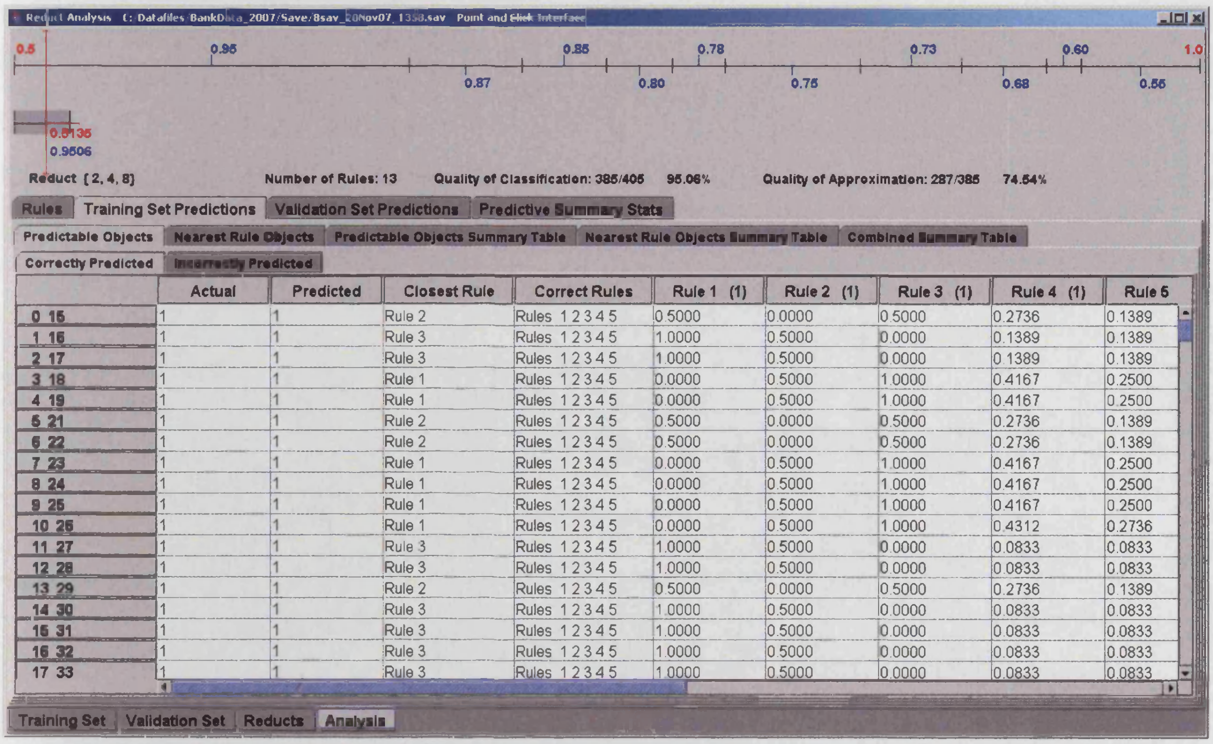
Finally, it is pertinent to note, that the rule set associated with the β-reduct $\{c_2, c_4, c_8\}$ contains no rules capable of classifying banks to the 'A' (0) and 'E' (4) grades. It is hypothesised that the β-reduct $\{c_2, c_4, c_8\}$ does not contain enough condition attributes, hence enough detail, to produce a rule set capable of distinguishing between the three well represented bank ratings 'B' (1), 'C' (2) and 'D' (3), and the under represented bank ratings 'A' (0) and 'E' (4).

In summary, it is reasonable to assume, that the threshold value associated with the selected β-reduct $\{c_2, c_4, c_8\}$ is too low, because it lacked the capability to predict banks belonging to all five rating grades. That is, 'A' and 'E' grade banks would be classified to the other three grades. Hence, it is presumable that selecting a β-reduct whose β sub-domain is associated with a higher β threshold value, may be inclusive of rules that predict the 'A' (0) and 'E' (4) grade banks, and in addition, removes the strong rules that are associated with weak certainty values (such as rule 10 in Figure 7.2.3). Although, it should be noted that β-reducts whose β sub-domains are associated with higher β threshold values, are typically linked with more complex, less interpretable rules (shown later). This issue echoes Breiman's (1996a) statement referenced earlier in Chapter 3 section 3.5, suggesting that what we can gain in accuracy, we may loose in interpretability.

7.3 Prediction of FIBR Data Set

The developed VPRS vein graph software, utilises the rules associated with a selected β-reduct, to predict both the training and the validation sets. These predictive results, are presented through a number of separate results panels, which elucidate a range of aspects associated with the prediction process. This includes, amongst other aspects, informing the analyst to which banks were predicted by matching rules (matching their condition attribute values), and which were predicted using the nearest rule method, as described by Słowiński (1992) (see Chapter 3 section 3.6). This section discusses these aspects in detail, beginning with the predictions made on the training set.

In Figure 7.3.1, the 'Correctly Predicted' panel, displays the banks from the training set, predicted correctly by rules associated with the β-reduct $\{c_2, c_4, c_8\}$, which have matching condition values. The adjacent panel, namely the 'Incorrectly Predicted' panel, contains those banks from the training set, that have been incorrectly predicted by rules, which have matching condition attribute values, shown in Figure 7.3.2 (discussed later). Banks from the training set predicted correctly and incorrectly based on the nearest rule method are described later in this section.



Reduct {2, 4, 8}   Number of Rules: 13   Quality of Classification: 385/405   95.06%   Quality of Approximation: 287/385   74.54%

	Actual	Predicted	Closest Rule	Correct Rules	Rule 1 (1)	Rule 2 (1)	Rule 3 (1)	Rule 4 (1)	Rule 5
0 15	1	1	Rule 2	Rules 1 2 3 4 5	0.5000	0.0000	0.5000	0.2736	0.1389
1 16	1	1	Rule 3	Rules 1 2 3 4 5	1.0000	0.5000	0.0000	0.1389	0.1389
2 17	1	1	Rule 3	Rules 1 2 3 4 5	1.0000	0.5000	0.0000	0.1389	0.1389
3 18	1	1	Rule 1	Rules 1 2 3 4 5	0.0000	0.5000	1.0000	0.4167	0.2500
4 19	1	1	Rule 1	Rules 1 2 3 4 5	0.0000	0.5000	1.0000	0.4167	0.2500
5 21	1	1	Rule 2	Rules 1 2 3 4 5	0.5000	0.0000	0.5000	0.2736	0.1389
6 22	1	1	Rule 2	Rules 1 2 3 4 5	0.5000	0.0000	0.5000	0.2736	0.1389
7 23	1	1	Rule 1	Rules 1 2 3 4 5	0.0000	0.5000	1.0000	0.4167	0.2500
8 24	1	1	Rule 1	Rules 1 2 3 4 5	0.0000	0.5000	1.0000	0.4167	0.2500
9 25	1	1	Rule 1	Rules 1 2 3 4 5	0.0000	0.5000	1.0000	0.4167	0.2500
10 26	1	1	Rule 1	Rules 1 2 3 4 5	0.0000	0.5000	1.0000	0.4312	0.2736
11 27	1	1	Rule 3	Rules 1 2 3 4 5	1.0000	0.5000	0.0000	0.0833	0.0833
12 28	1	1	Rule 3	Rules 1 2 3 4 5	1.0000	0.5000	0.0000	0.0833	0.0833
13 29	1	1	Rule 2	Rules 1 2 3 4 5	0.5000	0.0000	0.5000	0.2736	0.1389
14 30	1	1	Rule 3	Rules 1 2 3 4 5	1.0000	0.5000	0.0000	0.0833	0.0833
15 31	1	1	Rule 3	Rules 1 2 3 4 5	1.0000	0.5000	0.0000	0.0833	0.0833
16 32	1	1	Rule 3	Rules 1 2 3 4 5	1.0000	0.5000	0.0000	0.0833	0.0833
17 33	1	1	Rule 3	Rules 1 2 3 4 5	1.0000	0.5000	0.0000	0.0833	0.0833

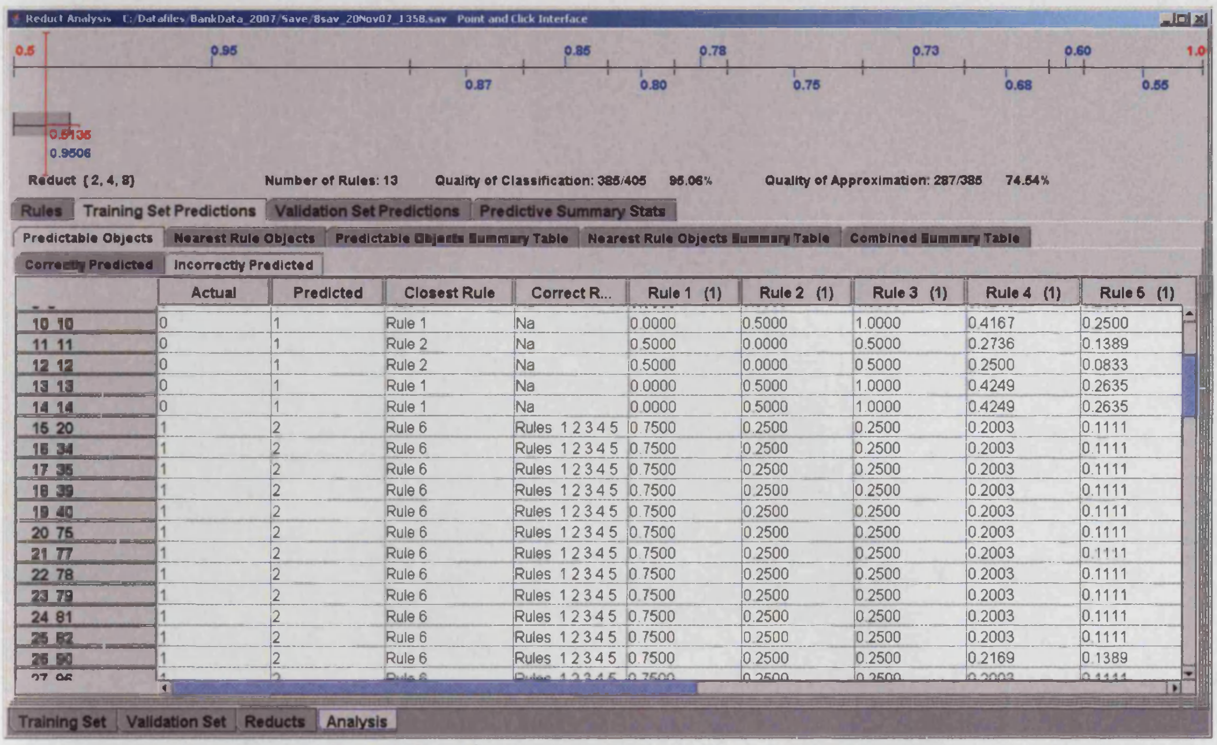
Figure 7.3.1: Information Panel Displaying Banks from the Training Set, Correctly Predicted by Matching Rules from the Rule Set Associated with β-reduct $\{c_2, c_4, c_8\}$

Describing the 'Correctly Predicted' panel presented within Figure 7.3.1, the row headers display, in the left of the row header cell, a simple row index for reference purposes, and to the right, the banks original index value, read in from the data file (not successive values since banks in the validation set are not included). The index value to the right is useful to the analyst, because it allows them to find its original row position either under the 'Training Set' panel (which also indicates the banks original index values read in from the data file), or the original 'Data Set' panel (both of which are still accessible through the pre-processing software, see Chapter 6 section 6.2), and hence this enables the analyst to view the banks condition attribute values.

In Figure 7.3.1 the 'Actual' column displays the decision class values '0' to '4' associated with each bank (values '0' to '4' represent FIBRs 'A' to 'E', respectively). The 'Predicted' column displays the decision class predicted by the matching rule from the rule set (described next). The 'Closest Rule' column displays the closest (nearest) rule based on Słowiński's (1992) distance measure. The 'Correct Rules' column indicates which rules would have correctly classified the bank.

The remaining columns display the distance measure between each rule and the predicted banks. It is notable that, because these banks have been predicted by a rule which exactly matches the banks condition values, then intuitively, one of the rules will have a distance measure of zero. For example, bank 15 in row 0, is predicted correctly by rule 2, which has a distance measure of zero; therefore rule 2 has been recorded as the closest rule. The decision class value (FIBR grade) for each rule is displayed in the brackets to the right of the respective column heading, for example, 'rule 1 (1)' classifies banks to decision class '1'. The table can be scrolled horizontally to view the full set of rules. Where banks are referred to as 'predictable' in the remainder of this dissertation, this is referring to them being predictable by matching rules (not Słowiński's, 1992, nearest rule method).

As stated earlier, the banks from the training set, predicted incorrectly by rules which have matching condition values, are displayed under a separate, but adjacent panel, as shown in Figure 7.3.2.



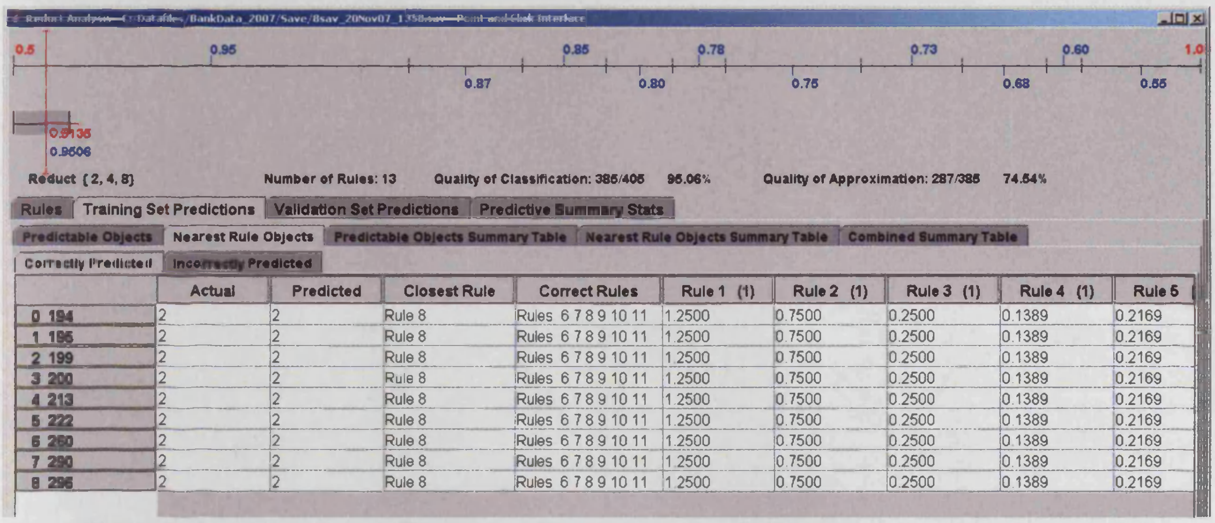
Reduct {2, 4, 8} Number of Rules: 13 Quality of Classification: 385/405 95.06% Quality of Approximation: 287/385 74.54%

	Actual	Predicted	Closest Rule	Correct R...	Rule 1 (1)	Rule 2 (1)	Rule 3 (1)	Rule 4 (1)	Rule 5 (1)
10 10	0	1	Rule 1	Na	0.0000	0.5000	1.0000	0.4167	0.2500
11 11	0	1	Rule 2	Na	0.5000	0.0000	0.5000	0.2736	0.1389
12 12	0	1	Rule 2	Na	0.5000	0.0000	0.5000	0.2500	0.0833
13 13	0	1	Rule 1	Na	0.0000	0.5000	1.0000	0.4249	0.2635
14 14	0	1	Rule 1	Na	0.0000	0.5000	1.0000	0.4249	0.2635
15 20	1	2	Rule 6	Rules 1 2 3 4 5	0.7500	0.2500	0.2500	0.2003	0.1111
16 34	1	2	Rule 6	Rules 1 2 3 4 5	0.7500	0.2500	0.2500	0.2003	0.1111
17 35	1	2	Rule 6	Rules 1 2 3 4 5	0.7500	0.2500	0.2500	0.2003	0.1111
18 39	1	2	Rule 6	Rules 1 2 3 4 5	0.7500	0.2500	0.2500	0.2003	0.1111
19 40	1	2	Rule 6	Rules 1 2 3 4 5	0.7500	0.2500	0.2500	0.2003	0.1111
20 75	1	2	Rule 6	Rules 1 2 3 4 5	0.7500	0.2500	0.2500	0.2003	0.1111
21 77	1	2	Rule 6	Rules 1 2 3 4 5	0.7500	0.2500	0.2500	0.2003	0.1111
22 78	1	2	Rule 6	Rules 1 2 3 4 5	0.7500	0.2500	0.2500	0.2003	0.1111
23 79	1	2	Rule 6	Rules 1 2 3 4 5	0.7500	0.2500	0.2500	0.2003	0.1111
24 81	1	2	Rule 6	Rules 1 2 3 4 5	0.7500	0.2500	0.2500	0.2003	0.1111
25 82	1	2	Rule 6	Rules 1 2 3 4 5	0.7500	0.2500	0.2500	0.2003	0.1111
26 90	1	2	Rule 6	Rules 1 2 3 4 5	0.7500	0.2500	0.2500	0.2169	0.1389
27 96	1	2	Rule 6	Rules 1 2 3 4 5	0.7500	0.2500	0.2500	0.2003	0.1111

Figure 7.3.2: Information Panel, Displaying Banks from the Training Set, Incorrectly Predicted by Matching Rules from the Rule Set Associated with β-reduct $\{c_2, c_4, c_8\}$

The format of the information displayed in Figure 7.3.2, is identical to that described previously for Figure 7.3.1. Note here though, the 'Correct Rule' column displays the value 'NA' (None Available), in regards to the banks whose actual decision class values were '0' ('A' grade banks), as there are no rules capable of predicting these banks correctly. Intuitively, this is a consequence of the earlier realisation that, the rule set contained no rules capable of classifying either the decision classes '0' ('A' grade banks) or '4' ('E' grade banks). The analysis presented in Figure 7.3.2, may be useful to an analyst who is interested in seeing the nearness of the rules other than the closest rule, and may give them further incite into the rating mechanism.

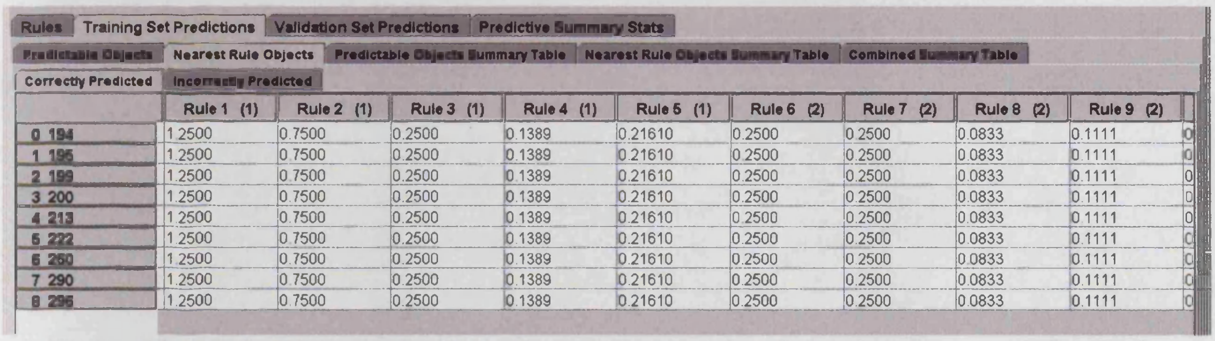
Moving on to discuss those banks predicted by the nearest rule method. Figure 7.3.3 displays those banks which have been correctly classified by the nearest rule method. Figure 7.3.4 presents the same information to that in Figure 7.3.3, but scrolled to the right so that more of the rule distance measures are visible.


Reduct Analysis — C:\Datafiles\BankData_2007\Save\8sav_20Nov07_1358sav — Point-and-Click Interface
0.5
0.95
0.85
0.78
0.73
0.60
1.0
0.87
0.80
0.75
0.68
0.55
0.9136
0.9506

Reduct {2, 4, 8} Number of Rules: 13 Quality of Classification: 385/405 95.06% Quality of Approximation: 287/385 74.54%

	Actual	Predicted	Closest Rule	Correct Rules	Rule 1 (1)	Rule 2 (1)	Rule 3 (1)	Rule 4 (1)	Rule 5
0 194	2	2	Rule 8	Rules 6 7 8 9 10 11	1.2500	0.7500	0.2500	0.1389	0.2169
1 195	2	2	Rule 8	Rules 6 7 8 9 10 11	1.2500	0.7500	0.2500	0.1389	0.2169
2 199	2	2	Rule 8	Rules 6 7 8 9 10 11	1.2500	0.7500	0.2500	0.1389	0.2169
3 200	2	2	Rule 8	Rules 6 7 8 9 10 11	1.2500	0.7500	0.2500	0.1389	0.2169
4 213	2	2	Rule 8	Rules 6 7 8 9 10 11	1.2500	0.7500	0.2500	0.1389	0.2169
5 222	2	2	Rule 8	Rules 6 7 8 9 10 11	1.2500	0.7500	0.2500	0.1389	0.2169
6 260	2	2	Rule 8	Rules 6 7 8 9 10 11	1.2500	0.7500	0.2500	0.1389	0.2169
7 290	2	2	Rule 8	Rules 6 7 8 9 10 11	1.2500	0.7500	0.2500	0.1389	0.2169
8 296	2	2	Rule 8	Rules 6 7 8 9 10 11	1.2500	0.7500	0.2500	0.1389	0.2169

Figure 7.3.3: Information Panel, Displaying Banks from the Training Set, Correctly Predicted by the Nearest Rule method, from the Rule Set Associated with β-reduct $\{c_2, c_4, c_8\}$

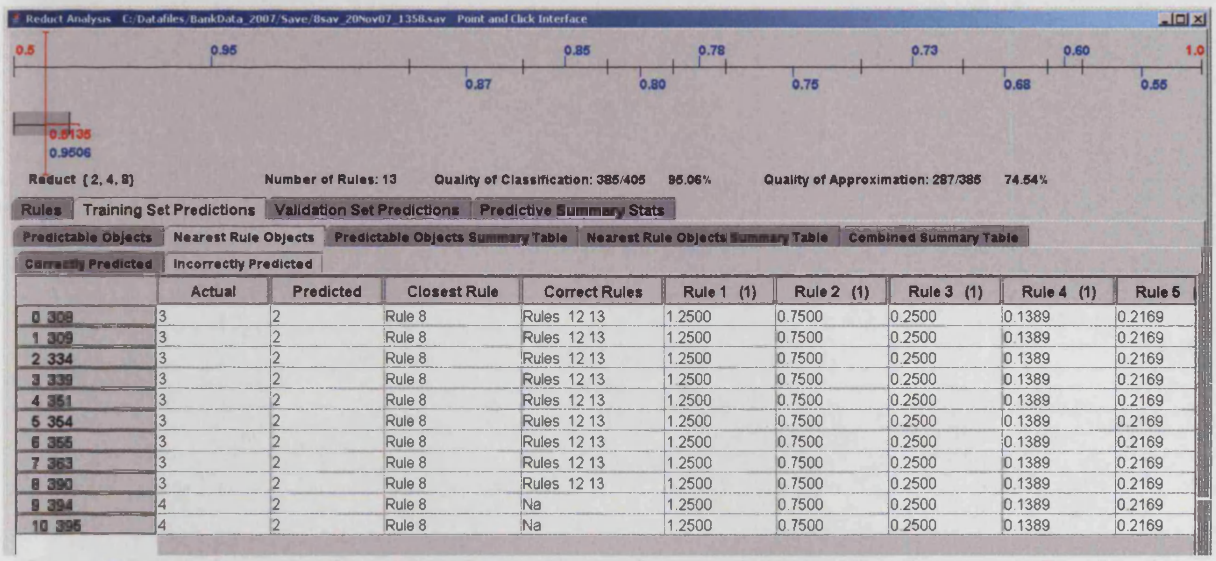


	Rule 1 (1)	Rule 2 (1)	Rule 3 (1)	Rule 4 (1)	Rule 5 (1)	Rule 6 (2)	Rule 7 (2)	Rule 8 (2)	Rule 9 (2)	
0 194	1.2500	0.7500	0.2500	0.1389	0.21610	0.2500	0.2500	0.0833	0.1111	0
1 195	1.2500	0.7500	0.2500	0.1389	0.21610	0.2500	0.2500	0.0833	0.1111	0
2 199	1.2500	0.7500	0.2500	0.1389	0.21610	0.2500	0.2500	0.0833	0.1111	0
3 200	1.2500	0.7500	0.2500	0.1389	0.21610	0.2500	0.2500	0.0833	0.1111	0
4 213	1.2500	0.7500	0.2500	0.1389	0.21610	0.2500	0.2500	0.0833	0.1111	0
5 222	1.2500	0.7500	0.2500	0.1389	0.21610	0.2500	0.2500	0.0833	0.1111	0
6 260	1.2500	0.7500	0.2500	0.1389	0.21610	0.2500	0.2500	0.0833	0.1111	0
7 290	1.2500	0.7500	0.2500	0.1389	0.21610	0.2500	0.2500	0.0833	0.1111	0
8 296	1.2500	0.7500	0.2500	0.1389	0.21610	0.2500	0.2500	0.0833	0.1111	0

Figure 7.3.4: Rule Distances for Rules 1 to 9, Associated with β-reduct $\{c_2, c_4, c_8\}$, for Banks Classified Correctly by the Nearest Rule Method

The format of the table displayed in Figure 7.3.3 is similar to those described previously in Figures 7.3.1 and 7.3.2. Note here though, because the closest (nearest) rule is not a matching rule, the distance of the closest rule is above zero. For example, bank 194 in row 0 of Figure 7.3.3, is predicted by rule 8; it can be seen from Figure 7.3.4, that the distance of rule 8 to the condition attribute values associated with bank 194 is 0.0833. Where there is more than one rule with equal lowest distance measure to a bank, the rule strength and certainty measures are used to discern between each candidate rule (as per the description in Chapter 3 section 3.6).

Figure 7.3.5, the 'Incorrectly Predicted' panel, displays those banks predicted by the nearest rule method, but have been incorrectly predicted.



Reduct {2, 4, 8} Number of Rules: 13 Quality of Classification: 385/405 95.06% Quality of Approximation: 287/385 74.54%

	Actual	Predicted	Closest Rule	Correct Rules	Rule 1 (1)	Rule 2 (1)	Rule 3 (1)	Rule 4 (1)	Rule 5
0 308	3	2	Rule 8	Rules 12 13	1.2500	0.7500	0.2500	0.1389	0.2169
1 309	3	2	Rule 8	Rules 12 13	1.2500	0.7500	0.2500	0.1389	0.2169
2 334	3	2	Rule 8	Rules 12 13	1.2500	0.7500	0.2500	0.1389	0.2169
3 339	3	2	Rule 8	Rules 12 13	1.2500	0.7500	0.2500	0.1389	0.2169
4 351	3	2	Rule 8	Rules 12 13	1.2500	0.7500	0.2500	0.1389	0.2169
5 354	3	2	Rule 8	Rules 12 13	1.2500	0.7500	0.2500	0.1389	0.2169
6 355	3	2	Rule 8	Rules 12 13	1.2500	0.7500	0.2500	0.1389	0.2169
7 363	3	2	Rule 8	Rules 12 13	1.2500	0.7500	0.2500	0.1389	0.2169
8 390	3	2	Rule 8	Rules 12 13	1.2500	0.7500	0.2500	0.1389	0.2169
9 394	4	2	Rule 8	Na	1.2500	0.7500	0.2500	0.1389	0.2169
10 395	4	2	Rule 8	Na	1.2500	0.7500	0.2500	0.1389	0.2169

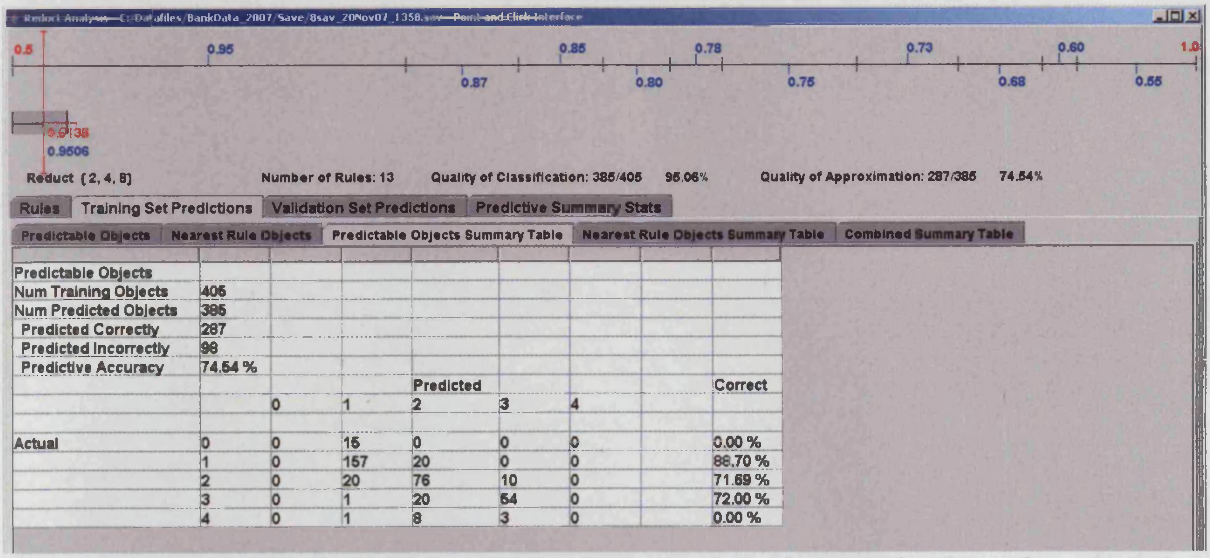
Figure 7.3.5: Information Panel Displaying Banks from the Training Set, Incorrectly Predicted by the Nearest Rules from the Rule Set Associated with β-reduct $\{c_2, c_4, c_8\}$

In Figure 7.3.5, again, the 'Correct Rule' column displays the value 'NA' with regards to those banks whose actual decision class values are '4' ('E' grade banks), which as realised previously, is a consequence of the respective rule set containing no rules capable of predicting decision class '4' ('E' grade bank).

The results presented in Figure 7.3.3 to Figure 7.3.5 shows that, only a small proportion of the training set is actually predicted using the nearest rule method (only nine banks in Figure 7.3.3, and 11 banks in Figure 7.3.5). Here, the limited number of banks predicted by the nearest rule method, is an effect of selecting a β-reduct associated with a low β threshold value, which has, consequently, allowed for a large portion of the banks in the training set, to be given a classification, be it correctly or incorrectly by a matching rule.

Within the developed VPRS software, exact values on the number of banks classified, and the predictive accuracies of the rule set on the training set, are presented through three additional information tables, within three selectable panels. Namely, the 'Predictable Objects Summary Table', the 'Nearest Rule Objects Summary Table' and the 'Combined Summary Table', each of which are described next.

Firstly, the 'Predictable Objects Summary Table', shown in Figure 7.3.6, displays summary information based on those banks predicted by rules with matching values (both correctly and incorrectly).


Reduct Analysis — C:\Datafiles\BankData_2007\Save\8sav_20Nov07_1358.sav — Point-and-Click-Interface
0.5
0.95
0.85
0.78
0.73
0.60
1.0
0.87
0.80
0.75
0.68
0.55
0.9138
0.9506
Rules
Training Set Predictions
Validation Set Predictions
Predictive Summary Stats
Predictable Objects
Nearest Rule Objects
Predictable Objects Summary Table
Nearest Rule Objects Summary Table
Combined Summary Table

Reduct {2, 4, 8}   Number of Rules: 13   Quality of Classification: 385/405   95.06%   Quality of Approximation: 287/385   74.54%

Predictable Objects								
Num Training Objects	405							
Num Predicted Objects	385							
Predicted Correctly	287							
Predicted Incorrectly	98							
Predictive Accuracy	74.54 %							
				Predicted				Correct
		0	1	2	3	4		
Actual	0	0	15	0	0	0		0.00 %
	1	0	157	20	0	0		88.70 %
	2	0	20	76	10	0		71.69 %
	3	0	1	20	54	0		72.00 %
	4	0	1	8	3	0		0.00 %

Figure 7.3.6: Summary Table for Training Set Banks Predicted by Matching Rules from the Rule Set Associated with β-reduct $\{c_2, c_4, c_8\}$

Within Figure 7.3.6, the table shows the number of banks in the training set (405); the number of banks classified by matching rules (385), and of those banks, the number of them predicted correctly (287) and incorrectly (98). In addition, it displays as a percentage, the predictive accuracy of those 385 banks that have been predicted correctly, that is 74.54% (287/305). This 74.54% value, is equivalent to the displayed QoA (as a percentage), displayed below the "vein line" on the far right of the screenshot. This is understandable, because the predictive accuracy, is based on applying the rules to the data they were constructed upon, namely the training set. Moreover, the predictive accuracy could be considered the apparent predictive accuracy on the banks that are predictable (see Chapter 3 section 3.1). This is an important point, because the QoA value is known prior to applying the rule set to the training set. Hence, with regards to the VPRS framework, the apparent predictive accuracy is known without the need for further calculations.

The lower portion of the table in Figure 7.3.6, displays information cross referencing the 'Actual' decision classes (rating grades) of the banks, and the 'Predicted' decision classes of the banks, through the implementation of the confusion matrix (shown in Chapter 3 subsection 3.2.1). To describe further, 15 'A' (0) grade banks have been incorrectly classified as 'B' (1) grade banks; 157 'B' (1) grade banks have been correctly classified, but 20 have been incorrectly classified as 'C' (2) grade banks, and so forth.

The final column of this confusion matrix displays the predictive accuracies associated with each individual decision class. For example, 88.7% of the 'B' (1) grade banks have been classified correctly. These predictive accuracies play a key role in determining the performance (predictive capability) of the classifier (the β-reduct and its associated rules), because the predictive accuracy of 74.54% based on all the decision classes,[22] does not reflect the fact that the predictive accuracies over the five individual decisions classes, predictable by the rule set, varies between 0.0% and 88.70%.

Displaying the range of predictive accuracies over all the decision classes (the confusion matrix), allows for a level of transparency which, may be lacking from, or not reported within many studies (Poon et al., 1999; Oelericha and Poddig, 2006). Clearly, quoting the single overall predictive accuracy could be misleading and indeed with reference to the analysis based on the validation set (shown later), this variance between the predictive accuracies of the individual decision classes is magnified. Moreover, when dealing with an imbalanced data set, such as the FIBR data, there is a likelihood that, the well represented decision classes will be associated with a relatively high predictive accuracy, and the under represented decision classes will be associated with a low predictive accuracy, but based on the overall predictive accuracy, the dominance of the larger well represented decision classes would mask the disparity.

A final pertinent observation, with regards to the distribution of the predicted banks within the confusion matrix shown in Figure 7.3.6, is that, 89.8% (88/98) of the incorrectly predicted banks are at most, only one decision class away from their actual decision class. This is also reflected in the later validation set analysis (see Figure 7.3.9). These banks could be border line cases, and by

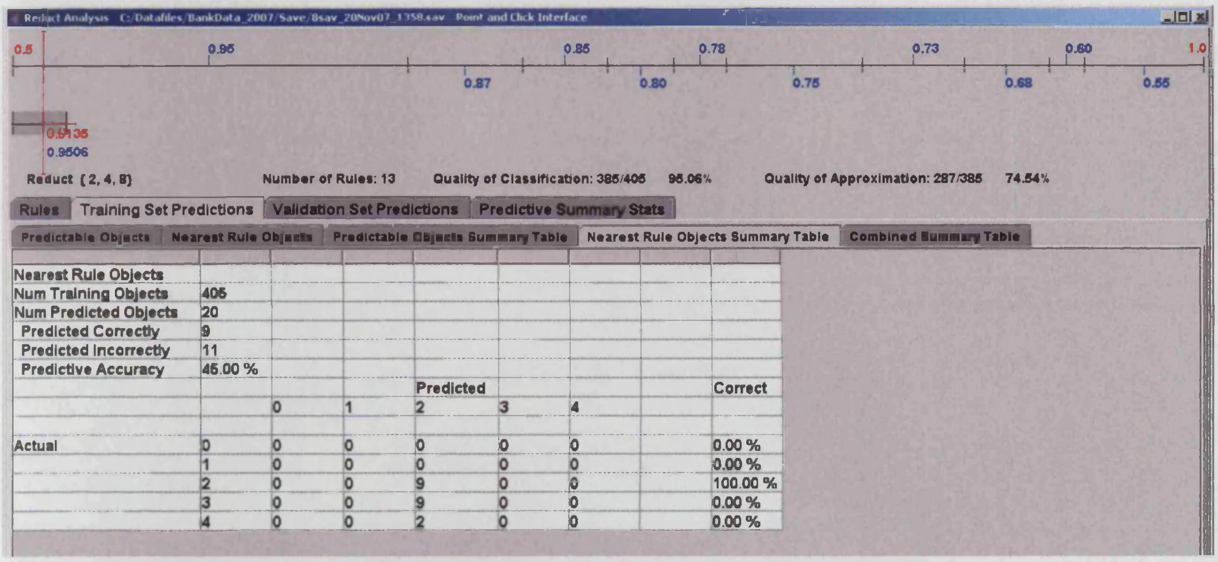
[22] Note the value of 74.54% is not the average of the individual predictive accuracies associated with each decision class.

selecting a β-reduct associated with a higher β threshold value (which typically implies the β-reduct is associated with more condition attributes, and hence, a larger more complex, potentially more accurate rule set), may allow for more banks to be correctly classified. Choosing a β-reduct with a higher β threshold value, appears to "thin out" these border line cases, particularly with regards to the training set. This trend though, is not reflected in the later validation set analysis. Moreover, it is shown in the next section, that by choosing a β-reduct too far to the right of the vein graph, that is, β-reducts whose β sub-domains are associated with high β thresholds, can lead to overfitting of the classifier (the β-reduct and associated rules). Additionally, they are associated with high, but misleading predictive accuracies based on the training set, as there is no noticeable improvement on the more important predictive accuracies based on the validation set (more important since the predictive accuracies based on the validation set are less biased, and the analyst can take more confidence from them, see Chapter 3 section 3.1). In fact, it can impair the predictive performance on the validation set. These issues are presented, in more depth, within the next section.

With regards to bank rating prediction, and an analyst's investment strategy. If it is the case that the majority of misclassified banks are typically predicated only one rating grade away from the true grade, then there may be an argument that, the losses attributed to the incorrectly predicted banks, are in some sense minimalised.

Considering next, the 'Nearest Objects Summary Table' panel as shown in Figure 7.3.7, it displays summary information based on those banks predicted, both correctly and incorrectly by the nearest rule method.

The format of the table within Figure 7.3.7 is identical to that described in Figure 7.3.6. Here, there is less information displayed within the confusion matrix portion of the table, due to there being fewer banks predicted by the nearest rule method. The most interesting point to note here, is, the poor predictive performance, with only nine out of the 20 banks (45.0%) classified correctly (note they are all predicted 'B' grade banks). From experience, based on this, and other testing analyses undertaken during the software development, the poor predictive performance associated with the nearest rule method seems characteristic.

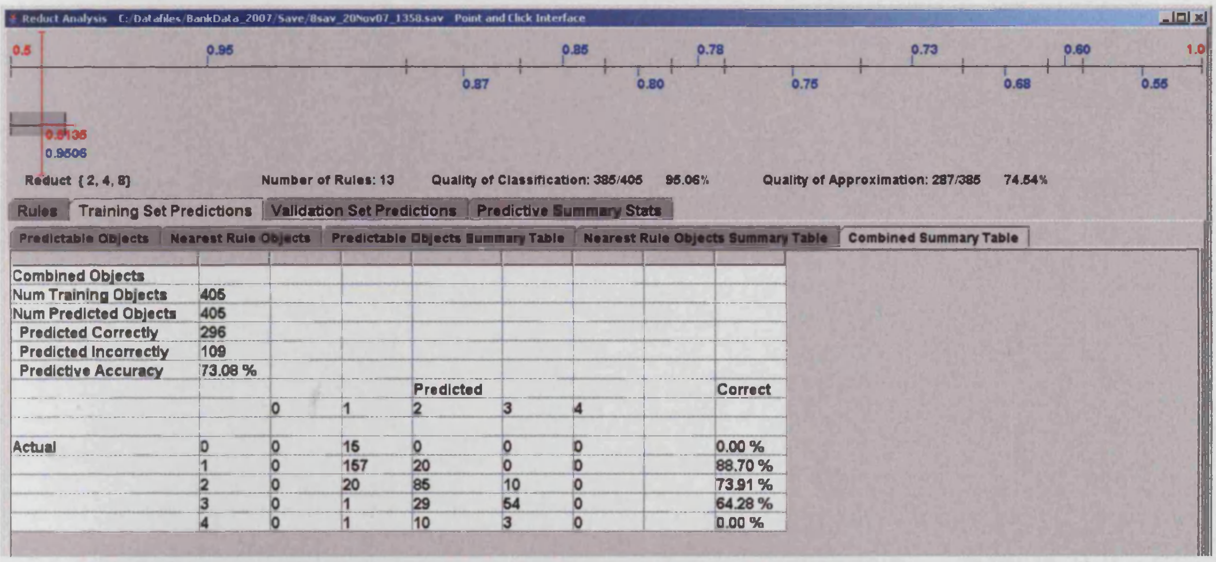

Reduct Analysis C:/Datafiles/BankData_2007/Save/8sav_20Nov07_1358.sav Point and Click Interface
0.5
0.95
0.85
0.78
0.73
0.60
1.0
0.87
0.80
0.75
0.68
0.55
0.9136
0.9506
Rules
Training Set Predictions
Validation Set Predictions
Predictive Summary Stats
Predictable Objects
Nearest Rule Objects
Predictable Objects Summary Table
Nearest Rule Objects Summary Table
Combined Summary Table

Reduct {2, 4, 8} Number of Rules: 13 Quality of Classification: 385/405 95.06% Quality of Approximation: 287/385 74.54%

Nearest Rule Objects								
Num Training Objects	405							
Num Predicted Objects	20							
Predicted Correctly	9							
Predicted Incorrectly	11							
Predictive Accuracy	45.00 %							
				Predicted				Correct
		0	1	2	3	4		
Actual	0	0	0	0	0	0		0.00 %
	1	0	0	0	0	0		0.00 %
	2	0	0	9	0	0		100.00 %
	3	0	0	9	0	0		0.00 %
	4	0	0	2	0	0		0.00 %

Figure 7.3.7: Summary Table for Training Set Banks Predicted by Nearest Rules from the Rule Set Associated with β-reduct $\{c_2, c_4, c_8\}$

With final reference to the training set predictions, the 'Combined Summary Table' panel, shown in Figure 7.3.8, displays summary information based on all the banks within the training set, that is, on both those groups of banks predicted correctly and incorrectly, by matching rules and by the nearest rule method, respectively.


Reduct Analysis C:/Datafiles/BankData_2007/Save/8sav_20Nov07_1358.sav Point and Click Interface
0.5
0.95
0.85
0.78
0.73
0.60
1.0
0.87
0.80
0.75
0.68
0.55
0.9136
0.9506
Rules
Training Set Predictions
Validation Set Predictions
Predictive Summary Stats
Predictable Objects
Nearest Rule Objects
Predictable Objects Summary Table
Nearest Rule Objects Summary Table
Combined Summary Table

Reduct {2, 4, 8} Number of Rules: 13 Quality of Classification: 385/405 95.06% Quality of Approximation: 287/385 74.54%

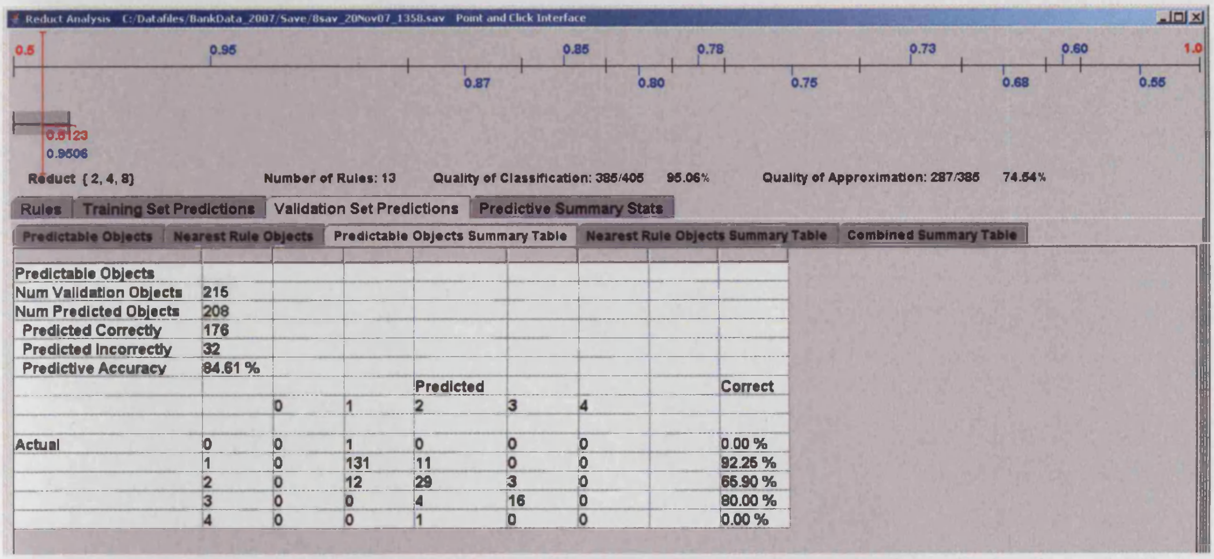
Combined Objects								
Num Training Objects	405							
Num Predicted Objects	405							
Predicted Correctly	296							
Predicted Incorrectly	109							
Predictive Accuracy	73.08 %							
				Predicted				Correct
		0	1	2	3	4		
Actual	0	0	15	0	0	0		0.00 %
	1	0	157	20	0	0		88.70 %
	2	0	20	85	10	0		73.91 %
	3	0	1	29	54	0		64.28 %
	4	0	1	10	3	0		0.00 %

Figure 7.3.8: Summary Table for Training Set Banks Predicted by Matching and Nearest Rules from the Rule Set Associated with β-reduct $\{c_2, c_4, c_8\}$

The purpose of the information displayed in Figure 7.3.8 is to allow the analyst to compare the results of this VPRS analysis with other classifier methods, such as regression analysis. A major advantage the developed VPRS software has to offer over other classifier methods, is the ability to distinguish between, and report separately on, those banks that the rule set can and cannot predict (i.e. predicted by a matching rule and predicted by nearest rule, respectively).

Moreover, in the knowledge that the nearest rule method performs poorly on those banks the rule set does not have a matching rule for, it is likely that the analyst would rather discount those banks as unpredictable, and concentrate on those banks which the software indicates are predictable. To bring this notion back to the FIBR data, and the field of financial investments, this allows the analyst to "cherry pick" and invest in those banks that the VPRS systems indicates are predictable. Furthermore, having knowledge on the strength and certainty of a rule used to predict a bank, the analyst may steer away from predictions made by weaker or less certain rules. This of course, depends on the amount of risk the analyst is willing to take and their investment strategy, for example, invest small amounts on high risk predictions, on a large scale, or large amounts on low risk predictions, on a relatively smaller scale (hedging). In addition, the level of interaction the analyst wishes to have with the process, may also be a factor, as many analysts prefer to just monitor a fully automated trading system (Harnett and Young, 2004, 2007; Hull, 2006).

As the above statement suggests, the predictive accuracies based only on the banks predicted by matching rules, are, especially here, more important than the predictive accuracies based on the nearest rule method or all predictions on the whole training set (i.e. all banks predicted by matching and nearest rules). This is equally true of the validation set, and as such, only the 'Predictable Objects Summary Table' for the validation set is presented here, and shown in Figure 7.3.9. The other panels (not shown here) relating to the validation set take the same format as those described previously in this subsection.


Reduct Analysis C:/Datafiles/BankData_2007/Save/8sav_20Nov07_1358.sav Point and Click Interface
0.5
0.95
0.85
0.78
0.73
0.60
1.0
0.87
0.80
0.75
0.68
0.55
0.8123
0.9506
Rules
Training Set Predictions
Validation Set Predictions
Predictive Summary Stats
Predictable Objects
Nearest Rule Objects
Predictable Objects Summary Table
Nearest Rule Objects Summary Table
Combined Summary Table

Reduct {2, 4, 8}   Number of Rules: 13   Quality of Classification: 385/405   95.06%   Quality of Approximation: 287/385   74.54%

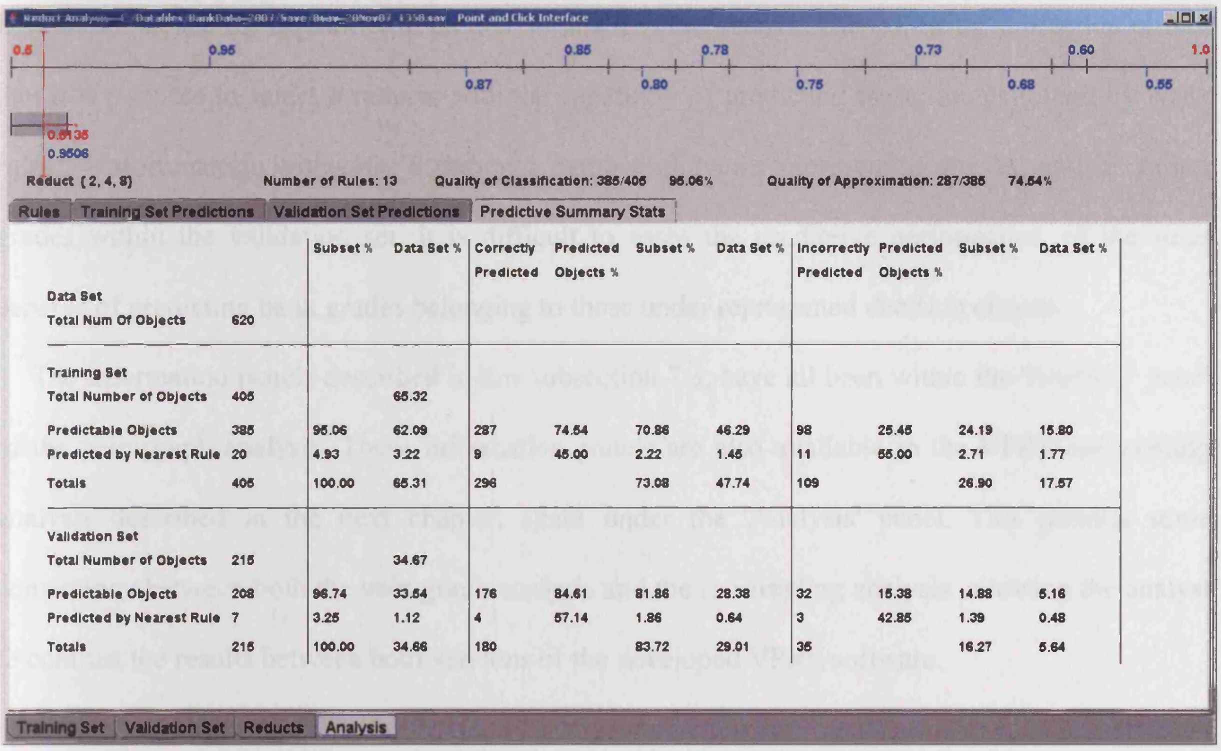
Predictable Objects								
Num Validation Objects	215							
Num Predicted Objects	208							
Predicted Correctly	176							
Predicted Incorrectly	32							
Predictive Accuracy	84.61 %							
				Predicted				Correct
		0	1	2	3	4		
Actual	0	0	1	0	0	0		0.00 %
	1	0	131	11	0	0		92.25 %
	2	0	12	29	3	0		65.90 %
	3	0	0	4	16	0		80.00 %
	4	0	0	1	0	0		0.00 %

Figure 7.3.9: Summary Table for Validation Set Banks Predicted by Matching Rules from the Rule Set Associated with β-reduct $\{c_2, c_4, c_8\}$

Surprisingly, for the validation set, the overall predictive accuracy of 84.61% shown in Figure 7.3.9 is higher than the predictive accuracy of 74.54% reported for the training set shown in Figure 7.3.6. This is surprising, because, the theory outlined in Chapter 3 suggested that, predictive accuracies based on the training set, would typically be over-optimistic (expected to be higher than the validation set). However, it will be seen in the following section that, the predictive accuracies relating to β-reducts associated with higher β threshold values, do tend to have, higher, over-optimistic, predictive accuracies based on the training set.

Comparing the individual predictive accuracies over the individual decision classes, between both the training and validation sets (Figures 7.3.6 and 7.3.9 respectively), based on the rule set associated with β-reduct $\{c_2, c_4, c_8\}$; the accuracies based on the validation set are higher than the accuracies based on the training set for the 'B' (1) and 'D' (3) grade banks, but lower for the 'C' (2) grade banks. However, too much confidence should not be placed on the accuracies for the 'C' (2) and 'D' (3) grade banks, for either the training or validation sets, as these accuracies are based, on a small number of banks. That is, for example, correctly or incorrectly classifying a single bank that should be a 'D' grade bank, can change the predictive accuracy by 5%. Encouragingly though, both the overall predictive accuracies and the predictive accuracies based on the individual decision classes are relatively respectable, based on the training and validation sets, and comparable with results of similar studies (Oelericha and Poddig, 2006).

Finally, as it can be observed throughout this subsection, based on the one selected β-reduct, there is a plethora of information (displayed on a number of panels), relating both to the training and validation sets for the analyst to digest. Hence, the final panel of information to be described here, and the remaining panel to be described within the vein graph analysis, summarises the more pertinent information presented within the other panels, such as, the sample sizes and overall predictive accuracies for the training and validation sets, shown here in Figure 7.3.10.



Reduct {2, 4, 8} Number of Rules: 13 Quality of Classification: 385/405 95.06% Quality of Approximation: 287/385 74.54%

		Subset %	Data Set %	Correctly Predicted	Predicted Objects %	Subset %	Data Set %	Incorrectly Predicted	Predicted Objects %	Subset %	Data Set %
Data Set											
Total Num Of Objects	620										
Training Set											
Total Number of Objects	405		65.32								
Predictable Objects	385	95.06	62.09	287	74.54	70.86	46.29	98	25.45	24.19	15.80
Predicted by Nearest Rule	20	4.93	3.22	9	45.00	2.22	1.45	11	55.00	2.71	1.77
Totals	405	100.00	65.31	296		73.08	47.74	109		26.90	17.57
Validation Set											
Total Number of Objects	215		34.67								
Predictable Objects	208	96.74	33.54	176	84.61	81.86	28.38	32	15.38	14.88	5.16
Predicted by Nearest Rule	7	3.25	1.12	4	57.14	1.86	0.64	3	42.85	1.39	0.48
Totals	215	100.00	34.66	180		83.72	29.01	35		16.27	5.64

Figure 7.3.10: Summary Statistics Associated with the Selected β-reduct $\{c_2, c_4, c_8\}$

The 'Predictive Summary Stats' panel in Figure 7.3.10, allows the analyst to quickly observe the performance of a selected β-reduct, and to compare between the performance based on the training and validation sets. From experience, the summary panel is most useful at the initial stages of the analysis, when searching for β-reducts that have a good predictive accuracy on the validation set and can predict a large portion of the banks by matching rules (displayed as 'Predictable Objects' in the first column within the table in Figure 7.3.10). The analyst may then further investigate any β-

reducts of interest, using the more comprehensive panels of information described throughout this section.

Due to constraints on the size of this dissertation, presentation of a deeper analysis involving screenshots based on more β-reducts is not presented. However, the following section displays in tabular form, the collated results of all the β-reducts identified from the FIBR data set for discussion and comparison.

For the single β-reduct investigated, the predictive accuracies were as stated previously, quite respectable and comparable with other studies (Oelericha and Poddig, 2006). Although noticeably, the rule set lacked the capability to predict 'A' and 'E' grade banks. The following subsection proves that it is possible to select β-reducts with the capability of predicting these ratings (albeit by weak rules). Unfortunately, with only a minimal number of banks representing the 'A' and 'E' rating grades within the validation set, it is difficult to asses the predictive performance, of the rules capable of predicting bank grades belonging to these under represented decision classes.

The information panels described in this subsection 7.3, have all been within the 'Analysis' panel of the vein graph analysis. These information panels are also available in the VPRS re-sampling analysis described in the next chapter, again under the 'Analysis' panel. This permits some consistency between both the vein graph analysis and the re-sampling analysis, allowing the analyst to contrast the results between both versions of the developed VPRS software.

7.4 Comparison of FIBR β-reducts

This subsection contains the necessary information collated from the vein graph analysis of the FIBR data, to allow comparisons to be made, and conclusions to be drawn upon the identified β-reducts. The information is represented in tabular form, starting with Table 7.4.1, which displays the number of rules, QoC and QoA (both as percentages), associated with each β-reduct over all β sub-domains.

β-reduct	β Sub-domain	Number of Rules	QoC (%)	QoA (%)
$\{c_2, c_4, c_8\}$	(0.5, 0.5238]	13	95.06	74.54
$\{c_2, c_4, c_7, c_8\}$	(0.5238, 0.5385]	23	95.06	77.92
$\{c_1, c_3, c_4, c_6, c_8\}$	(0.5 0.5556]	67	95.06	85.71
$\{c_1, c_2, c_3, c_4, c_5, c_8\}$	(0.5238, 0.6]	96	95.06	90.64
$\{c_1, c_2, c_3, c_4, c_5, c_7, c_8\}$	(0.6, 0.6667]	109	95.06	92.46
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	(0.6667, 0.7143]	108	87.65	94.92
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	(0.741, 0.75]	107	85.92	95.40
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	(0.75, 0.7778]	107	80.98	96.64
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	(0.7778, 0.8]	106	78.76	97.17
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	(0.8, 0.8571]	106	75.06	98.02
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	(0.8571, 0.9]	105	73.33	98.31
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	(0.9 0.9355]	103	68.39	98.91
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	(0.9355, 0.95]	102	60.74	99.59
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	(0.95, 1.0]	101	55.80	100.00

Table 7.4.1:FIBR β-reducts and Associated Information

It can be seen in Table 7.4.1, as the size (number of associated condition attributes) of a β-reduct increases, there is a corresponding increase in the number of associated rules. This is a consequence of, the potential increase in the number of condition classes associated with the β-reducts, as their sizes' increase. The effect on the rule set, is to produce more rules but with weaker strengths, as there are less supporting banks in the associated condition classes. Taking β-reduct $\{c_1, c_2, c_3, c_4, c_5, c_8\}$ for example, it is associated with 96 rules, based on a the training set of 405 banks; this implies that the average rule strength is just over 4 (405 divided by 96). Although, when inspecting the rule set associated with β-reduct $\{c_1, c_2, c_3, c_4, c_5, c_8\}$ (not shown), many of the rules are much stronger (with 20 or more banks in support of some individual rules) and hence, by deduction (and by inspection), it was evident that a large portion of the remaining weaker rules, were based upon a condition class associated with a single bank, and associated with a certainty value of 1.0. Furthermore, the size and complexity of the rule sets associated with the larger β-reducts, makes them practically impossible to interpret.

The smaller sized β-reducts however, have smaller size rule sets, containing stronger more general rules, which typically offer better opportunity for interpretation by the analyst, relative to the larger rule sets. They are more general in the sense that, a single rule may classify relatively more banks, however, there is more potential for banks to be misclassified.

Although the number of rules associated with the β-reducts tends to increase as the size of the β-reduct increases; for a β-reduct multiply identified over more than one β sub-domain, selecting a sub-domain associated with a higher β threshold value appears to decrease the number of associated rules. This is observable with regards to the β-reduct $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$ (equal to the full set of condition attributes *C*), whose β sub-domains shown in Table 7.4.1 are associated with higher β threshold values than the other β-reducts. To explain this relationship, as the β threshold value is increased, less condition classes are included in the analysis because of the majority inclusion principle, hence rules supported by relatively weaker condition classes are eliminated. In theory, this should imply that only the more accurate rules remain, with the rule sets classifying less banks (decreased QoC). Though in practice, particularly with regards to this analysis, it reduces the rule set to a set of weaker rules (many based on a single banks) with higher certainties.

Table 7.4.1 also shows the inverse relationship between the β-reducts' associated β threshold values and the concomitant QoCs, with the β-reducts associated with higher β threshold values having lower concomitant QoCs. Note also, that as the QoC decreases, there is a corresponding increase in QoA. According to the theory (Beynon, 2001), this is a consequence of the associated rule set containing more accurate rules, but at the expense of classifying less banks.

Increased accuracy is associated with the larger sized β-reducts, as a consequence of their relatively larger rule sets. This increase in accuracy, is also shown in the predictive accuracies based across the individual decision classes of the training set, as displayed in Table 7.4.2.

β-reduct	Overall Predictive Accuracy	Individual Decision Class Predictive Accuracies				
		0	1	2	3	4
$\{c_2, c_4, c_8\}$	74.54	0.00	88.70	71.69	72.00	0.00
$\{c_2, c_4, c_7, c_8\}$	77.92	0.00	93.06	73.83	76.92	0.00
$\{c_1, c_3, c_4, c_6, c_8\}$	85.71	0.00	95.85	80.73	90.24	46.15
$\{c_1, c_2, c_3, c_4, c_5, c_8\}$	90.64	30.76	97.05	90.82	90.12	66.66
$\{c_1, c_2, c_3, c_4, c_5, c_7, c_8\}$	92.46	30.76	98.81	93.57	91.46	66.66
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	94.92	36.36	99.37	96.07	95.94	57.14
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	95.40	36.36	99.37	96.07	95.65	80.00
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	96.64	44.44	99.34	96.93	96.92	100.00
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	97.17	57.14	99.31	96.93	96.92	100.00
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	98.02	57.14	100.00	98.91	96.72	100.00
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	98.31	66.66	100.00	98.91	96.72	100.00
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	98.91	66.66	100.00	98.64	100.00	100.00
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	99.59	80.00	100.00	100.00	100.00	100.00
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	100.00	100.00	100.00	100.00	100.00	100.00

Table 7.4.2: Predictive Accuracies Associated with the Full Set of β-reducts, Applied to the FIBR Training Set

Table 7.4.2 displays the predictive accuracies for each β-reduct on the FIBR training set. The second column displays the overall predictive accuracies, and the remaining columns display the predictive accuracies associated with each individual decision class. It can clearly be seen that, as the size of the β-reducts increases (and hence the associated rule set increases), that both the overall predictive accuracies and the predictive accuracies over the five decision classes, also increases. This could initially lead the analyst to assume that the larger β-reducts would perform better in terms of classification accuracy on any future unseen data, but as Table 7.4.3 shows, based on the validation set, it would be a naïve assumption.

Table 7.4.3 displays the predictive accuracies for each β-reduct on the FIBR validation set. Note that there is not much difference between the predictive accuracies based on the smaller β-reducts and those based on the larger β-reducts. This demonstrates that, the results based on the training set are, as anticipated, misleading and over-optimistic (anticipated, based on the theory outlined in Chapter 3). Moreover, these results lead to the hypothesis that the rule sets associated with the larger β-reducts, appearing in Table 7.4.2, demonstrate a clear case of overfitting (see Chapter 3 subsection 3.2.2). That is, the considerable amount of weak rules associated with the larger β-reducts are affectively predicting the banks they were constructed on (typically a single bank), but are of little value in predicting unseen banks such as those in the validation set. It is most likely that these weaker rules, especially those based on a single bank, are not representative of any general trends.

β-reduct	Overall Predictive Accuracy	Individual Decision Class Predictive Accuracies				
		0	1	2	3	4
$\{c_2, c_4, c_8\}$	84.61	0.00	92.25	65.90	80.00	0.00
$\{c_2, c_4, c_7, c_8\}$	84.21	0.00	93.52	64.44	73.91	0.00
$\{c_1, c_3, c_4, c_6, c_8\}$	84.00	0.00	95.45	63.63	63.63	0.00
$\{c_1, c_2, c_3, c_4, c_5, c_8\}$	82.03	0.00	94.11	63.04	52.17	0.00
$\{c_1, c_2, c_3, c_4, c_5, c_7, c_8\}$	83.65	0.00	94.28	64.44	59.09	0.00
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	84.04	0.00	95.16	58.13	75.00	0.00
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	84.04	0.00	95.16	58.13	75.00	0.00
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	84.53	0.00	95.00	58.53	78.94	0.00
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	84.26	0.00	94.87	58.53	78.94	0.00
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	86.14	0.00	95.32	66.66	78.94	0.00
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	86.30	0.00	95.41	65.00	78.94	0.00
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	85.97	0.00	95.41	61.11	78.94	0.00
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	84.56	0.00	94.68	61.11	78.94	0.00
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	82.57	0.00	93.50	61.11	78.94	0.00

Table 7.4.3: Predictive Accuracies Associated with the Full Set of β-reducts, Applied to the FIBR Validation Set

However, unlike in the circumstances of overfitting associated with other classifier methods, such as neural networks, where predictive accuracy can decrease in relation to overfitting, here, there is no decrease in the predictive accuracies associated with the larger β-reducts. Possible, because there are still a small proportion of rules associated with the larger β-reducts, that are representing general trends (retaining residual strength as the condition classes decrease in size), and they are stronger than, and more likely to be used than, the other weaker rules in the rule set. Though, these slightly stronger rules are not necessarily as strong as the rules associated with the smaller sized β-reducts.

Finally, the number of rules within the rule sets, which predict each individual decision class, must be given some consideration. Table 7.4.4 displays the breakdown of the rules associated with each β-reduct over all β sub-domains, and the number of rules that classify to each individual decision class.

β-reduct	Number of rules predicting Each Individual Decision Class					Total
	0	1	2	3	4	
$\{c_2, c_4, c_8\}$	0	5	6	2	0	13
$\{c_2, c_4, c_7, c_8\}$	0	9	9	4	0	22
$\{c_1, c_3, c_4, c_6, c_8\}$	0	20	27	17	3	67
$\{c_1, c_2, c_3, c_4, c_5, c_8\}$	4	33	35	19	5	96
$\{c_1, c_2, c_3, c_4, c_5, c_7, c_8\}$	4	36	43	21	5	109
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	4	39	40	21	4	108
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	4	39	40	20	4	107
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	4	41	39	19	4	107
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	4	40	39	19	4	106
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	4	39	40	19	4	106
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	4	38	40	19	4	105
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	4	38	38	19	4	103
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	4	37	38	19	4	102
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	4	36	38	19	4	101

Table 7.4.4: Number of Rules Predicting Each Individual Decision Class

It is clear from Table 7.4.4 that, although β-reduct $\{c_2, c_4, c_8\}$ has comparable predictive performance to the larger β-reducts (based on the results of Table 7.4.3), as stated in the previous section, it lacks rules with the capability to predict the 'A' (0) and 'E' (4) grade banks. Thus, it might be wiser for the analyst to select from the vein graph analysis, a β-reduct with a slightly higher associated β threshold value.

Unfortunately, the limited size of the validation set mitigates any real possibility of assessing their predictive performance on the 'A' and 'E' decision classes. This issue is a significant factor with regards to the next chapter on re-sampling, and β-reduct aggregation of the FIBR data. That is, during the automated β-reduct selection, should the β value be set higher than the β threshold value associated with β-reduct $\{c_2, c_4, c_8\}$ (investigated in the previous section), as a higher β threshold value would allow the automated process more opportunity to select β-reducts that are capable of predicting the 'A' and 'E' grade banks.

The great advantage of re-sampling over the vein graph analysis, is the use of the out-of-bag predictive accuracies (as described in Chapter 3 section), because the out-of-bag estimations are based on the training set, which includes 'A' and 'E' grade banks (since the training set is repetitively split into the in-sample and out-of-sample data sets). Hence, the out-of-bag estimates are more inclusive of all the decision classes. Also, the out-of-bag estimations are not compromised by overfitting because they use the out-of-sample data. Hence, the out-of-bag estimates reflect a more candid estimation of the predictive accuracies, than that provided here, by the validation set.

7.5 Summary

This chapter has presented the VPRS vein graph analysis of the FIBR data set. The developed software has been demonstrated through the selection of a single β-reduct, and the rules associated with that β-reduct applied to the training and validation sets. A number of panels of information were described, which offered further breakdown, transparency and incite into the predictive accuracy results.

Comparisons have been drawn between all the identified β-reducts, based on the results of predicting the banks belonging to the training and validation sets. The predictive accuracies, in general, are relatively respectable (Oelericha and Poddig, 2006), and confidence can be taken that the system is capable (in the case of FIBRs), of producing β-reducts that can classify the banks with a reasonable degree of accuracy.

It is evident from the results, that, although it appears that larger β-reducts provide a superior predictive performance based on the training set, this trend does not translate to the validation set. Moreover, there is no evidence that the larger β-reducts associated with higher β threshold values perform any better than the smaller β-reducts which are associated with, lower β threshold values and smaller more interpretable rule sets.

It is clear that the results based on the training set tend to be over-optimistic and that the validation set provides more transparent results, particularly in relation to the predictive accuracies associated with the individual decision classes. It should be noted though, the training set does indicate that, β-reducts whose β domains are associated with a β-threshold value above a certain level, are more capable of predicting the under represented 'A' and 'E' bank ratings (see Table 7.4.2).

Based on the information presented from the analysis within this chapter, the analyst can make judgements on how a re-sampling analysis may perform. Moreover, it allows them to judge the β threshold value they should choose for the automated β-reduct selection process (used within the following chapter concerned with the VPRS re-sampling software). That is, should they choose a very low value, say just over 0.5, that implies a greater opportunity for the system to select small sized β-reducts that have relatively smaller, more general rule sets, but less chance of having rules that can predict 'A' and 'E' grade banks; or should they choose a slightly higher β threshold value, say around 0.55, that implies a greater opportunity for the system to select β-reducts that have slightly larger, less interpretable rule sets, but with more chance of having rules that can predict 'A' and 'E' grade banks.

With regards to the performance of the developed software, over a number of different analyses, the functionality and the usability of the interface has proved quite affective. There is scope to improve the software's performance (in terms of speed), in particular, either through prior calculation and storage of all the rule sets and their classification of the banks in the training and validation sets; or storing the rule sets of the β-reducts that the analyst has selected during the analysis, hence allowing quick retrieval of the information if the analyst reselects one of the β-reducts.

The following chapter continues with the VPRS analysis of the FIBR data, but within the re-sampling environment, and culminates in an appraisal of the developed β-reduct aggregation process.

Chapter 8

Re-sampling Analysis of FIBR Data Set and Further Benchmark Analyses

This chapter continues the exposition of the developed software and the analysis of the FIBR data started in the previous two chapters. Here, the FIBR data will be analysed using the three re-sampling methods, as described in Chapter 3, namely, leave-one-out, *k*-fold cross-validation, and bootstrapping, within the context of the developed VPRS re-sampling software. The results of these re-sampling analyses, and the predictive performances based on the aggregation of selected β-reducts, from each of these analyses, are compared and contrasted. A further set of results are also presented in this chapter, based on the application of the VPRS re-sampling software to a number of data sets for the purpose of benchmarking against other studies.

The sections within this chapter are described as follows:

- Section 8.1 **Exposition of the VPRS Re-sampling Software Within the Leave-one-out Environment**. This section exposits the re-sampling and β-reduct aggregation aspects of the developed VPRS re-sampling software, within the context of the leave-one-out analysis.
- Section 8.2 ***k*-fold Cross-validation, Comparison with Leave-one-out**. This section compares the results of the *k*-fold cross-validation and leave-one-out analyses in the context of the developed VPRS re-sampling software. With particular attention being drawn to the asymptotic

nature of the two re-sampling methods as *k* tends to the number of banks ***n***.

- Section 8.3 **Bootstrap Re-sampling and Bootstrap *β*-reduct Aggregation**. This section follows the boostrap re-sampling of the FIBR data. It also discusses the issues pertinent to the process of *β*-reduct aggregation, with regards to the bootstrapping results, and demonstrated within the software.
- Section 8.4 **Comparison of Leave-one-out, *k*-fold Cross-validation and Bootstrapping Predictive Results**. This section compares the results of the estimated predictive accuracies across the three re-sampling methods. The predictive results of the aggregated *β*-reducts with regards to leave-one-out, bootstrapping, and those *β*-reducts identified from the single run vein graph analysis of the FIBR data (from Chapter 7), are also compared and discussed.
- Section 8.5 **Further Benchmark Results**. This section presents a VPSRS re-sampling analysis of a number of data sets for the purpose of benchmarking. The results are compared to a number of other studies and classifier methods, in particular Multi Discriminant Analysis.
- Section 8.6 **Summary**. This section summarises the many results presented within this chapter, with regards to both the re-sampling predictive accuracies and the aggregated *β*-reducts.

From the initial parameter set-up phase of the VPRS software (see Chapter 6 section 6.1), the analyst has the option of selecting one of the three re-sampling methods (as opposed to the single run vein graph analysis). If they choose *k*-fold cross-validation, they have the further option of choosing stratified or non-stratified *k*-fold cross-validation (see Chapter 3 subsection 3.1.1). The subsequent pre-processing of the FIBR data by the software, is identical to that described in Chapter 6. However the following stage, implements a VPRS re-sampling analysis of the FIBR data. Section 8.1. describes a leave-one-out analysis, but the description of the software given in that section is also pertinent to both *k*-fold cross-validation and bootstrapping analyses.

8.1 Exposition of the VPRS Re-sampling Software Within the Leave-one-out Environment

This section presents, a leave-one-out analysis of the FIBR data, and demonstrates the β-reduct aggregation process. Before the software based results are presented, it is necessary to reiterate the re-sampling β-reduct selection process, as previously described in Chapter 3 subsection 3.5.1. The selection of a β-reduct at each repetition, is based on the automatic selection criteria (see subsection 3.5.1), which essentially selects a β-reduct from the vein graph. Though, the vein graph and its peripheral elements are not literally constructed, only the associated calculations are undertaken (identification of the β-reducts and their concomitant β sub-domains), and recorded at each repetition. However, the most appropriate order of the automated selection criteria with regards to the analysis of the FIBR data undertaken here, differs to that outlined in Chapter 3, for reasons given next. The order of the criteria used here, is outlined below:

i. Selection of β-reduct(s) with a $\beta_{\min}$ threshold value greater than 0.55 (as suggested at the end of the previous chapter, sections 7.4 and 7.5).

ii. Selection of β-reduct(s) with least number of decision rules associated with the β-reduct(s) selected in i).

iii. Selection of β-reduct(s) from ii), with the highest quality of classification possible.

iv. Selection of β-reduct(s) with the highest $\beta_{\min}$ value from those β-reduct(s) selected in iii).

v. Selection of β-reduct(s) with least number of condition attributes associated with the β-reduct(s) selected in to iv).

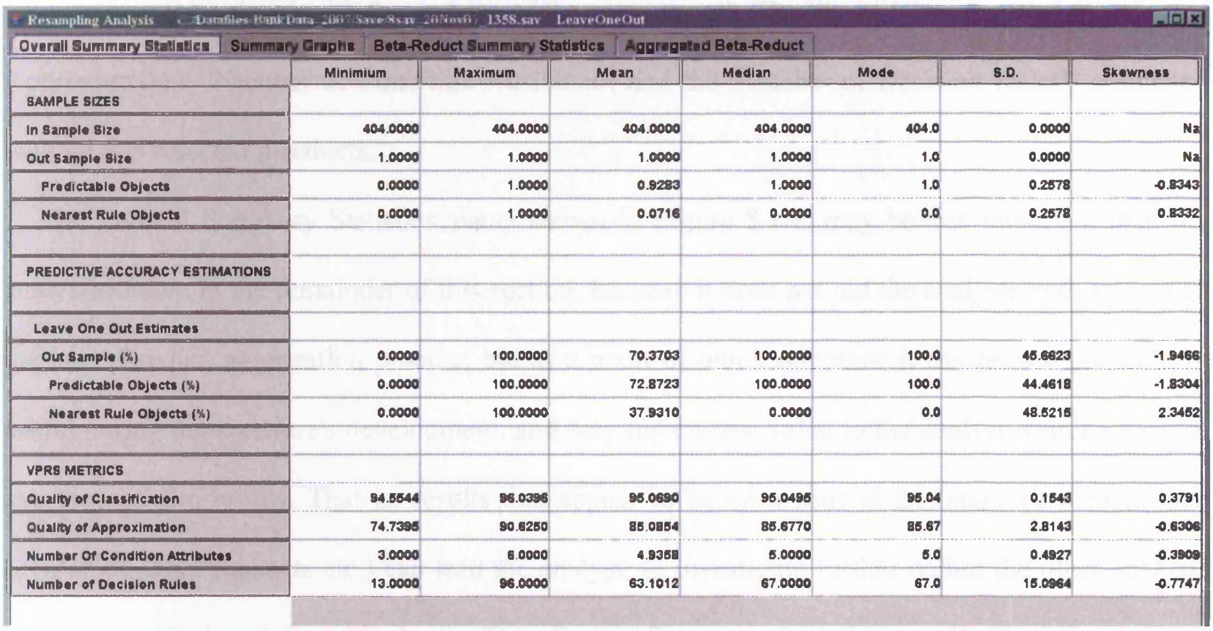
vi. Random selection of a single β-reduct from those remaining β-reducts after point v). Typically, a single β-reduct has been selected before this sixth step is required.

The most appropriate ordering of this criteria, was judged through experimentation with a number of different data sets (Griffiths and Beynon, 2007, 2008); which indicated that β-reducts associated with smaller rules sets (point ii., in the criteria shown here, previously point iv., in the criteria shown in Chapter 3 subsection 3.1.1), tended to correspond to higher predictive accuracies and higher QoCs, when compared to those β-reducts associated with larger rule sets. Thus, also following the science tenet of Occam's Razor, described previously (Domingos, 1999; all things being equal, the simplest solution tends to be the right one). However, through the evidence gained by the vein graph analysis of the considered FIBR data set, β-reducts associated with smaller rule sets, did not necessarily contain rules capable of classifying all five decision classes of the FIBR data set (see Chapter 7, Tables 7.4.2 and Table 7.4.4 for example).

Here, by allowing the analyst the autonomy to set the β-threshold value (during the parameters set-up phase, see Chapter 6 Figure 6.1.1), it is possible to increase the likelihood that a selected β-reduct will contain rules capable of predicting to all five decision classes. Thus, this criterion was placed first within the criteria. Though, increasing the β-threshold value, may be at the expense of the associated rules sets, being less general (rules associated with lower β-support and β-strengths, see Chapter 2 subsection 2.2.3). In addition, the rule set may be larger and less interpretable. This balance between generality and decision class predictive scope, is a subjective decision the analyst must make.

Returning to the description of the developed VPRS re-sampling software, on continuing from the pre-processing phase to a re-sampling analysis, the analyst is initially presented with a screenshot as shown in Figure 8.1.1, which presents the overall summary statistics of a number of metrics based on all 405 selected β-reducts, selected during the 405 repetitions of the leave-one-out analysis (or associated re-sampling analysis if another re-sampling option was chosen). Within Figure 8.1.1, the columns display the descriptive statistics, minimum, maximum, mean, median, mode, standard deviation (S.D.) and skewness with regards to the associated metrics, shown in the row headers, and described next.

The first four rows of values, under the heading 'SAMPLE SIZES', show the descriptive statistics associated with the in-sample and out-of-sample (out-of-bag) size metrics, under the headings 'In Sample Size' and 'Out Sample Size', respectively. The out-of-sample is further broken down into the number of banks that were predictable by a matching rule and those predicted by nearest rules, under the headings 'Predictable Objects' and 'Nearest Rule Objects', respectively (see Chapter 3 section 3.6 for nearest rule explanation).


Resampling Analysis   C:\Datafiles\BankData_2007\Save\8sav_20Nov07_1358.sav   LeaveOneOut
Overall Summary Statistics
Summary Graphs
Beta-Reduct Summary Statistics
Aggregated Beta-Reduct

	Minimium	Maximum	Mean	Median	Mode	S.D.	Skewness
SAMPLE SIZES							
In Sample Size	404.0000	404.0000	404.0000	404.0000	404.0	0.0000	Na
Out Sample Size	1.0000	1.0000	1.0000	1.0000	1.0	0.0000	Na
Predictable Objects	0.0000	1.0000	0.9283	1.0000	1.0	0.2578	-0.8343
Nearest Rule Objects	0.0000	1.0000	0.0716	0.0000	0.0	0.2578	0.8332
PREDICTIVE ACCURACY ESTIMATIONS							
Leave One Out Estimates							
Out Sample (%)	0.0000	100.0000	70.3703	100.0000	100.0	45.6623	-1.9466
Predictable Objects (%)	0.0000	100.0000	72.8723	100.0000	100.0	44.4618	-1.8304
Nearest Rule Objects (%)	0.0000	100.0000	37.9310	0.0000	0.0	48.5215	2.3452
VPRS METRICS							
Quality of Classification	94.5544	96.0396	95.0690	95.0495	95.04	0.1543	0.3791
Quality of Approximation	74.7395	90.6250	85.0854	85.6770	85.67	2.8143	-0.6306
Number Of Condition Attributes	3.0000	6.0000	4.9358	5.0000	5.0	0.4927	-0.3909
Number of Decision Rules	13.0000	96.0000	63.1012	67.0000	67.0	15.0964	-0.7747

Figure 8.1.1: Summary Statistics of the Leave-one-out Analysis of the FIBR Training Set

As there are 405 banks within the training data set, there are 404 banks within each repetition of the leave-one-out analysis (404 banks in the in-sample, one in the out-of-sample). Consequently within Figure 8.1.1, the maximum, minimum, mean, median and mode for the 'In Sample Size' are all 404, the standard deviation is zero and there is no skewness value to calculate. Similarly, the values for the 'Out Sample Size' are also recorded as one, with a standard deviation of zero and no skewness.

The three rows under the heading 'PREDICTIVE ACCURACY ESTIMATIONS', and under the sub heading 'Leave One Out Estimates', display the descriptive statistics relating to the concomitant predictive accuracies, as percentages on the out-of-sample, associated with all selected β-reducts. It is further broken down, to display the descriptive statistics relating to, those banks within the out-of-sample that are predictable by a matching rule, and those only predictable by a nearest rule. With regards to the leave-one-out method, because the single bank in the out-of-sample is either predicted correctly or incorrectly, the minimum and maximum values are respectively, 0% (predicted incorrectly) or 100% (predicted correctly).

The final four rows under the heading 'VPRS METRICS', display the descriptive statistics under the headings (associated with the metrics of the same name), 'Quality of Classification', 'Quality of Approximation', 'Number of Condition Attributes', and the 'Number of Decision Rules', associated with all 405 selected β-reducts.

The 'Overall Summary Statistics' panel shown in Figure 8.1.1, may be less important than the analyses shown in the remainder of this section, because it does not aid the analysts with respect to the later β-reduct aggregation process; but as a point of quick reference it has proved particularly useful during the software's development, and may have some value to the analyst wanting a quick overview of the results. That is, results that appear to be erroneous at this stage (low predictive accuracies, large rule sets etc.) can lead the analyst, to investigate further within the other analysis panels (described next, in particular the 'Beta-Reduct Summary Statistics' panel). The information is also useful with respect to investigating the bootstrap re-sampling method (described in section 8.3). Furthermore, the average (mean) predictive accuracy estimates also allow comparisons to be drawn between the three re-sampling methods.

The remainder of this section is divided into three subsections, describing the three remaining panels within the software screenshot shown in Figure 8.1.1, namely, 'Summary Graphs', 'Beta-reduct Summary Statistics' and 'Aggregated Beta-Reduct'. The subsection headings relate to those headings, respectively.

8.1.1 Summary Graphs

The 'Summary Graphs' panel, displays three graphs associated with the selected β-reducts. The first graph, shown in Figure 8.1.1.1, displays the occurrence of the top ten most frequently selected β-reducts.

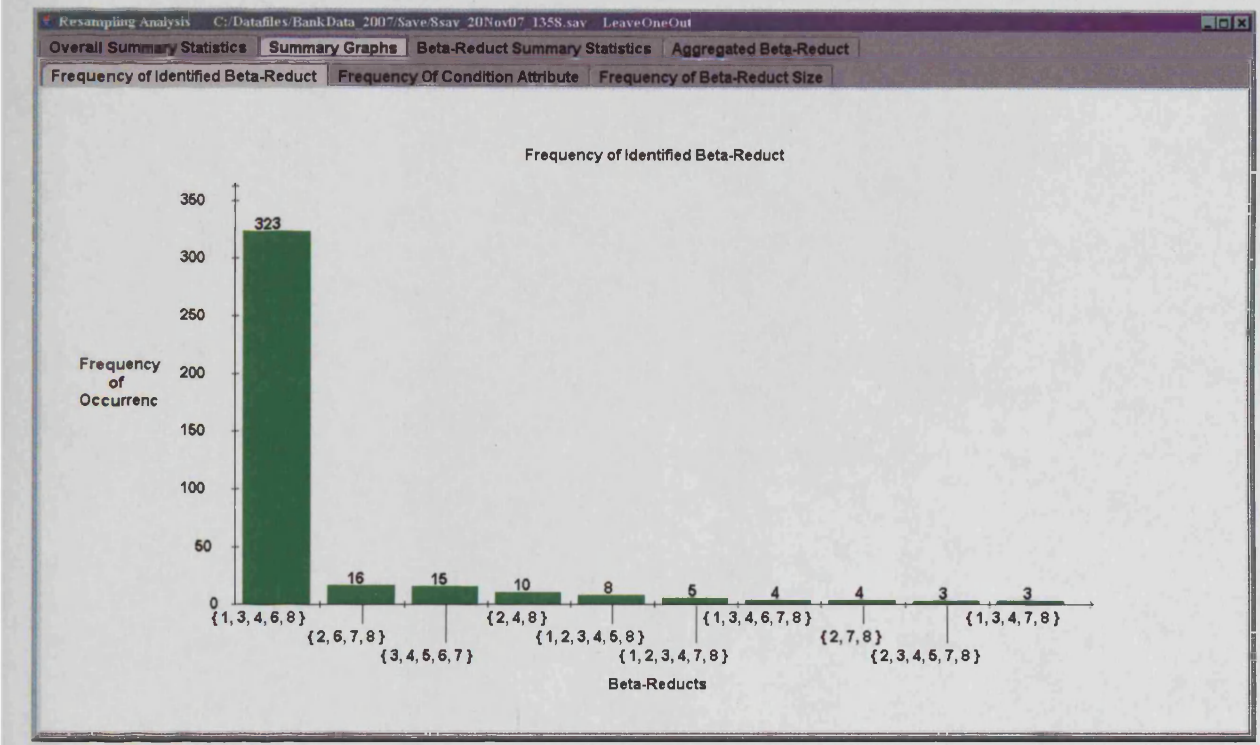

Resampling Analysis C:/Datafiles/BankData_2007/Save/Ssav_20Nov07_1358.sav LeaveOneOut
Overall Summary Statistics
Summary Graphs
Beta-Reduct Summary Statistics
Aggregated Beta-Reduct
Frequency of Identified Beta-Reduct
Frequency Of Condition Attribute
Frequency of Beta-Reduct Size
Frequency of Identified Beta-Reduct
Frequency of Occurrenc
350
300
250
200
150
100
50
0
323
16
15
10
8
5
4
4
3
3
{1, 3, 4, 6, 8}
{2, 6, 7, 8}
{3, 4, 5, 6, 7}
{2, 4, 8}
{1, 2, 3, 4, 5, 8}
{1, 2, 3, 4, 7, 8}
{1, 3, 4, 6, 7, 8}
{2, 7, 8}
{2, 3, 4, 5, 7, 8}
{1, 3, 4, 7, 8}
Beta-Reducts

Figure 8.1.1.1: Frequency of Occurrence of the Selected β-reducts, Associated with the Leave-one-out Analysis of the FIBR Training Set

Figure 8.1.1.1 indicates that, within the leave-one-out analysis, the β-reduct $\{c_1, c_3, c_4, c_6, c_8\}$ was most frequently selected by the automated selection criteria; moreover, it was particularly dominant, being selected 323 out of the 405 repetitions (nearly 80% of the repetitions). It should be noted that β-reduct $\{c_1, c_3, c_4, c_6, c_8\}$, also appears in the vein graph analysis of the FIBR training data (see Chapter 7 Figure 7.2.2), as do the β-reducts $\{c_2, c_4, c_8\}$ and $\{c_1, c_2, c_3, c_4, c_5, c_8\}$ which also appear in Figure 8.1.1.1. Thus, showing some consistency with the single run vein graph analysis of the FIBR data, previously exposited. Additionally, in some respects, the β-reduct $\{c_1, c_3, c_4, c_6, c_8\}$ was the target β-reduct of the automated selection criteria (under the order of automated selection criteria set here, with the β-threshold value set to 0.55 for the first criterion point i), as the collated results of the vein graph analysis showed in the previous chapter (see subsection 7.4.4), it had the potential to predict banks from the minority 'E' grade decision class, and was associated with a relatively medium sized rule set.

It is interesting to note, for about 20% of the 405 repetitions, by selecting a particular single bank as the out-of-sample (as per the leave-one-out process), it affected the β-reduct selected, that is, a β-reduct was selected other than β-reduct $\{c_1, c_3, c_4, c_6, c_8\}$. It will be shown later in sections 8.2 and 8.3, that there is more variation within the β-reducts selected, for the k-fold cross-validation and bootstrapping analyses, because the out-of-sample sizes are larger (more than one bank). Also note, that the vein graph analysis of the FIBR data, did not provide the analyst with this extra level of knowledge to aid in their decision making.

As the selection of a single particular bank for the out-of-sample can affect the selected β-reduct from the in-sample, it is interesting to see how this affects the condition attributes associated with all selected β-reducts. The graph shown within Figure 8.1.1.2, displays the distribution (frequency of occurrence) of the condition attributes within all selected β-reducts.

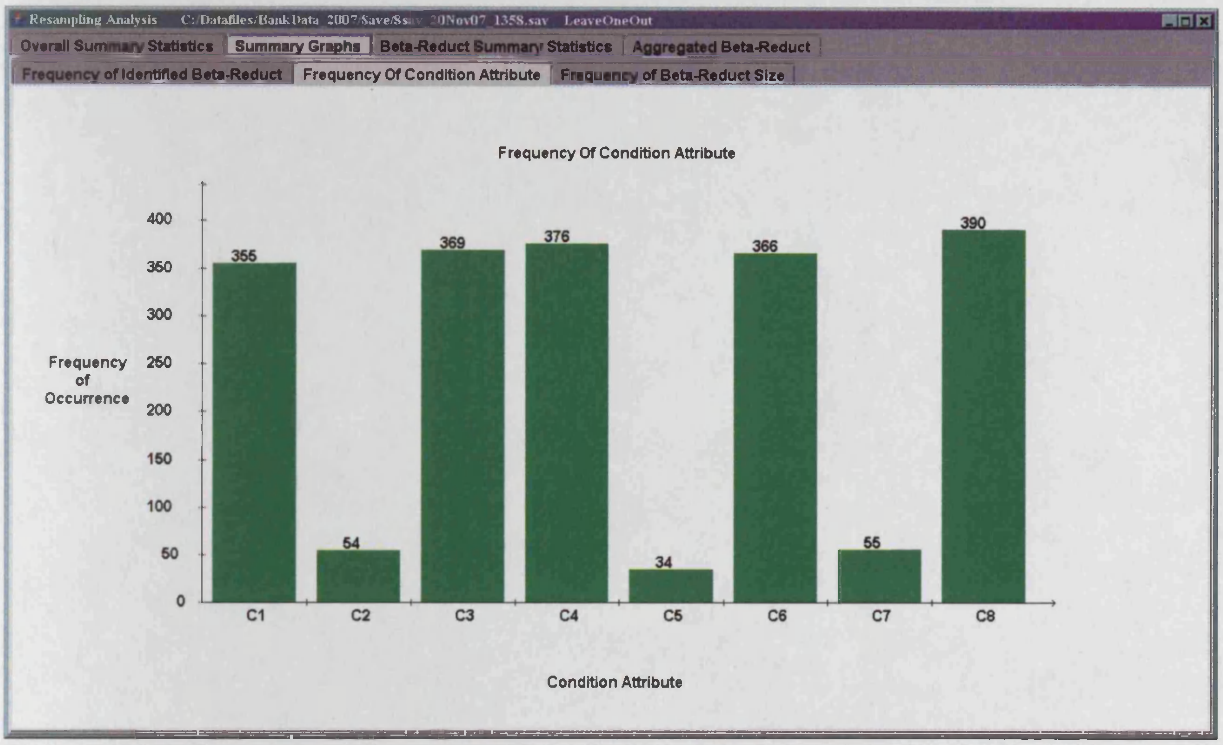

Resampling Analysis C:/Datafiles/BankData_2007/Save/Ssav_20Nov07_1358.sav LeaveOneOut
Overall Summary Statistics
Summary Graphs
Beta-Reduct Summary Statistics
Aggregated Beta-Reduct
Frequency of Identified Beta-Reduct
Frequency Of Condition Attribute
Frequency of Beta-Reduct Size
Frequency Of Condition Attribute
Frequency of Occurrence
0
50
100
150
200
250
300
350
400
355
54
369
376
34
366
55
390
C1
C2
C3
C4
C5
C6
C7
C8
Condition Attribute

Figure 8.1.1.2: Frequency of Occurrence of the Condition Attributes Associated with the Selected β-reducts, with Regards to the Leave-one-out Analysis of the FIBR Training Set

It can be seen from Figure 8.1.1.2, even though the out-of-sample affects which β-reduct is selected, the dominance of the most frequently occurring condition attributes associated with all selected β-reducts remains fairly stable. Indeed, the attribute c_8 (GDP/Head) within Figure 8.1.1.2, was present within 97% (390/405) of the repetitions, again an indication of the importance of that particular attribute (concurring with the vein graph analysis in Chapter 7 section 7.2). Information such as the importance or dominance of particular condition attributes across all β-reducts is less overt when considering the graph shown previously within Figure 8.1.1.1. Moreover, the most frequently occurring condition attributes c_1, c_3, c_4, c_6 and c_8, are directly associated with the most frequently selected β-reduct $\{c_1, c_3, c_4, c_6, c_8\}$.

The final graph, shown here within Figure 8.1.1.3, displays the frequency of selected β-reduct size (number of condition attributes associated with the selected β-reducts). Clearly, the graph within Figure 8.1.1.3 shows that the dominant β-reduct size is five; which is understandable, since the most frequently selected β-reduct $\{c_1, c_3, c_4, c_6, c_8\}$ contains five condition attributes. Here, with regards to the leave-one-out analysis, the graph is less interesting to the analyst, because of the dominance of the most frequently selected β-reduct $\{c_1, c_3, c_4, c_6, c_8\}$. However, as will be shown later, the dominant β-reduct associated with the bootstrap analysis is not reflected by the dominant size of the most frequently selected β-reduct, and this graphical analysis is most useful with regards to that particular re-sampling analysis.

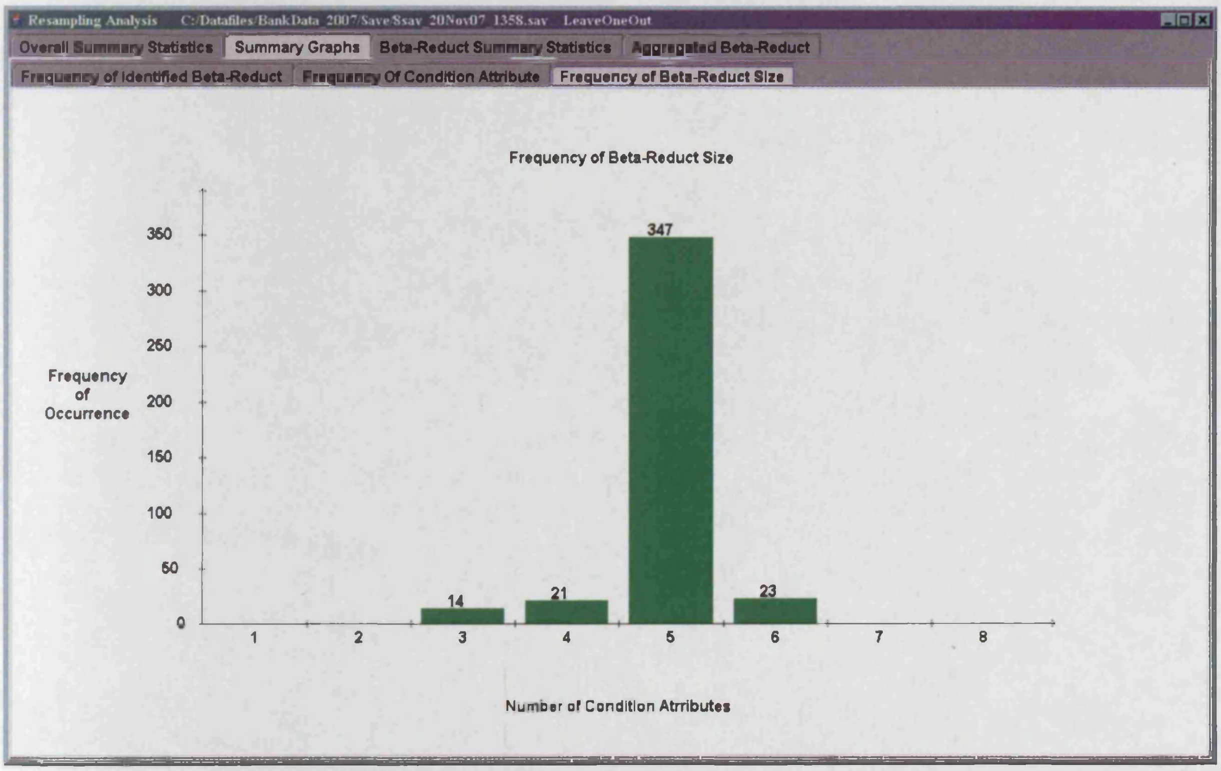

Resampling Analysis C:/Datafiles/BankData_2007/Save/Ssav_20Nov07_1358.sav LeaveOneOut
Overall Summary Statistics
Summary Graphs
Beta-Reduct Summary Statistics
Aggregated Beta-Reduct
Frequency of Identified Beta-Reduct
Frequency Of Condition Attribute
Frequency of Beta-Reduct Size
Frequency of Beta-Reduct Size
Frequency of Occurrence
350
300
250
200
150
100
50
0
347
14
21
23
1
2
3
4
5
6
7
8
Number of Condition Atrributes

Figure 8.1.1.3: Frequency of Selected β-reducts Size, Associated with the Leave-one-out Analysis of the FIBR Training Set

In general, the summary graphs presented (Figures 8.1.1.1 to 8.1.1.3) aid the analyst in the subsequent β-reduct aggregation process (shown later in subsection 8.1.3), as they convey information that may not be immediately obvious when considering the overall summary statistics, or the more in-depth individual β-reduct statistics (shown in the following subsection). Moreover, they have proved valuable during the development of the software, and for understanding the affects of the re-sampling processes, such as indicating the disparity between the most dominant selected β-reduct and most dominant size of β-reduct with regards to bootstrapping, and the asymptotic link between leave-one-out and k-fold cross-validation, described in the following sections 8.2 and 8.3.

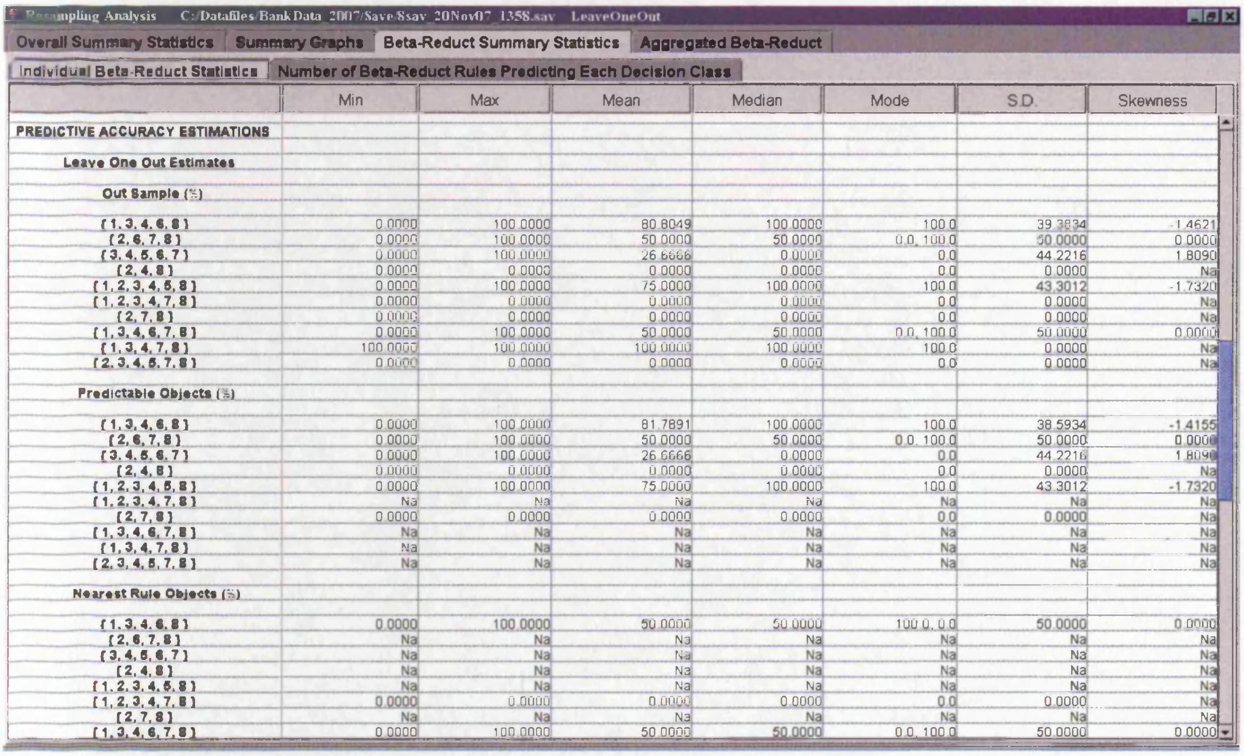
8.1.2 Beta-Reduct Summary Statistics

The 'Beta-Reduct Summary Statistics' panel provides the analyst with further information, relating to the reducts selected during the re-sampling process. There are two internal panels contained within the 'Beta-reduct Summary Statistics' panel, namely, 'Individual Beta-Reduct Statistics' and 'Number of Beta-reduct Rules Predicting Each Decision Class' panel, which contain statistical information relating to metrics associated with the occurrence of the top ten most frequently selected β-reducts. They give a comprehensive breakdown, with the specific aim to aid the analyst with their decision making during the β-reduct aggregation phase. These two internal panels are described in the following two subsections (internal relative to the 'Beta-reduct Summary Statistics' panel they are contained in).

8.1.2.1 Individual Beta-reduct Statistics

The 'Individual Beta-Reducts Statistics' panel, shown in Figure 8.1.2.1.1, displays statistical information associated with metrics regarding the occurrence of the top ten most frequently selected β-reducts (sorted into descending order). Note, it does not show the occurrence of each individually selected β-reduct, but rather statistics based on the accumulated results of β-reducts which have been selected and are identical (in terms of condition attributes), to any previously selected β-reducts. For example, the mean predictive accuracy on the out-of-sample associated with the most dominant β-reduct $\{c_1, c_3, c_4, c_6, c_8\}$, which occurred on 323 of the 405 repetitions is 80.8049%, shown in the eighth row, and the fourth column of the table within Figure 8.1.2.1.1. Note that the number of occurrences of each β-reduct is not obtainable within this table, but is obtainable from the summary graphs shown previously (Figure 8.1.1.1), or through the β-reduct aggregation phase described later.

The metrics utilised within Figure 8.1.2.1.1, are identical to the metrics described with regards to the 'Overall Summary' panel (sample sizes, predictive accuracy estimates etc. see Figure 8.1.1), but gives a comprehensive breakdown of the top ten most frequently selected β-reducts.


Resampling Analysis   C:/Datafiles/BankData_2007/Save/Ssav_20Nov07_1358.sav   LeaveOneOut
Overall Summary Statistics
Summary Graphs
Beta-Reduct Summary Statistics
Aggregated Beta-Reduct
Individual Beta-Reduct Statistics
Number of Beta-Reduct Rules Predicting Each Decision Class

	Min	Max	Mean	Median	Mode	S.D.	Skewness
PREDICTIVE ACCURACY ESTIMATIONS							
Leave One Out Estimates							
Out Sample (%)							
{1, 3, 4, 6, 8}	0.0000	100.0000	80.8049	100.0000	100.0	39.3834	-1.4621
{2, 6, 7, 8}	0.0000	100.0000	50.0000	50.0000	0.0, 100.0	50.0000	0.0000
{3, 4, 5, 6, 7}	0.0000	100.0000	26.6666	0.0000	0.0	44.2216	1.8090
{2, 4, 8}	0.0000	0.0000	0.0000	0.0000	0.0	0.0000	Na
{1, 2, 3, 4, 5, 8}	0.0000	100.0000	75.0000	100.0000	100.0	43.3012	-1.7320
{1, 2, 3, 4, 7, 8}	0.0000	0.0000	0.0000	0.0000	0.0	0.0000	Na
{2, 7, 8}	0.0000	0.0000	0.0000	0.0000	0.0	0.0000	Na
{1, 3, 4, 6, 7, 8}	0.0000	100.0000	50.0000	50.0000	0.0, 100.0	50.0000	0.0000
{1, 3, 4, 7, 8}	100.0000	100.0000	100.0000	100.0000	100.0	0.0000	Na
{2, 3, 4, 5, 7, 8}	0.0000	0.0000	0.0000	0.0000	0.0	0.0000	Na
Predictable Objects (%)							
{1, 3, 4, 6, 8}	0.0000	100.0000	81.7891	100.0000	100.0	38.5934	-1.4155
{2, 6, 7, 8}	0.0000	100.0000	50.0000	50.0000	0.0, 100.0	50.0000	0.0000
{3, 4, 5, 6, 7}	0.0000	100.0000	26.6666	0.0000	0.0	44.2216	1.8090
{2, 4, 8}	0.0000	0.0000	0.0000	0.0000	0.0	0.0000	Na
{1, 2, 3, 4, 5, 8}	0.0000	100.0000	75.0000	100.0000	100.0	43.3012	-1.7320
{1, 2, 3, 4, 7, 8}	Na	Na	Na	Na	Na	Na	Na
{2, 7, 8}	0.0000	0.0000	0.0000	0.0000	0.0	0.0000	Na
{1, 3, 4, 6, 7, 8}	Na	Na	Na	Na	Na	Na	Na
{1, 3, 4, 7, 8}	Na	Na	Na	Na	Na	Na	Na
{2, 3, 4, 5, 7, 8}	Na	Na	Na	Na	Na	Na	Na
Nearest Rule Objects (%)							
{1, 3, 4, 6, 8}	0.0000	100.0000	50.0000	50.0000	100.0, 0.0	50.0000	0.0000
{2, 6, 7, 8}	Na	Na	Na	Na	Na	Na	Na
{3, 4, 5, 6, 7}	Na	Na	Na	Na	Na	Na	Na
{2, 4, 8}	Na	Na	Na	Na	Na	Na	Na
{1, 2, 3, 4, 5, 8}	Na	Na	Na	Na	Na	Na	Na
{1, 2, 3, 4, 7, 8}	0.0000	0.0000	0.0000	0.0000	0.0	0.0000	Na
{2, 7, 8}	Na	Na	Na	Na	Na	Na	Na
{1, 3, 4, 6, 7, 8}	0.0000	100.0000	50.0000	50.0000	0.0, 100.0	50.0000	0.0000

Figure 8.1.2.1.1: Breakdown of Metric Descriptive Statistics, Relating to the Top Ten Most Frequently Selected β-reducts, Associated with the Leave-one-out Analysis

It is not possible, within one screenshot, to display all the information within the table shown in Figure 8.1.2.1.1. Hence, the vertical scroll bar has been scrolled down, to display only the statistics concerning the 'PREDICTIVE ACCURACY ESTIMATIONS' (as the results will be most pertinent to the following discussion).

As stated, the mean predictive accuracy for the β-reduct $\{c_1, c_3, c_4, c_6, c_8\}$ on the out-of-sample is 80.8049%. The mean predictive accuracy on the banks from the out-of-sample that are predictable by matching rule is 81.7891%. These predictive accuracies are encouragingly high, and more confidence can be taken from these results, compared to the predictive accuracies that would have been estimated on the validation set during a vein graph analysis (as shown in the previous chapter). That is, these out-of-bag estimates, include, in their estimated value, evidence gained from the predictive accuracy based on the available 'A' and 'E' grade banks (the under represented classes), which are almost absent within the validation set, but not so within the training set (see Chapter 6, section 6.1 for validation and training set banks' decision class distributions). Note that, the same validation set that was used for the vein graph analysis in the previous chapter, is used here, later in the chapter, to test the β-reduct aggregation process.

Within Figure 8.1.2.1.1, the mean predictive accuracies for banks predicted by nearest rule, are notably poorer than those predicted by matching rules, and this concurs with the results of the vein graph analysis, which showed the banks predicted by nearest rule were often incorrectly predicted (see chapter 7 Figure 7.3.7, and Appendix A.5). Also, the predictive accuracies associated with the β-reducts, other than β-reduct $\{c_1, c_3, c_4, c_6, c_8\}$ (the most dominant β-reduct), are also notably lower on the out-of-sample, over both, the predictable by matching rule, and predictable by nearest rule banks.

With regards to the reasons why these less dominant β-reducts may have been selected during the leave-one-out process; one may speculate that, the single banks left out, during those repetitions selecting the less dominant β-reducts, had unique condition attribute values and hence belonged to their own distinct condition classes (condition classes containing a single bank). By leaving one of these single banks out, it would have effectively removed a condition class, which in turn, affected the nature of the identified β-reducts (β_{min} values, Quality of Classification etc.).[23] Hence, there would have then been the possibility, that the criteria mentioned previously could have selected an alternate β-reduct to the dominant β-reduct $\{c_1, c_3, c_4, c_6, c_8\}$. Furthermore, if these banks formed their own unique condition classes, they may have been unpredictable by the rules constructed on the other condition classes, which could explain the low predictive accuracies reported in Figure 8.1.2.1.1, associated with the less dominant β-reducts.

An alternative reason for the selection of a less dominant β-reduct may be related to the absence

[23] This could be visualised as affecting the 'topology' of the veins within a vein graph analysis, that is, how the veins appear within the vein graph.

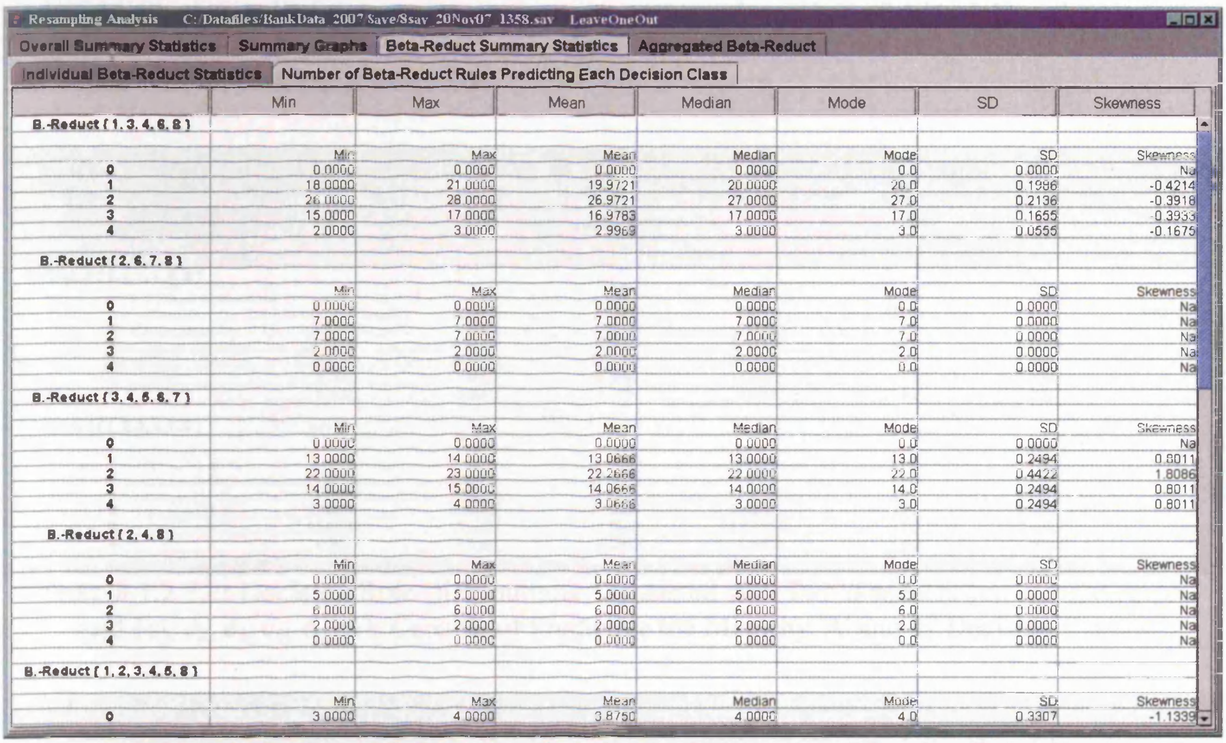
or presence of a single bank, resulting in, the difference between a condition class being associated to one decision class or another (in accordance with the majority inclusion principle); again this may affect the nature of the identified and consequently selected β-reducts.

Finally, with regards to the general trend of the leave-one-out estimated predictive accuracy results, it is notable within Figure 8.1.2.1.1, that the standard deviation is quite large, for all β-reducts where a standard deviation has been recorded. This conforms with the related literature (Weiss and Kulikowski, 1991) which suggested, the leave-one-out predictive accuracy estimate would have low bias but high variability (variance, here shown by the high standard deviation). This variability is understandable, because either a bank is predicted correctly (100% of the out-of-sample) or incorrectly (0% of the out-of-sample). The issues of bias and variability with regards to the analysis of the FIBR data, is discussed in more depth, within section 8.4.

8.1.2.2 Number of Beta-Reduct Rules Predicting Each Decision Class

The 'Number of Beta-Reduct Rules Predicting Each Decision Class' panel shown in Figure 8.1.2.2.1, displays statistical information relating to, the number of rules that predicted banks belonging to the different decision classes (FIBR rating grades 'A' to 'E' recoded '0' to '5'), associated with the top ten most frequently selected β-reducts. For example, there are on average (mean), zero rules associated with the dominant β-reduct $\{c_1, c_3, c_4, c_6, c_8\}$ capable of predicting 'A' grade banks (decision class '0'), 19.9721 for predicting the 'B' grade banks (decision class '1'), 26.9721 for predicting the 'C' grade banks (decision class '2'), 16.9783 for predicting the 'D' grade banks (decision class '3') and 2.9969 rules for predicting the 'E' grade banks (decision class '4').

The table presented within Figure 8.1.2.2.1, is particularly useful in aiding the analyst during the β-reduct aggregation and aggregated rule selection process, described in more depth later, in subsection 8.1.3.



Resampling Analysis C:/Datafiles/BankData_2007/Save/Ssav_20Nov07_1358.sav LeaveOneOut

Overall Summary Statistics
Summary Graphs
Beta-Reduct Summary Statistics
Aggregated Beta-Reduct
Individual Beta-Reduct Statistics
Number of Beta-Reduct Rules Predicting Each Decision Class

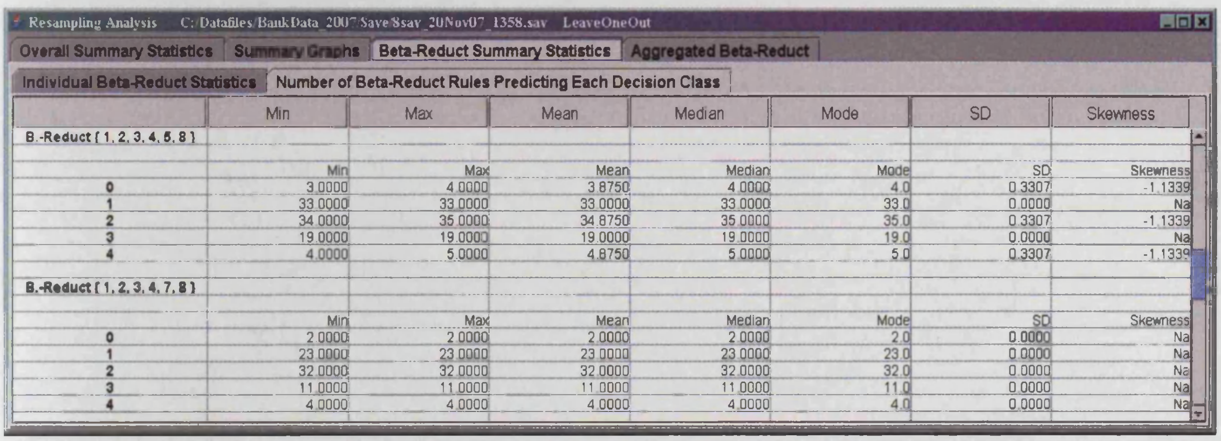
	Min	Max	Mean	Median	Mode	SD	Skewness
B.-Reduct { 1, 3, 4, 6, 8 }							
	Min	Max	Mean	Median	Mode	SD	Skewness
0	0.0000	0.0000	0.0000	0.0000	0.0	0.0000	Na
1	18.0000	21.0000	19.9721	20.0000	20.0	0.1986	-0.4214
2	26.0000	28.0000	26.9721	27.0000	27.0	0.2136	-0.3918
3	15.0000	17.0000	16.9783	17.0000	17.0	0.1655	-0.3933
4	2.0000	3.0000	2.9969	3.0000	3.0	0.0555	-0.1675
B.-Reduct { 2, 6, 7, 8 }							
	Min	Max	Mean	Median	Mode	SD	Skewness
0	0.0000	0.0000	0.0000	0.0000	0.0	0.0000	Na
1	7.0000	7.0000	7.0000	7.0000	7.0	0.0000	Na
2	7.0000	7.0000	7.0000	7.0000	7.0	0.0000	Na
3	2.0000	2.0000	2.0000	2.0000	2.0	0.0000	Na
4	0.0000	0.0000	0.0000	0.0000	0.0	0.0000	Na
B.-Reduct { 3, 4, 5, 6, 7 }							
	Min	Max	Mean	Median	Mode	SD	Skewness
0	0.0000	0.0000	0.0000	0.0000	0.0	0.0000	Na
1	13.0000	14.0000	13.0666	13.0000	13.0	0.2494	0.8011
2	22.0000	23.0000	22.2666	22.0000	22.0	0.4422	1.8086
3	14.0000	15.0000	14.0666	14.0000	14.0	0.2494	0.8011
4	3.0000	4.0000	3.0666	3.0000	3.0	0.2494	0.8011
B.-Reduct { 2, 4, 8 }							
	Min	Max	Mean	Median	Mode	SD	Skewness
0	0.0000	0.0000	0.0000	0.0000	0.0	0.0000	Na
1	5.0000	5.0000	5.0000	5.0000	5.0	0.0000	Na
2	6.0000	6.0000	6.0000	6.0000	6.0	0.0000	Na
3	2.0000	2.0000	2.0000	2.0000	2.0	0.0000	Na
4	0.0000	0.0000	0.0000	0.0000	0.0	0.0000	Na
B.-Reduct { 1, 2, 3, 4, 5, 8 }							
	Min	Max	Mean	Median	Mode	SD	Skewness
0	3.0000	4.0000	3.8750	4.0000	4.0	0.3307	-1.1339

Figure 8.1.2.2.1: Statistical Information Relating to the Number of β-reduct Rules Capable of Predicting Each Decision Class, Associated with the Top Ten Most Frequently Selected β-reducts, with Regards to the Leave-one-out Analysis of the FIBR Training Set

The table in Figure 8.1.2.2.1 indicates that, seven out of the top ten β-reducts do not have the capability to predict 'A' grade banks (not all β-reducts are shown within the screenshot), even though it was stated earlier that the β-threshold value had been set to a level (0.55), which would hopefully, force the system to select β-reducts capable of predicting all decision classes. It should be noted though, seven out of the ten most frequently occurring β-reducts in Figure 8.1.2.2.1 (a different set of seven) were capable of predicting the 'E' grade banks. Unfortunately, as stated previously, setting the β-threshold value posed a trade off, between the capability of the β-reducts to predict all decision classes (including minority decision classes 'A' and 'E') and decision rule generality (small set of high strength, interpretable decision rules).

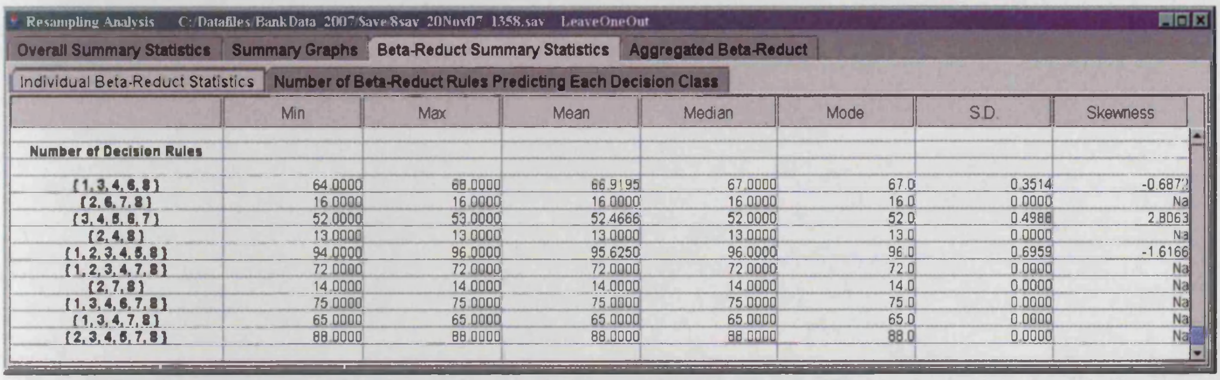
For example purposes, two β-reducts capable of predicting all decision classes are now considered. Figure 8.1.2.2.2 shows the rule set distributions scrolled down to display the two β-reducts $\{c_1, c_2, c_3, c_4, c_5, c_8\}$ and $\{c_1, c_2, c_3, c_4, c_7, c_8\}$, that were both capable of predicting the minority 'A' grade and 'E' grade banks; with Figure 8.1.2.2.3 showing the 'Individual Beta-Reduct

Statistics' panel, scrolled down to view the average (mean) number of rules (plus other statistics) associated with both of these β-reducts.


Resampling Analysis C:/Datafiles/BankData_2007/Save/8sav_20Nov07_1358.sav LeaveOneOut
Overall Summary Statistics
Summary Graphs
Beta-Reduct Summary Statistics
Aggregated Beta-Reduct
Individual Beta-Reduct Statistics
Number of Beta-Reduct Rules Predicting Each Decision Class

	Min	Max	Mean	Median	Mode	SD	Skewness
B.-Reduct { 1, 2, 3, 4, 5, 8 }							
	Min	Max	Mean	Median	Mode	SD	Skewness
0	3.0000	4.0000	3.8750	4.0000	4.0	0.3307	-1.1339
1	33.0000	33.0000	33.0000	33.0000	33.0	0.0000	Na
2	34.0000	35.0000	34.8750	35.0000	35.0	0.3307	-1.1339
3	19.0000	19.0000	19.0000	19.0000	19.0	0.0000	Na
4	4.0000	5.0000	4.8750	5.0000	5.0	0.3307	-1.1339
B.-Reduct { 1, 2, 3, 4, 7, 8 }							
	Min	Max	Mean	Median	Mode	SD	Skewness
0	2.0000	2.0000	2.0000	2.0000	2.0	0.0000	Na
1	23.0000	23.0000	23.0000	23.0000	23.0	0.0000	Na
2	32.0000	32.0000	32.0000	32.0000	32.0	0.0000	Na
3	11.0000	11.0000	11.0000	11.0000	11.0	0.0000	Na
4	4.0000	4.0000	4.0000	4.0000	4.0	0.0000	Na

Figure 8.1.2.2.2: Decision Rule Distributions Associated with Two β-reducts $\{c_1, c_2, c_3, c_4, c_5, c_8\}$ and $\{c_1, c_2, c_3, c_4, c_7, c_8\}$, Capable of Predictive the Minority 'A' and 'E' Decision Classes


Resampling Analysis C:/Datafiles/BankData_2007/Save/8sav_20Nov07_1358.sav LeaveOneOut
Overall Summary Statistics
Summary Graphs
Beta-Reduct Summary Statistics
Aggregated Beta-Reduct
Individual Beta-Reduct Statistics
Number of Beta-Reduct Rules Predicting Each Decision Class

	Min	Max	Mean	Median	Mode	S.D.	Skewness
Number of Decision Rules							
{ 1, 3, 4, 6, 8 }	64.0000	68.0000	66.9195	67.0000	67.0	0.3514	-0.687
{ 2, 6, 7, 8 }	16.0000	16.0000	16.0000	16.0000	16.0	0.0000	Na
{ 3, 4, 5, 6, 7 }	52.0000	53.0000	52.4666	52.0000	52.0	0.4988	2.8063
{ 2, 4, 8 }	13.0000	13.0000	13.0000	13.0000	13.0	0.0000	Na
{ 1, 2, 3, 4, 5, 8 }	94.0000	96.0000	95.6250	96.0000	96.0	0.6959	-1.6166
{ 1, 2, 3, 4, 7, 8 }	72.0000	72.0000	72.0000	72.0000	72.0	0.0000	Na
{ 2, 7, 8 }	14.0000	14.0000	14.0000	14.0000	14.0	0.0000	Na
{ 1, 3, 4, 6, 7, 8 }	75.0000	75.0000	75.0000	75.0000	75.0	0.0000	Na
{ 1, 3, 4, 7, 8 }	65.0000	65.0000	65.0000	65.0000	65.0	0.0000	Na
{ 2, 3, 4, 5, 7, 8 }	88.0000	88.0000	88.0000	88.0000	88.0	0.0000	Na

Figure 8.1.2.2.3: Statistical Information Relating to the Number of Rules Associated with the Top Ten Most Frequently Selected β-reducts, Associated with the Leave-one-out Analysis of the FIBR Training Set

It can be seen that, if the analyst was to consider the subsequent analysis based on the aggregation of the occurrences of the β-reduct $\{c_1, c_2, c_3, c_4, c_5, c_8\}$, Figure 8.1.2.2.3 implies, the average number of rules required would be, almost thirty more than compared with the dominant β-reduct $\{c_1, c_3, c_4, c_6, c_8\}$ (i.e. 66.9195 compared to 95.6250). The rule increase associated with β-reduct $\{c_1, c_2, c_3, c_4, c_7, c_8\}$ is lower (i.e. 66.9195 compared to 72.0000), but the analyst must consider the other information associated with the β-reducts, such as the stability of the β-reducts (their frequency of occurrence), other metric statistics (such as those shown in Figure 8.1.2.1.1) and the summary graphs. For example, according to the predictive accuracies within the table shown in

Figure 8.1.2.1.1, there is no reason to believe that the β-reduct $\{c_1, c_2, c_3, c_4, c_7, c_8\}$ will out-perform the dominant β-reduct $\{c_1, c_3, c_4, c_6, c_8\}$. It should be stressed that, it would have been difficult to make assessments of the β-reducts, as described above, without the aid of the developed software (in particular the developed VPRS re-sampling software).

This subsection has shown some of the thought processes the analyst may go through, when selecting a suitable β-reduct for any subsequent aggregation, and the necessity of using all the presented information, across the different analysis results, both statistical and graphical.

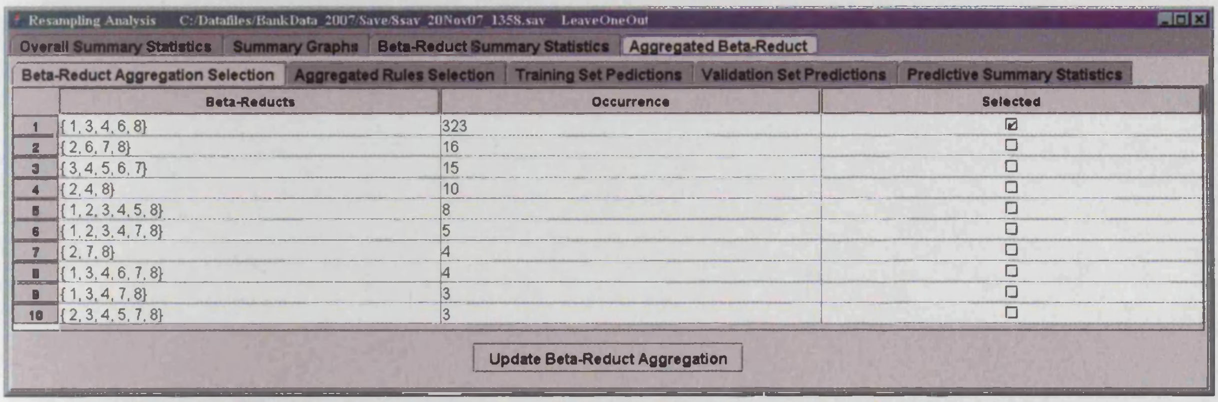
The software and results shown thus far, have had two purposes; firstly to indicate the future predictive performance of any selected β-reduct, in the guise of the predictive accuracy estimates (the out-of- bag estimates); and secondly, to aid the analyst in the subsequent β-reduct aggregation phase (described next). Finally, it is worth re-iterating at this point, that the analyst still retains a high level of autonomy, and it is they who make the final decisions, such as the selection of the β-reduct to aggregate.

8.1.3 β-Reduct Aggregation

This final subsection of the leave-one-out analysis, describes the process of β-reduct aggregation within the developed VPRS re-sampling software. On selecting the 'Aggregated Beta-Reduct' tab, the analyst is initially presented with the 'Beta-Reduct Aggregation Selection' panel, as shown in Figure 8.1.3.1.

This panel allows the analyst to select, using the tick boxes on the right of the table, the β-reduct that they wish to aggregate[24] (aggregating all occurrences of the selected β-reduct as described in Chapter 3 subsection3.5.2). The table shows the number of occurrences of the top ten most frequently selected β-reducts, this information was also represented graphically under the 'Summary Graphs' panel (see Figure 8.1.1.1).

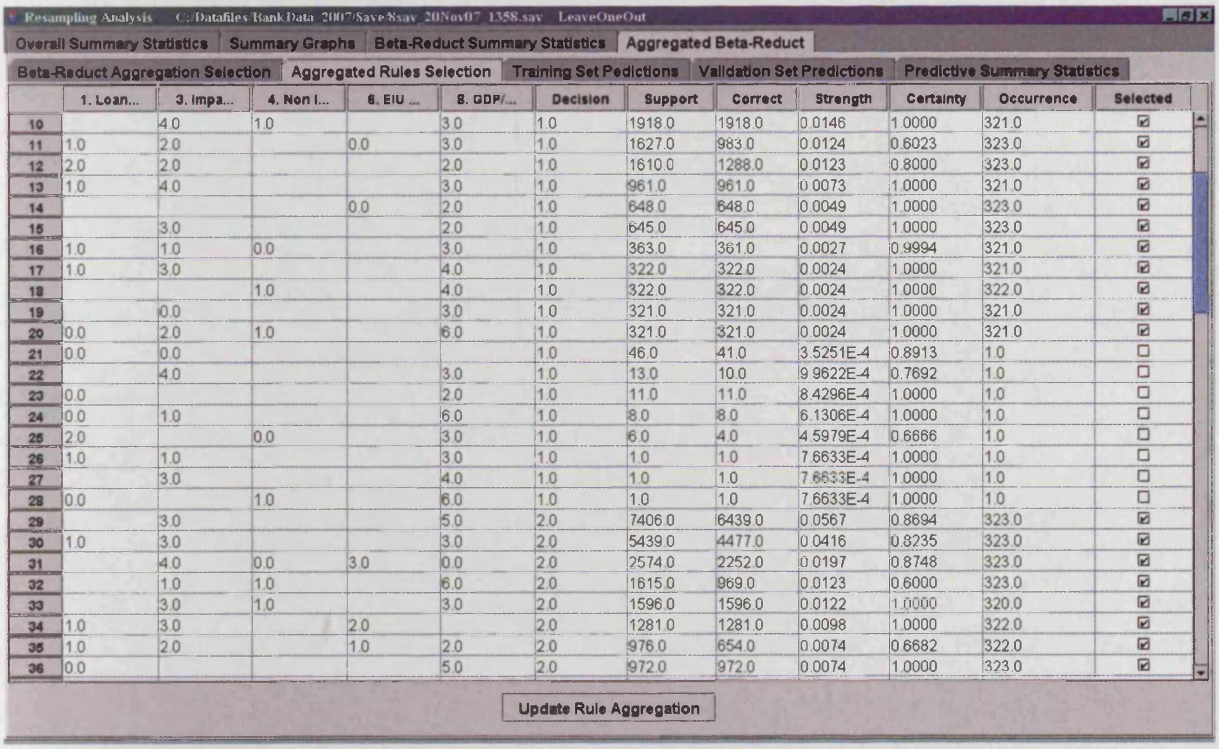
[24] The software does allow the analyst to select more than one β-reduct to aggregate, but the issue of aggregating across β-reducts was not explored within the theoretical conceptualisation of the β-reduct aggregation process in Chapter 3. However, the functionality remains in the software for any future expansion of the theory.


Resampling Analysis C:/Datafiles/BankData_2007/Save/Ssav_20Nov07_1358.sav LeaveOneOut
Overall Summary Statistics
Summary Graphs
Beta-Reduct Summary Statistics
Aggregated Beta-Reduct
Beta-Reduct Aggregation Selection
Aggregated Rules Selection
Training Set Pedictions
Validation Set Predictions
Predictive Summary Statistics
Update Beta-Reduct Aggregation

	Beta-Reducts	Occurrence	Selected
1	{1, 3, 4, 6, 8}	323	☑
2	{2, 6, 7, 8}	16	□
3	{3, 4, 5, 6, 7}	15	□
4	{2, 4, 8}	10	□
5	{1, 2, 3, 4, 5, 8}	8	□
6	{1, 2, 3, 4, 7, 8}	5	□
7	{2, 7, 8}	4	□
8	{1, 3, 4, 6, 7, 8}	4	□
9	{1, 3, 4, 7, 8}	3	□
10	{2, 3, 4, 5, 7, 8}	3	□

Figure 8.1.3.1: 'Beta Reduct Aggregation Selection' Panel Associated with the Leave-one-out Analysis of the FIBR Training Set

Once the analyst has selected the β-reduct for aggregation, aided by the analysis information displayed within the previously described panels, they must click the 'Update Beta-Reduct Aggregation' button seen at the bottom of the panel shown in Figure 8.1.3.1. The rules associated with all occurrences of the selected β-reduct are now aggregated, and the results are shown in the 'Aggregated Rules Selection' panel in Figure 8.1.3.2.


Resampling Analysis C:/Datafiles/BankData_2007/Save/Ssav_20Nov07_1358.sav LeaveOneOut
Overall Summary Statistics
Summary Graphs
Beta-Reduct Summary Statistics
Aggregated Beta-Reduct
Beta-Reduct Aggregation Selection
Aggregated Rules Selection
Training Set Pedictions
Validation Set Predictions
Predictive Summary Statistics
Update Rule Aggregation

	1. Loan...	3. Impa...	4. Non I...	6. EIU ...	8. GDP/...	Decision	Support	Correct	Strength	Certainty	Occurrence	Selected
10		4.0	1.0		3.0	1.0	1918.0	1918.0	0.0146	1.0000	321.0	☑
11	1.0	2.0		0.0	3.0	1.0	1627.0	983.0	0.0124	0.6023	323.0	☑
12	2.0	2.0			2.0	1.0	1610.0	1288.0	0.0123	0.8000	323.0	☑
13	1.0	4.0			3.0	1.0	961.0	961.0	0.0073	1.0000	321.0	☑
14				0.0	2.0	1.0	648.0	648.0	0.0049	1.0000	323.0	☑
15		3.0			2.0	1.0	645.0	645.0	0.0049	1.0000	323.0	☑
16	1.0	1.0	0.0		3.0	1.0	363.0	361.0	0.0027	0.9994	321.0	☑
17	1.0	3.0			4.0	1.0	322.0	322.0	0.0024	1.0000	321.0	☑
18			1.0		4.0	1.0	322.0	322.0	0.0024	1.0000	322.0	☑
19		0.0			3.0	1.0	321.0	321.0	0.0024	1.0000	321.0	☑
20	0.0	2.0	1.0		6.0	1.0	321.0	321.0	0.0024	1.0000	321.0	☑
21	0.0	0.0				1.0	46.0	41.0	3.5251E-4	0.8913	1.0	□
22		4.0			3.0	1.0	13.0	10.0	9.9622E-4	0.7692	1.0	□
23	0.0				2.0	1.0	11.0	11.0	8.4296E-4	1.0000	1.0	□
24	0.0	1.0			6.0	1.0	8.0	8.0	6.1306E-4	1.0000	1.0	□
25	2.0		0.0		3.0	1.0	6.0	4.0	4.5979E-4	0.6666	1.0	□
26	1.0	1.0			3.0	1.0	1.0	1.0	7.6633E-4	1.0000	1.0	□
27		3.0			4.0	1.0	1.0	1.0	7.6633E-4	1.0000	1.0	□
28	0.0		1.0		6.0	1.0	1.0	1.0	7.6633E-4	1.0000	1.0	□
29		3.0			5.0	2.0	7406.0	6439.0	0.0567	0.8694	323.0	☑
30	1.0	3.0			3.0	2.0	5439.0	4477.0	0.0416	0.8235	323.0	☑
31		4.0	0.0	3.0	0.0	2.0	2574.0	2252.0	0.0197	0.8748	323.0	☑
32		1.0	1.0		6.0	2.0	1615.0	969.0	0.0123	0.6000	323.0	☑
33		3.0	1.0		3.0	2.0	1596.0	1596.0	0.0122	1.0000	320.0	☑
34	1.0	3.0		2.0		2.0	1281.0	1281.0	0.0098	1.0000	322.0	☑
35	1.0	2.0		1.0	2.0	2.0	976.0	654.0	0.0074	0.6682	322.0	☑
36	0.0				5.0	2.0	972.0	972.0	0.0074	1.0000	323.0	☑

Figure 8.1.3.2: Example Selection of Aggregated Rules Concomitant with the Aggregated β-reduct $\{c_1, c_3, c_4, c_6, c_8\}$, Associated with the Leave-one-out Analysis of the FIBR Training Set

Figure 8.1.3.2, shows the selection of the aggregated rules, associated with the aggregated β-

reduct $\{c_1, c_3, c_4, c_6, c_8\}$. The analyst has complete autonomy to select the decision rules they want to use with regards to their analysis, using the tick boxes in the column on the right of the panel. Their choice of selection, would be based on the information provided in the previously described panels, or through the information displayed within the 'Aggregated Rules Selection' panel itself (based on strength, certainty values etc.). Figure 8.1.3.2 illustrates an instance where the analyst has not selected all rules capable of predicting the 'B' (1) grade banks (some tick boxes not ticked).

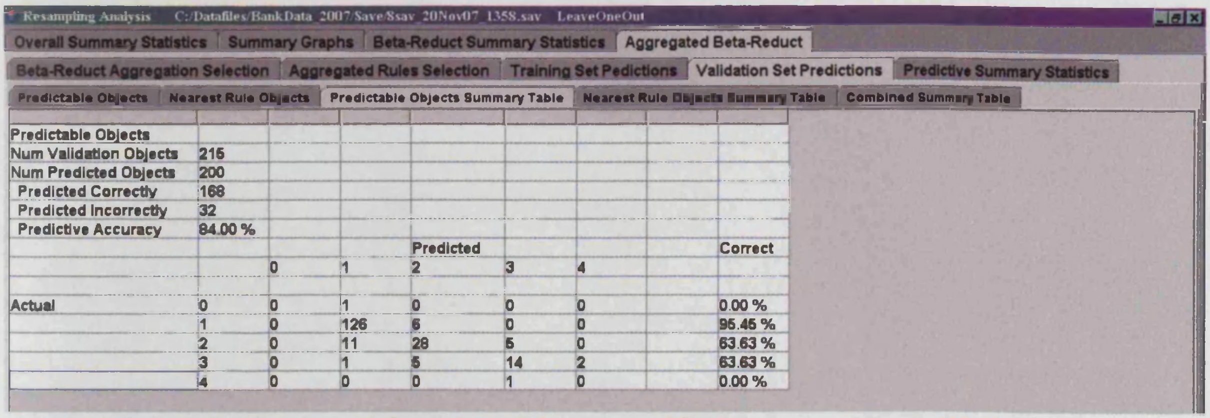
The rules within Figure 8.1.3.2, are ordered by decision class '0' to '4' (grades 'A' to 'E') and sorted within those five decision classes, first by strength (aggregated β-reduct strength, see Chapter 3 Equation 3.5.2.1) and then by certainty (see Equation 3.5.2.2).

Within Figure 8.1.3.2, the number of rules selected (ticked in the boxes), was based on the mean frequency of the rule occurrence over the five decision classes, utilising the decision rule distributions shown previously in Figure 8.1.2.2.1 (means rounded to nearest integer). It follows that, zero rules for predicting 'A' (0) grade banks (none were available), 20 rules for predicting 'B' (1) grade banks (only 11 shown within Figure 8.1.3.2), 27 rules for predicting the 'C' (2) grade banks (only eight shown within Figure 8.1.3.2), 17 rules for predicting 'D' (3) grade banks and three rules for predicting the 'E' (4) grade banks were selected.

Once the analyst has selected the final rule set (ticked boxes within Figure 8.1.3.2), they may proceed to test the predictive accuracy of the selected rules on the concomitant FIBR validation set. The analysis on the validation set now follows the same process (analysis panels), as those shown with regards to the vein graph analysis in Chapter 7, except here, the rules are associated with the aggregated β-reduct rule set, as opposed to a set of rules associated with a single β-reduct selected during a vein graph analysis. Using the same analysis panels/interface as before, gives a level of consistency between the both the re-sampling and vein graph analyses, and allows the analyst to easily compare results.[25]

[25] As stated previously, it is also good programming practise to use a consistent interface and re-use programming code, mitigating the need for multiple development and testing of the software. It also aids a user, if familiar interfaces are used throughout.

Here for brevity, only the results of those banks predictable by matching rules from the validation set are presented, shown below in Figure 8.1.3.3. A more comprehensive exposition of β-reduct aggregation is given in section 8.4. The format of Figure 8.1.3.3, was described previously in Chapter 7 section 7.3).

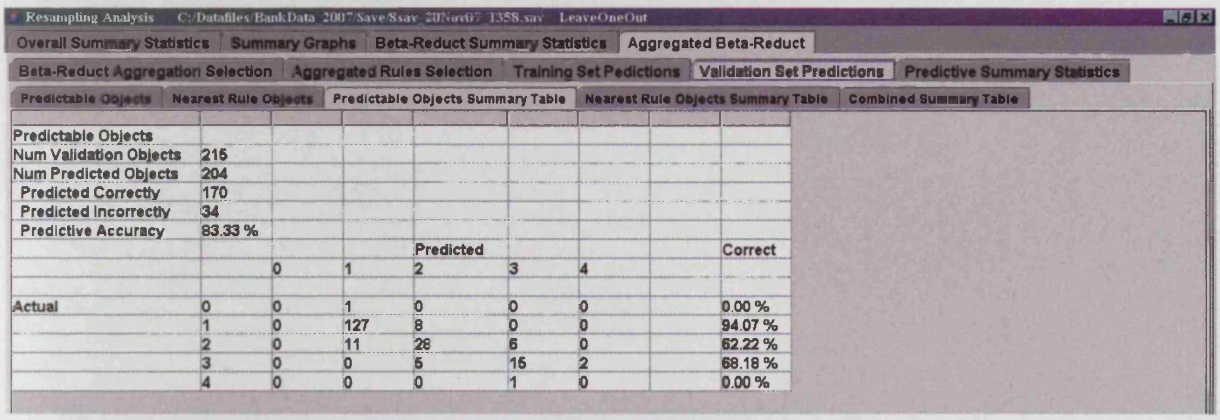

Resampling Analysis C:/Datafiles/BankData_2007/Save/Ssav_20Nov07_1358.sav LeaveOneOut
Overall Summary Statistics
Summary Graphs
Beta-Reduct Summary Statistics
Aggregated Beta-Reduct
Beta-Reduct Aggregation Selection
Aggregated Rules Selection
Training Set Pedictions
Validation Set Predictions
Predictive Summary Statistics
Predictable Objects
Nearest Rule Objects
Predictable Objects Summary Table
Nearest Rule Objects Summary Table
Combined Summary Table

Predictable Objects								
Num Validation Objects	215							
Num Predicted Objects	200							
Predicted Correctly	168							
Predicted Incorrectly	32							
Predictive Accuracy	84.00 %							
				Predicted				Correct
		0	1	2	3	4		
Actual	0	0	1	0	0	0		0.00 %
	1	0	126	6	0	0		95.45 %
	2	0	11	28	5	0		63.63 %
	3	0	1	5	14	2		63.63 %
	4	0	0	0	1	0		0.00 %

Figure 8.1.3.3: Application of the Selected Aggregated Rules Concomitant with the Aggregated β-reduct $\{c_1, c_3, c_4, c_6, c_8\}$, Associated with Leave-one-out and Applied to the FIBR Validation Set

Within Figure 8.1.3.3, 200 out of the 215 banks within the validation set were predictable by a matching rule from the selected aggregated rule set (see second and third rows of the table within Figure 8.1.3.3), with 168 or 84.00% of those banks being predicted correctly (fourth and fifth row). The breakdown of predictive accuracies over the decision classes (the confusion matrix portion of the table), gives exactly the same predictive accuracies as the results of the same β-reduct $\{c_1, c_3, c_4, c_6, c_8\}$ seen previously within the vein graph analysis of the FIBR data (see Chapter 7, Table 7.4.3). Thus, based on these results for this FIBR data, it can be concluded that, aggregation of the β-reducts selected through the leave-one-out analysis does not improve on the predictive accuracy (this is not necessarily the case with regards to bootstrap aggregation described later in sections 8.3 and 8.4).

In addition to looking at the predictive performance, based on selecting the average number of decision rules predicting each decision class (the process described earlier with regards to Figure 8.1.3.2), Figure 8.1.3.4 presents the results based on all the aggregated rules from the aggregated rule panel shown in Figure 8.1.3.2 (i.e. selection of all rules).


Resampling Analysis C:/Datafiles/BankData_2007/Save/Ssav_20Nov07_1358.sav LeaveOneOut
Overall Summary Statistics
Summary Graphs
Beta-Reduct Summary Statistics
Aggregated Beta-Reduct
Beta-Reduct Aggregation Selection
Aggregated Rules Selection
Training Set Pedictions
Validation Set Predictions
Predictive Summary Statistics
Predictable Objects
Nearest Rule Objects
Predictable Objects Summary Table
Nearest Rule Objects Summary Table
Combined Summary Table

Predictable Objects								
Num Validation Objects	215							
Num Predicted Objects	204							
Predicted Correctly	170							
Predicted Incorrectly	34							
Predictive Accuracy	83.33 %							
				Predicted				Correct
		0	1	2	3	4		
Actual	0	0	1	0	0	0		0.00 %
	1	0	127	8	0	0		94.07 %
	2	0	11	28	6	0		62.22 %
	3	0	0	5	15	2		68.18 %
	4	0	0	0	1	0		0.00 %

Figure 8.1.3.4: Application of All Aggregated Rules Concomitant with the Aggregated β-reduct $\{c_1, c_3, c_4, c_6, c_8\}$, Associated with Leave-one-out and Applied to the FIBR Validation Set

When compared to Figure 8.1.3.3, Figure 8.1.3.4 gives four more banks a classification (200 compared to 204 banks, respectively). However there is, a slight decrease in overall predictive accuracy within Figure 8.1.3.4 and specifically the predictive accuracies on the 'B' (1) and 'C' (2) grade banks, when compared to Figure 8.1.3.3. Though there is almost a 5% increase on the 'D' (3) grade banks (68.18% in Figure 8.1.3.4, compared to 63.63% in Figure 8.1.3.3).

The differences between the results shown in Figures 8.1.3.3 and 8.1.3.4, illustrate that, it is possible to obtain, better or worse predictive accuracies based on the validation set through the selection of different sets of aggregated rules from the panel shown in Figure 8.1.3.2. An increase in predictive accuracy on the validation set, could be orchestrated by selecting a specific set of aggregated rules, however, this manipulation of the results would be a naïve approach, as in effect, the analyst would be overfitting their selected aggregated rule set to the validation set. It would be wiser, to find ways of increasing the out-of-bag predictive accuracies, as they are less biased towards a particular predictive accuracy estimate (see Chapter 3, section 3.3, and subsection 3.4.1 for explanation of the out-of-bag estimate).

8.2 *k*-fold Cross-validation, Comparison with Leave-one-out

This section presents the re-sampling results of *k*-fold cross-validation for a number of values of *k* (stratified *k*-fold unless otherwise stated, see Chapter 3 section 3.3.2), and makes comparisons with the results of the leave-one-out analysis shown in the previous section. The graphs (as described in subsection 8.1.1) are used to elucidate the commonality between the results of *k*-fold cross-validation and leave-one-out, and to demonstrate the asymptotic nature between them, as the value of *k* increases. Moreover, it highlights that, the often recommended values for *k*, such as $k = 5$ and $k = 10$ (Efron, 1982; Davison and Hinkley, 1997; Zhang et al., 1999; Dietterich, 2000a) are viewed with regards to this analysis, as relatively inadequate for the purpose of VPRS re-sampling.

Initially, the value of *k* was set to ten (Thomassey and Fiordaliso, 2006). The graph in Figure 8.2.1 shows the frequency of the selected β-reducts (based on the automated selection criteria, described previously), Figure 8.2.2 shows the frequency of the condition attributes associated with the selected β-reducts, and Figure 8.2.3 shows the frequency of selected β-reducts size associated with the selected β-reducts.

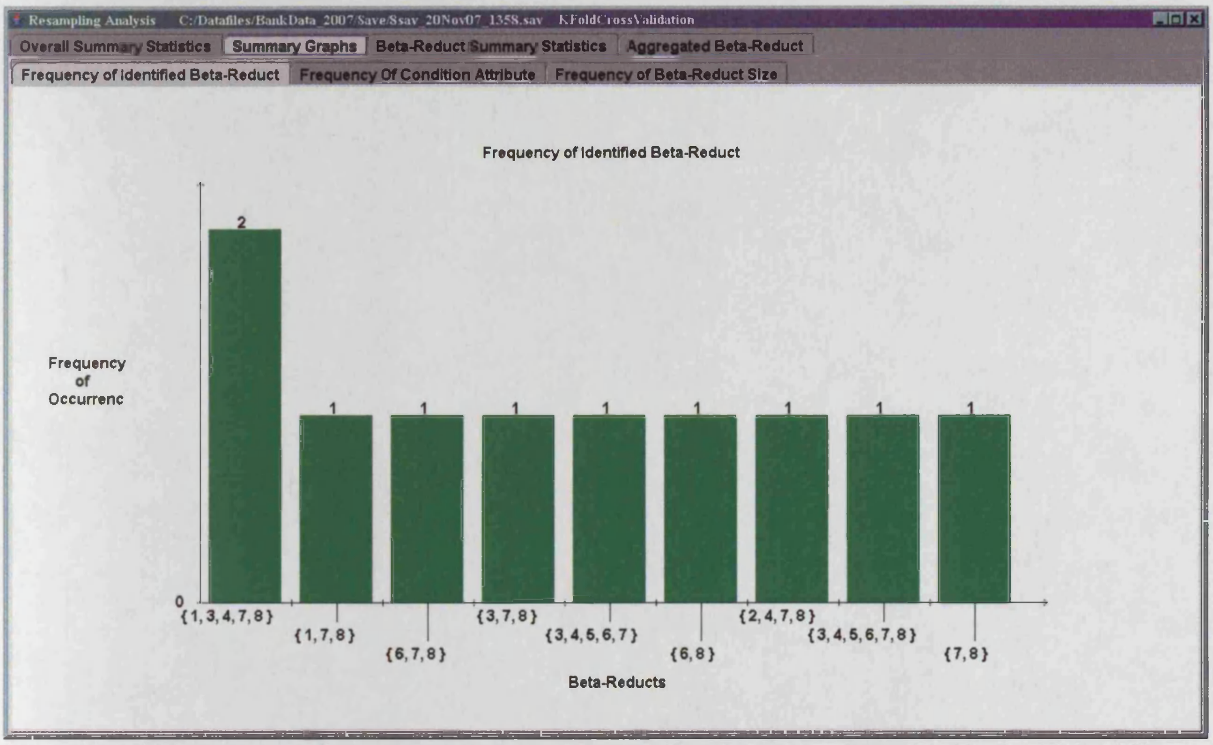

Resampling Analysis C:/Datafiles/BankData_2007/Save/Ssav_20Nov07_1358.sav KFoldCrossValidation
Overall Summary Statistics
Summary Graphs
Beta-Reduct Summary Statistics
Aggregated Beta-Reduct
Frequency of Identified Beta-Reduct
Frequency Of Condition Attribute
Frequency of Beta-Reduct Size
Frequency of Identified Beta-Reduct
Frequency of Occurrenc
0
2
1
1
1
1
1
1
1
1
{1,3,4,7,8}
{1,7,8}
{6,7,8}
{3,7,8}
{3,4,5,6,7}
{6,8}
{2,4,7,8}
{3,4,5,6,7,8}
{7,8}
Beta-Reducts

Figure 8.2.1: Frequency of Occurrence of the Selected β-reducts, Associated with the 10-fold Cross-validation Analysis of the FIBR Training Set

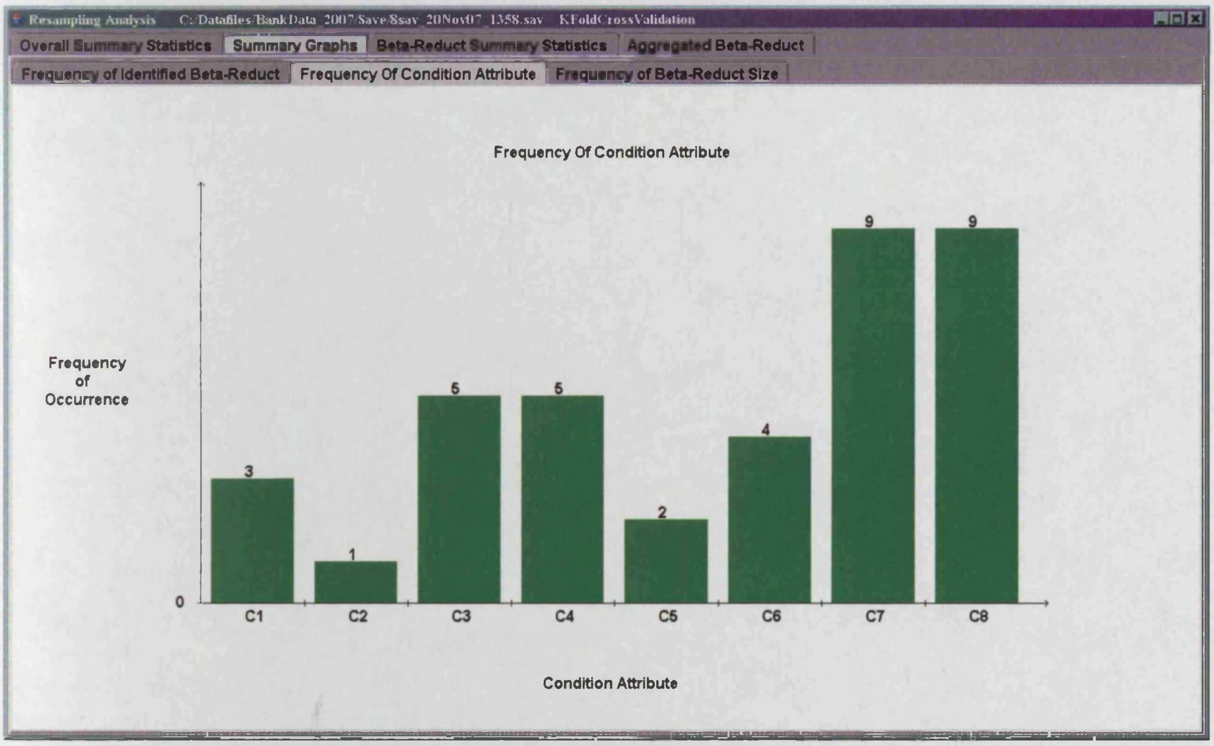

Resampling Analysis C:/Datafiles/BankData_2007/Save/Ssav_20Nov07_1358.sav KFoldCrossValidation
Overall Summary Statistics
Summary Graphs
Beta-Reduct Summary Statistics
Aggregated Beta-Reduct
Frequency of Identified Beta-Reduct
Frequency Of Condition Attribute
Frequency of Beta-Reduct Size
Frequency Of Condition Attribute
Frequency of Occurrence
0
3
1
5
5
2
4
9
9
C1
C2
C3
C4
C5
C6
C7
C8
Condition Attribute

Figure 8.2.2: Frequency of Occurrence of the Condition Attributes Associated with the Selected β-reducts, with Regards to the 10-fold Cross-validation Analysis of the FIBR Training Set

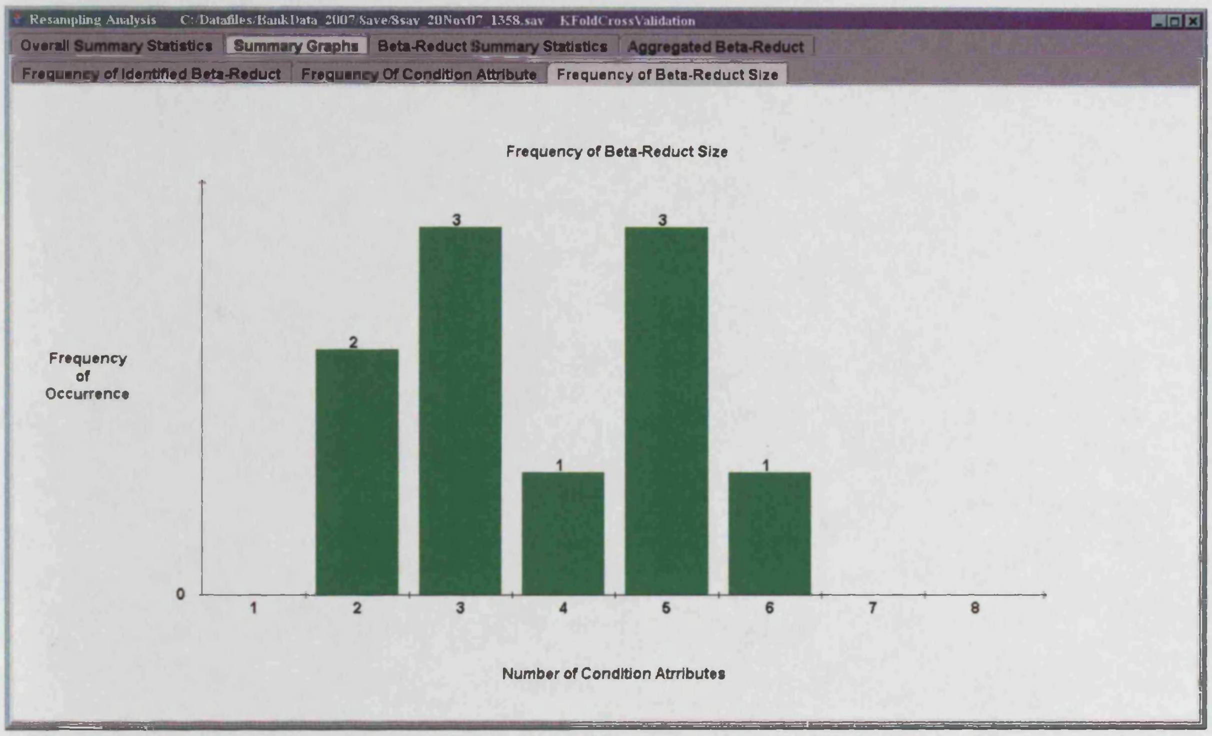


Figure 8.2.3: Frequency of Selected β-reducts Size, Associated with the 10-fold Cross-validation Analysis of the FIBR Training Set

With regards to Figure 8.2.1, clearly, due to the limited number of re-sampling repetitions (i.e. ten), there is no β-reduct showing any significant dominance (frequency of occurrence) over the other selected β-reducts; in contrast to the leave-one-out results shown previously within Figure 8.1.1.1 (which indicated β-reduct $\{c_1, c_3, c_4, c_6, c_8\}$ was significantly dominant). Indeed the impact of using 10-folds (only 10 repetitions), is the limited information and results, which it provides the analyst, to discern between β-reducts for any subsequent β-reduct aggregation analysis.

The graph shown in Figure 8.2.2 does show some consistency with the equivalent leave-one-out graph shown in Figure 8.1.1.2, with the condition attributes c_1, c_3, c_4, c_6, c_7 and c_8, showing some dominance over the other condition attributes, although the relative dominance of c_7 is inconsistent with Figure 8.1.1.2. Figure 8.2.3 appears quite inconsistent with the equivalent leave-one-out graph shown in Figure 8.1.1.2. Although, β-reducts of size five are dominant in Figure 8.1.1.2, this could not be conclusively inferred from Figure 8.2.3, as the result is only based on three of ten repetitions.

It should be reaffirmed, as stated in Chapter 3, one of the advantages of k-fold cross-validation, was a decrease in the overall processing time,[26] with little impact/difference between the predictive accuracies of *k*-fold cross-validation and leave-one-out analysis. Unfortunately here, within the VPRS re-sampling environment, as a number of different β-reducts may be selected during the re-sampling process, a low value of *k* represents an inadequate number of repetitions, leading to inconclusive results. Griffiths and Beynon (2007 and 2008) illustrated a similar finding with regards to the use of 10-fold Cross-validation applied to other data sets, thus indicating, that the inconclusive results asociated with 10-fold Cross-validation, is not unique to the bank data considered here. Hence, the value of *k* was increased to such a point, that the results converged to reflect the results of the leave-one-out analysis more closely.

The graphs within Figures 8.2.4, 8.2.5 and 8.2.6 are based on *k*-fold cross-validation where $k = 50$. For further comparative reference, Appendix A sections A.1 to A.4, illustrate the full set of graphs, elucidating the asymptotic nature of the convergence between the results of *k*-fold cross-validation and leave-one-out analyses, as *k* increases through $k = 10, 20, 30, 40$ and 50.

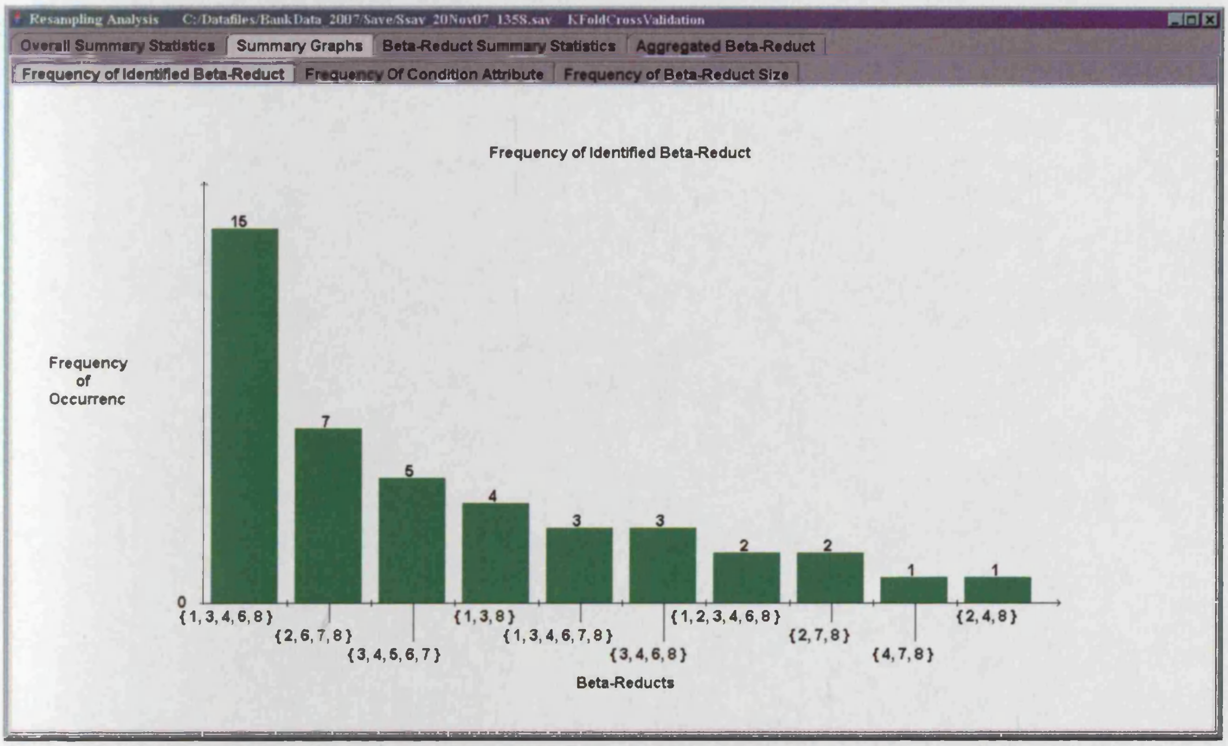

Resampling Analysis C:/Datafiles/BankData 2007/Save/Ssav 20Nov07 1358.sav KFoldCrossValidation
Overall Summary Statistics
Summary Graphs
Beta-Reduct Summary Statistics
Aggregated Beta-Reduct
Frequency of Identified Beta-Reduct
Frequency Of Condition Attribute
Frequency of Beta-Reduct Size
Frequency of Identified Beta-Reduct
Frequency of Occurrenc
0
15
7
5
4
3
3
2
2
1
1
{1,3,4,6,8}
{2,6,7,8}
{3,4,5,6,7}
{1,3,8}
{1,3,4,6,7,8}
{3,4,6,8}
{1,2,3,4,6,8}
{2,7,8}
{4,7,8}
{2,4,8}
Beta-Reducts

Figure 8.2.4: Frequency of Occurrence of the Selected β-reducts, Associated with the 50-fold Cross-validation Analysis of the FIBR Training Set

[26] The other advantage being, reduced variance of the estimated predictive accuracy (see Chapter 3 subsection 3.3.4).

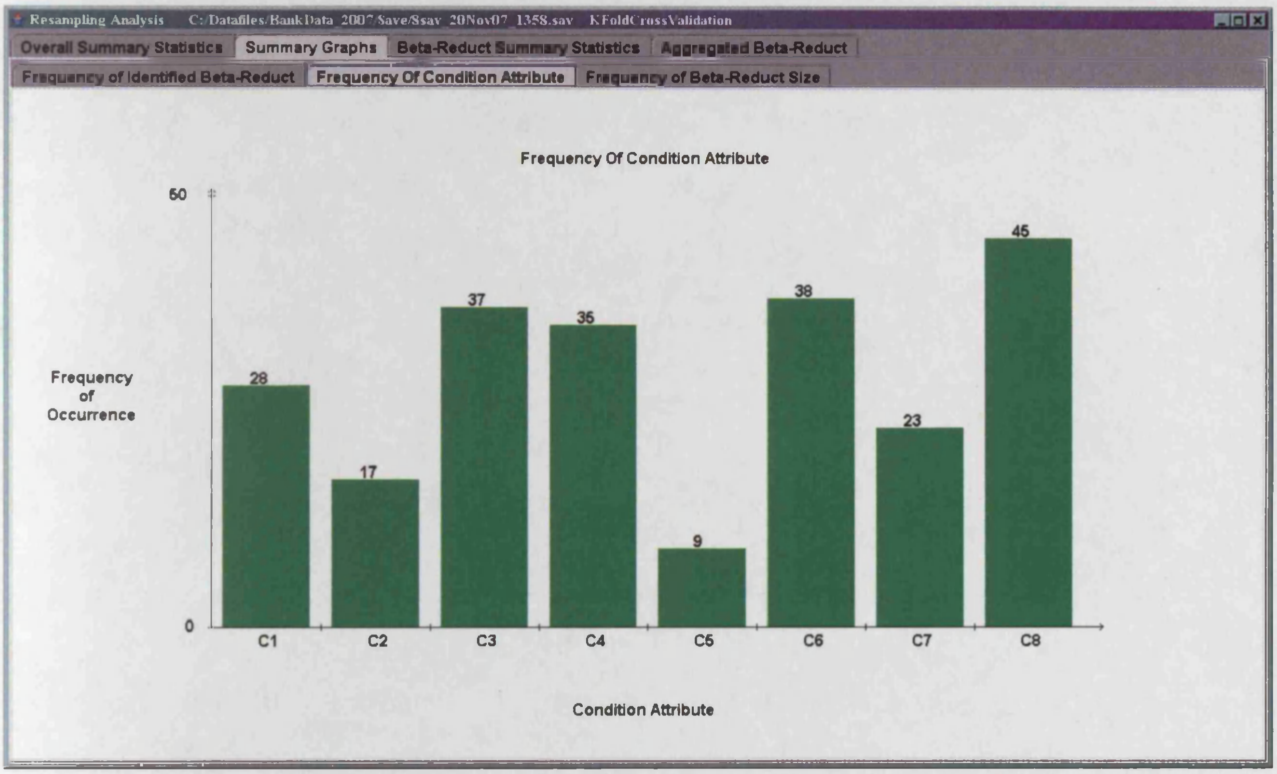

Resampling Analysis C:/Datafiles/BankData_2007/Save/Ssav_20Nov07_1358.sav KFoldCrossValidation
Overall Summary Statistics
Summary Graphs
Beta-Reduct Summary Statistics
Aggregated Beta-Reduct
Frequency of Identified Beta-Reduct
Frequency Of Condition Attribute
Frequency of Beta-Reduct Size
Frequency Of Condition Attribute
50
0
Frequency of Occurrence
28
17
37
35
9
38
23
45
C1
C2
C3
C4
C5
C6
C7
C8
Condition Attribute

Figure 8.2.5: Frequency of Occurrence of the Condition Attributes Associated with the Selected β-reducts, with Regards to the 50-fold Cross-validation Analysis of the FIBR Training Set

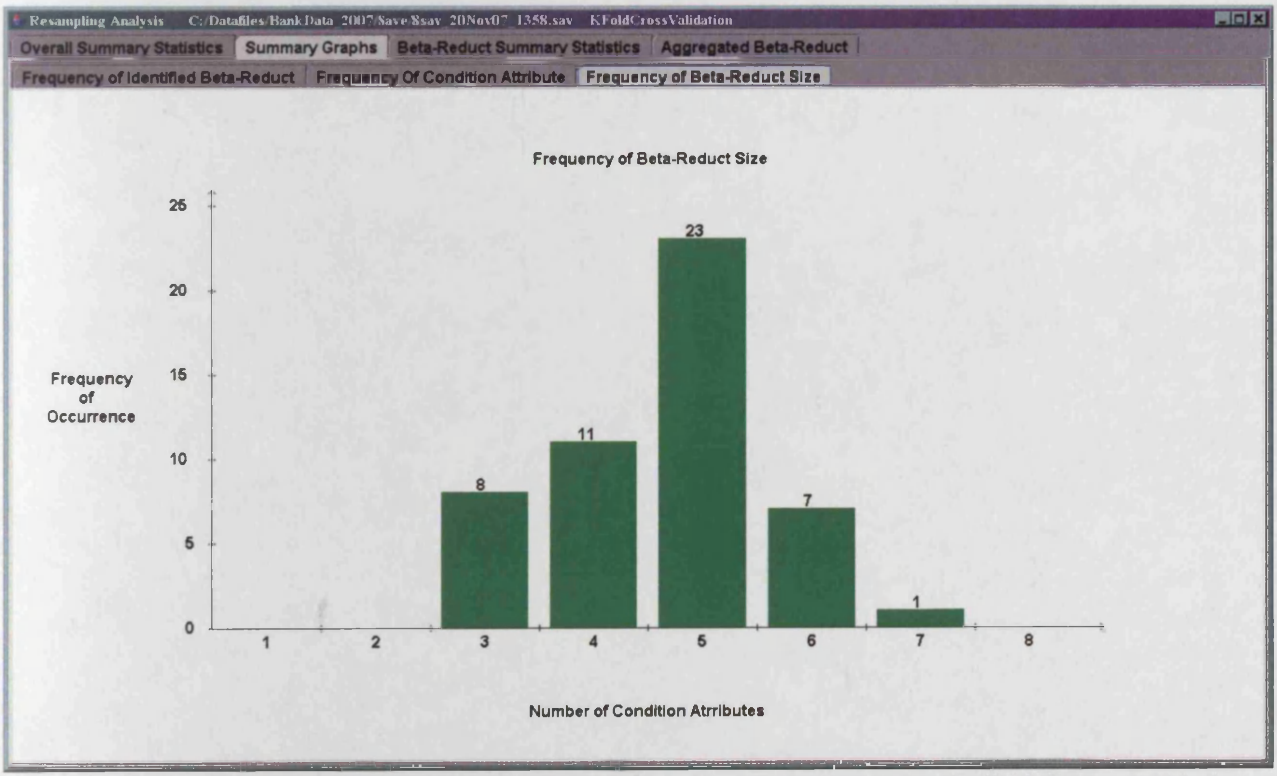

Resampling Analysis C:/Datafiles/BankData_2007/Save/Ssav_20Nov07_1358.sav KFoldCrossValidation
Overall Summary Statistics
Summary Graphs
Beta-Reduct Summary Statistics
Aggregated Beta-Reduct
Frequency of Identified Beta-Reduct
Frequency Of Condition Attribute
Frequency of Beta-Reduct Size
Frequency of Beta-Reduct Size
25
20
15
10
5
0
Frequency of Occurrence
8
11
23
7
1
1
2
3
4
5
6
7
8
Number of Condition Atrributes

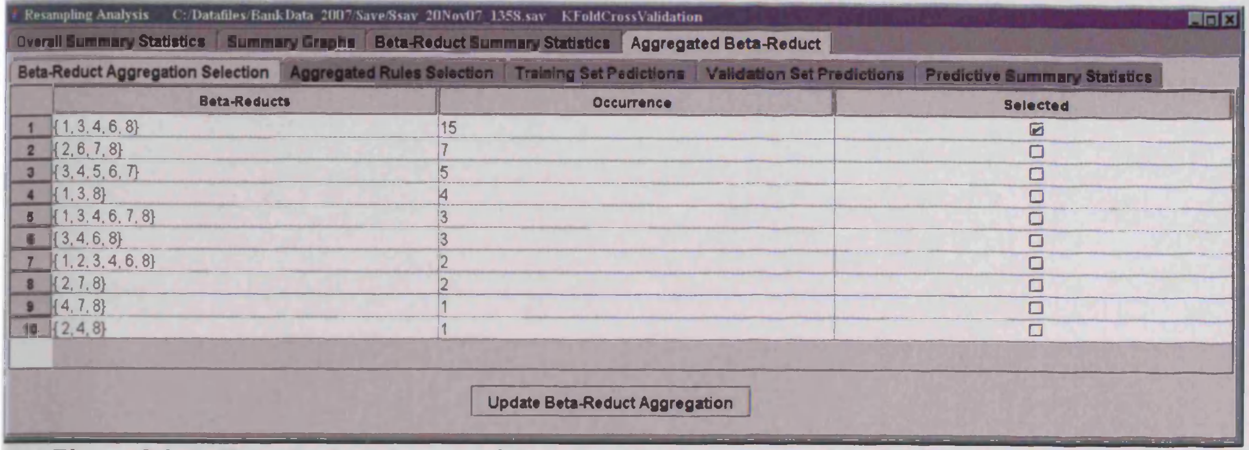
Figure 8.2.6: Frequency of Selected β-reducts Size, Associated with the 50-fold Cross-validation Analysis of the FIBR Training Set

Clearly, with $k = 50$, the graph in Figure 8.2.4 now shows more convergence with the leave-one-out equivalent (see Figure 8.1.1.1) than when k = 10 (see Figure 8.2.1), with the first three most frequently occurring β-reducts being identical in both graphs (in terms of condition attributes, not their frequency of occurrence). More importantly, β-reduct $\{c_1, c_3, c_4, c_6, c_8\}$ in Figure 8.2.4, now appears significantly dominant (compared to Figure 8.2.1). Figure 8.2.5 also shows convergence with its leave-one-out equivalent (see Figure 8.1.1.2), with the condition attributes, c_1, c_3, c_4, c_6 and c_8, displaying the most significance, and c_7 now displaying less significance than in the 10-fold equivalent (see Figure 8.2.2). Additionally, Figure 8.2.6 now demonstrates convergence with its leave-one-out equivalent (see Figure 8.2.2), with β-reducts of size five displaying significant dominance over the other β-reduct sizes.

It may be debatable whether such a high value of k = 50 is warranted, as for lower values of k, the frequency of condition attribute graph (such as previously shown in Figure 8.2.2), could be enough to indicate the most dominant β-reduct (see Appendix A, section A.1 and A.4). This is perhaps a judgement the analyst would have to make. It should be noted though, that k = 50 still represents an eight fold decrease in the processing time when compared to leave-one-out (50 repetitions as opposed to 405).

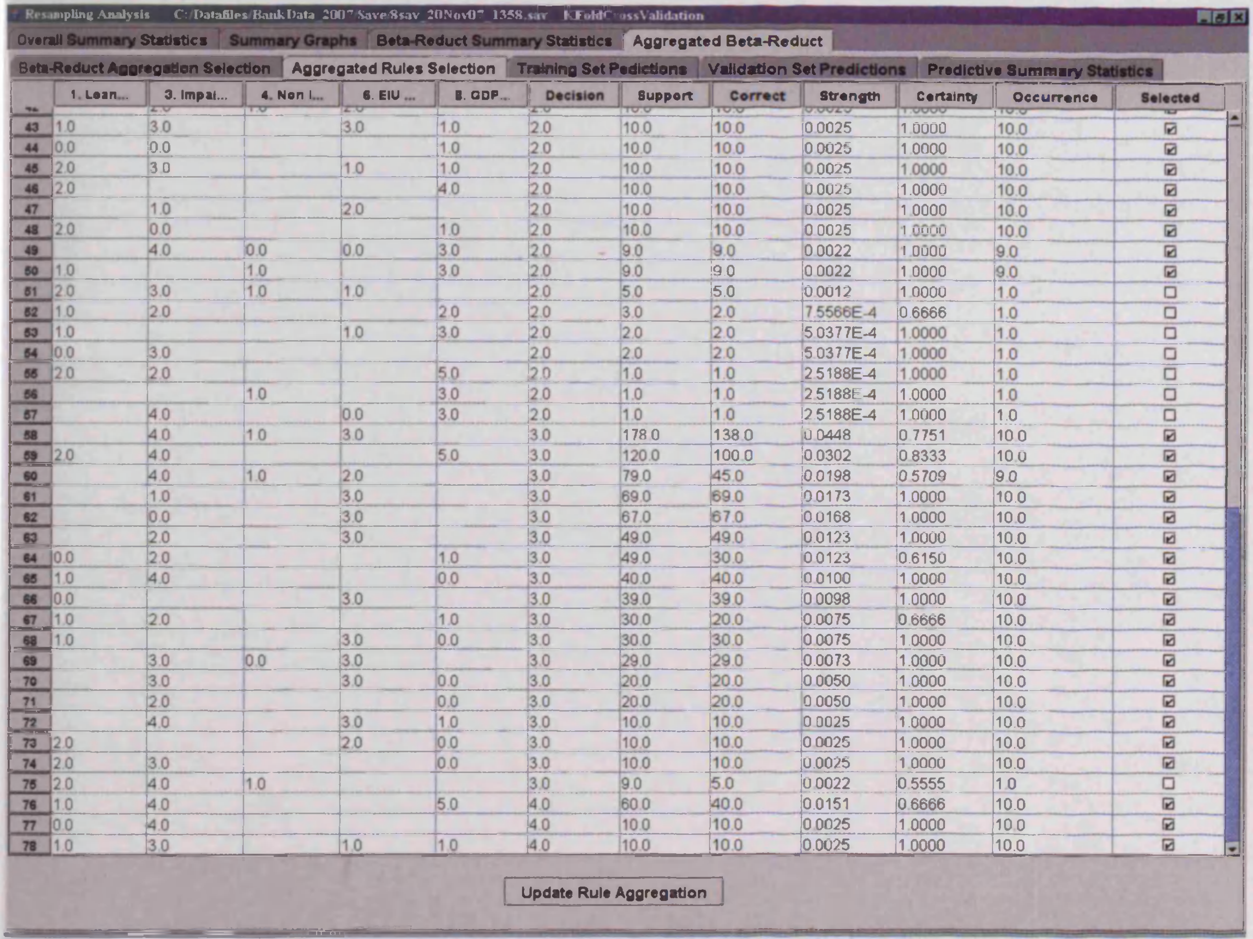
Due to limits on the amount of analyses that can be presented within this dissertation; here, only the results with regards to the stratified k-fold cross-validation are exposited. Appendix sections A.1 to A.4, show the full set of graphs necessary to allow comparisons to be made, based on values of k = 10, 20, 30, 40 and 50, for both the stratified and non-stratified k-fold cross-validation analyses. They illustrate that the results of the stratified k-fold cross-validation converge to the results of the leave-one-out analysis for lower values of k. Hence, vindicating the case for using stratified over non-stratified k-fold cross-validation (Thomassey and Fiordaliso, 2006).

Following on from the results of the graphs shown in Figures 8.2.4 to 8.2.6, the subsequent Figures 8.2.7 to 8.2.9, demonstrate the β-reduct aggregation process and validation set results, based on the 50-fold cross-validation of the FIBR training data set (n.b. stratified k-fold).


Resampling Analysis C:/Datafiles/BankData_2007/Save/8sav_20Nov07_1358.sav KFoldCrossValidation
Overall Summary Statistics
Summary Graphs
Beta-Reduct Summary Statistics
Aggregated Beta-Reduct
Beta-Reduct Aggregation Selection
Aggregated Rules Selection
Training Set Pedictions
Validation Set Predictions
Predictive Summary Statistics
Update Beta-Reduct Aggregation

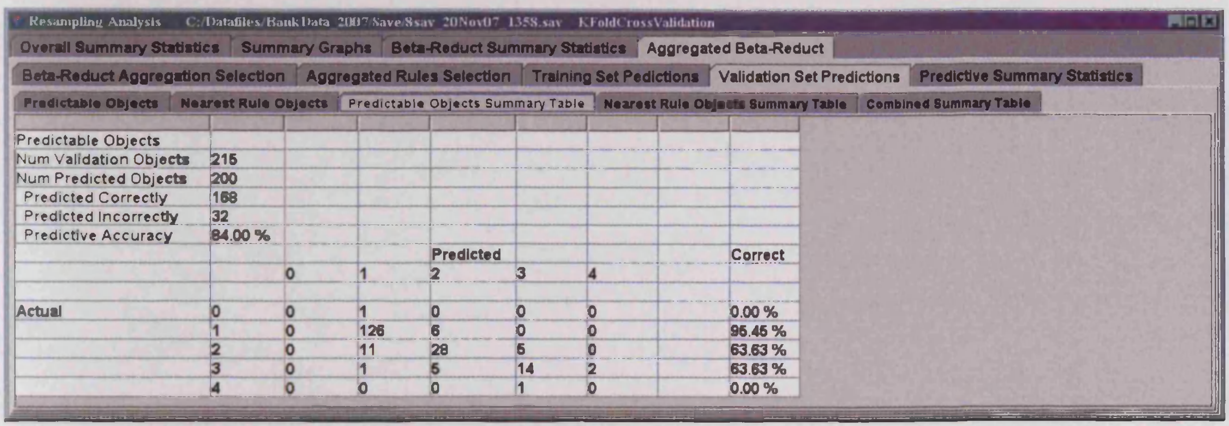
	Beta-Reducts	Occurrence	Selected
1	{1, 3, 4, 6, 8}	15	☑
2	{2, 6, 7, 8}	7	☐
3	{3, 4, 5, 6, 7}	5	☐
4	{1, 3, 8}	4	☐
5	{1, 3, 4, 6, 7, 8}	3	☐
6	{3, 4, 6, 8}	3	☐
7	{1, 2, 3, 4, 6, 8}	2	☐
8	{2, 7, 8}	2	☐
9	{4, 7, 8}	1	☐
10	{2, 4, 8}	1	☐

Figure 8.2.7: 'Beta-Reduct Aggregation Selection' Panel Associated with the 50-fold Cross-validation Analysis of the FIBR Training Set


Resampling Analysis C:/Datafiles/BankData_2007/Save/8sav_20Nov07_1358.sav KFoldCrossValidation
Overall Summary Statistics
Summary Graphs
Beta-Reduct Summary Statistics
Aggregated Beta-Reduct
Beta-Reduct Aggregation Selection
Aggregated Rules Selection
Training Set Pedictions
Validation Set Predictions
Predictive Summary Statistics
Update Rule Aggregation

	1. Lean...	3. Impai...	4. Non I...	6. EIU ...	8. GDP...	Decision	Support	Correct	Strength	Certainty	Occurrence	Selected
43	1.0	3.0		3.0	1.0	2.0	10.0	10.0	0.0025	1.0000	10.0	☑
44	0.0	0.0			1.0	2.0	10.0	10.0	0.0025	1.0000	10.0	☑
45	2.0	3.0		1.0	1.0	2.0	10.0	10.0	0.0025	1.0000	10.0	☑
46	2.0				4.0	2.0	10.0	10.0	0.0025	1.0000	10.0	☑
47		1.0		2.0		2.0	10.0	10.0	0.0025	1.0000	10.0	☑
48	2.0	0.0			1.0	2.0	10.0	10.0	0.0025	1.0000	10.0	☑
49		4.0	0.0	0.0	3.0	2.0	9.0	9.0	0.0022	1.0000	9.0	☑
50	1.0		1.0		3.0	2.0	9.0	9.0	0.0022	1.0000	9.0	☑
51	2.0	3.0	1.0	1.0		2.0	5.0	5.0	0.0012	1.0000	1.0	☐
52	1.0	2.0			2.0	2.0	3.0	2.0	7.5566E-4	0.6666	1.0	☐
53	1.0			1.0	3.0	2.0	2.0	2.0	5.0377E-4	1.0000	1.0	☐
54	0.0	3.0				2.0	2.0	2.0	5.0377E-4	1.0000	1.0	☐
55	2.0	2.0			5.0	2.0	1.0	1.0	2.5188E-4	1.0000	1.0	☐
56			1.0		3.0	2.0	1.0	1.0	2.5188E-4	1.0000	1.0	☐
57		4.0		0.0	3.0	2.0	1.0	1.0	2.5188E-4	1.0000	1.0	☐
58		4.0	1.0	3.0		3.0	178.0	138.0	0.0448	0.7751	10.0	☑
59	2.0	4.0			5.0	3.0	120.0	100.0	0.0302	0.8333	10.0	☑
60		4.0	1.0	2.0		3.0	79.0	45.0	0.0198	0.5709	9.0	☑
61		1.0		3.0		3.0	69.0	69.0	0.0173	1.0000	10.0	☑
62		0.0		3.0		3.0	67.0	67.0	0.0168	1.0000	10.0	☑
63		2.0		3.0		3.0	49.0	49.0	0.0123	1.0000	10.0	☑
64	0.0	2.0			1.0	3.0	49.0	30.0	0.0123	0.6150	10.0	☑
65	1.0	4.0			0.0	3.0	40.0	40.0	0.0100	1.0000	10.0	☑
66	0.0			3.0		3.0	39.0	39.0	0.0098	1.0000	10.0	☑
67	1.0	2.0			1.0	3.0	30.0	20.0	0.0075	0.6666	10.0	☑
68	1.0			3.0	0.0	3.0	30.0	30.0	0.0075	1.0000	10.0	☑
69		3.0	0.0	3.0		3.0	29.0	29.0	0.0073	1.0000	10.0	☑
70		3.0		3.0	0.0	3.0	20.0	20.0	0.0050	1.0000	10.0	☑
71		2.0			0.0	3.0	20.0	20.0	0.0050	1.0000	10.0	☑
72		4.0		3.0	1.0	3.0	10.0	10.0	0.0025	1.0000	10.0	☑
73	2.0			2.0	0.0	3.0	10.0	10.0	0.0025	1.0000	10.0	☑
74	2.0	3.0			0.0	3.0	10.0	10.0	0.0025	1.0000	10.0	☑
75	2.0	4.0	1.0			3.0	9.0	5.0	0.0022	0.5555	1.0	☐
76	1.0	4.0			5.0	4.0	60.0	40.0	0.0151	0.6666	10.0	☑
77	0.0	4.0				4.0	10.0	10.0	0.0025	1.0000	10.0	☑
78	1.0	3.0		1.0	1.0	4.0	10.0	10.0	0.0025	1.0000	10.0	☑

Figure 8.2.8: Selection of Aggregated Rules Concomitant with the Aggregated β-reduct $\{c_1, c_3, c_4, c_6, c_8\}$, Associated with the 50-fold Cross-validation Analysis of the FIBR Training Set


Resampling Analysis C:/Datafiles/Bank Data_2007/Save/8sav_20Nov07_1358.sav KFoldCrossValidation
Overall Summary Statistics
Summary Graphs
Beta-Reduct Summary Statistics
Aggregated Beta-Reduct
Beta-Reduct Aggregation Selection
Aggregated Rules Selection
Training Set Pedictions
Validation Set Predictions
Predictive Summary Statistics
Predictable Objects
Nearest Rule Objects
Predictable Objects Summary Table
Nearest Rule Objects Summary Table
Combined Summary Table

Predictable Objects								
Num Validation Objects	215							
Num Predicted Objects	200							
Predicted Correctly	168							
Predicted Incorrectly	32							
Predictive Accuracy	84.00 %							
				Predicted				Correct
		0	1	2	3	4		
Actual	0	0	1	0	0	0		0.00 %
	1	0	126	6	0	0		95.45 %
	2	0	11	28	5	0		63.63 %
	3	0	1	5	14	2		63.63 %
	4	0	0	0	1	0		0.00 %

Figure 8.2.9: Application of the Aggregated Rules Concomitant with the Aggregated β-reduct $\{c_1, c_3, c_4, c_6, c_8\}$, Associated with 50-fold Cross-validation and Applied to the FIBR Validation Set

Figure 8.2.7 illustrates the selection of β-reduct $\{c_1, c_3, c_4, c_6, c_8\}$ for the purpose of aggregation, and Figure 8.2.8 presents the selection of the associated aggregated rule set, which is applied to the FIBR validation set shown in Figure 8.2.9.

The rules selected within Figure 8.2.8 (ticked boxes), are selected, based on the same principle used previously, with regards to the leave-one-out analysis. That is, utilising the mean frequency of the rule occurrence over the five decision classes (using the *k*-fold cross-validation results equivalent to that shown previously in Figure 8.1.2.2.1, and means rounded to nearest integer). It follows that the selection consisted of, zero rules predicting 'A' (0) grade banks (none were available), 20 rules for predicting 'B' (1) grade banks, 27 rules for predicting the 'C' (2) grade banks (only 8 shown within Figure 8.2.8), 17 rules for predicting 'D' (3) grade banks (selection of all 17 shown within Figure 8.2.8) and three rules for predicting the 'E' (4) grade banks were selected.

It should be noted that, the average (mean) number of rules selected here, with regards to each decision class (grade), for the aggregated β-reduct $\{c_1, c_3, c_4, c_6, c_8\}$, is identical to the leave-one-out analysis discussed previously in subsection 8.1.3. Moreover, the results of applying the selected aggregated rule set to the validation set, shown in Figure 8.2.9, are also identical to those results shown previous with regards to the leave-one-out analysis (subsection 8.1.3 Figure 8.1.3.3). Thus these results further emphasise the asymptotic convergence, between the results of the *k*-fold cross-validation and leave-one-out analyses. However, there is a difference between the distribution of the occurrence values shown in the eleventh columns of Figures 8.2.8 and 8.1.3.2, with Figure 8.2.8 showing more range of values.[27]

The main difference however, between *k*-fold cross-validation and leave-one-out, can be seen within the predictive accuracy estimates (out-of-bag estimates), where, *k*-fold cross-validation demonstrates similar predictive accuracies as leave-one-out but are associated with less variability (standard deviation), as may be expected according to the theory presented within Chapter 3. A comparison of these predictive accuracy estimates across all three re-sampling methods are presented within section 8.4.

This section has demonstrated the functional necessity of the VPRS re-sampling graphs utilised within the developed software (in terms of *k*-fold cross-validation). They allow the analyst to recognise whether results may be conclusive or inconclusive, depending on, the occurrence of the selected β-reducts and their concomitant condition attributes. It has also demonstrated that, there may be a threshold value of *k*, below which, the results are not conclusive enough to aid the analyst with their decision making.

Finally, finding an appropriate value of *k* with regards to VPRS re-sampling, has highlighted an issue which would not have been considered prior to this investigation. This is reflected within the graphs presented in Appendix A sections A.1 to A.4, which illustrate that *k*-fold cross-validation, does not converge to the graphs shown for the leave-one-analysis, until a relatively high value of *k* is set. It should be noted though, from experimental work carried out during the development of the VPRS re-sampling software, that the most appropriate value of *k* is dependant on the specific data set, see for example Griffiths and Beynon (2007). These findings and conclusions, have only been made possible through the analysis tools implemented within the developed VPRS re-sampling

[27] Note that Figures 8.2.8 and 8.3.2 (shown later with regards to the bootstrapping analysis), were added to this dissertation after the respective 50-fold cross-validation and bootstrap analyses were undertaken, and results recorded. As such, and due to the random nature inherent in both re-sampling methods, it was impossible to obtain completely the same results as those found previously. That is, values such as the 'Occurrence' value may not be reflected in the respective analyses (e.g. within the graphs). However, Figures 8.2.8 and 8.3.2 are similar to their respective analyses in other respects (range of 'Occurrence' values, number of associated rules etc.), and have been used here, mainly for example purposes.

software.

8.3 Bootstrap Re-sampling and Bootstrap β-reduct Aggregation

This section illustrates that, unlike *k*-fold cross-validation, the results of bootstrapping within the developed VPRS re-sampling software applied to the FIBR data set, are markedly different to the results of leave-one-out. This section investigates the bootstrapping results of the FIBR data, and the aggregation of the respective β-reducts. The following section 8.4 compares the full set of results of the bootstrap aggregated β-reducts, to the leave-one-out aggregated β-reducts.

Here, since the number of bootstrap repetitions is dependent on a decision made by the analyst, 1,000 bootstraps were undertaken and β-reducts were selected based on the same automated selection criteria as outlined in section 8.1. It was decided to use 1,000 bootstraps, based on the discussion given in Chapter 3 subsection 3.4.1, processing time considerations, and through experimenting with different bootstrap values, whilst trying to obtain consistent results (consistent occurrence of selected β-reducts between bootstrap analyses); Brownstone and Valletta (2001), found their bootstrapping analyses also required up to 1,000 bootstraps.

Looking initially at the distribution of the top ten most frequently selected β-reducts, shown in Figure 8.3.1. Firstly it is interesting, that the top ten most frequently selected β-reducts with regards to bootstrapping, are different to those shown earlier with respect to leave-one-out (see Figure 8.1.1.1).

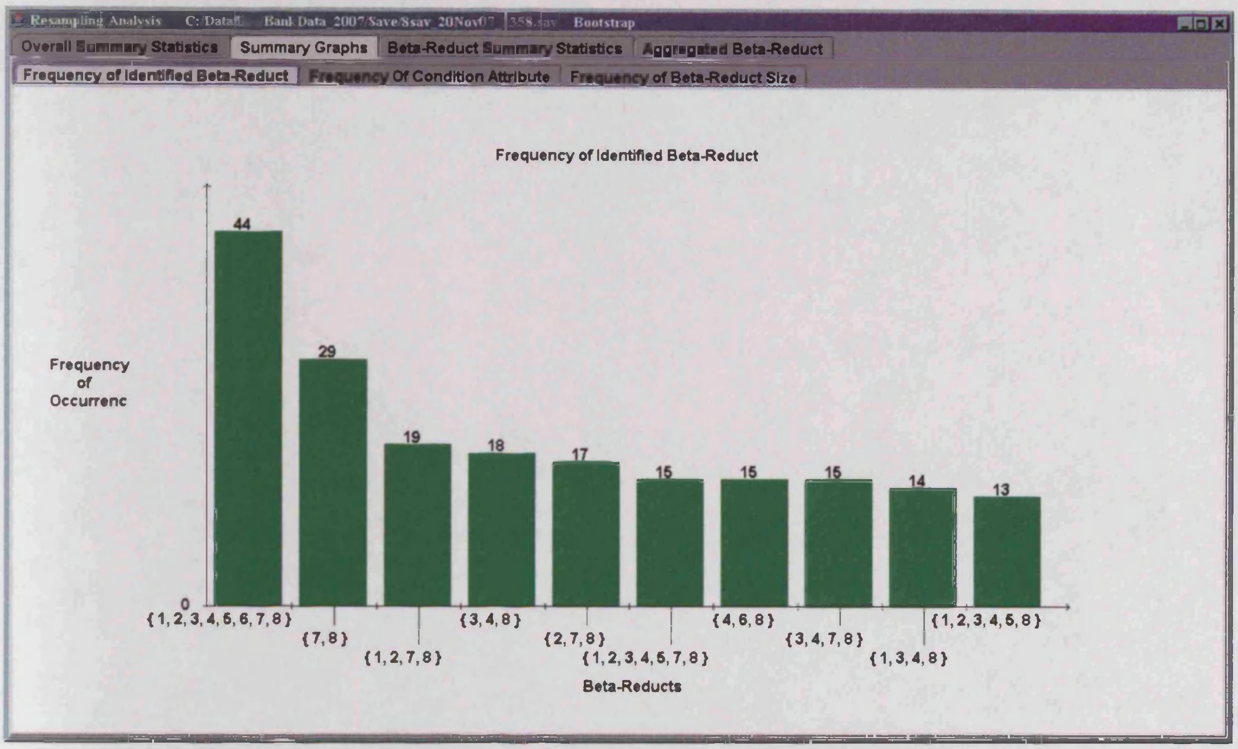


Figure 8.3.1: Frequency of Occurrence of the Selected β-reducts, Associated with the Bootstrap Analysis of the FIBR Training Set

Compared to those distributions shown previously, with regards to leave-one-out (see Figure 8.1.1.1) and *k*-fold cross-validation (see Figure 8.2.4), the distribution of the selected β-reducts shown in Figure 8.1.1.1, is less biased towards a single β-reduct. However, the β-reduct $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$ does appear to be the most dominant, and this is reflected with regards to the distribution of occurrence of the condition attributes associated with all selected β-reducts, shown within Figure 8.3.2.

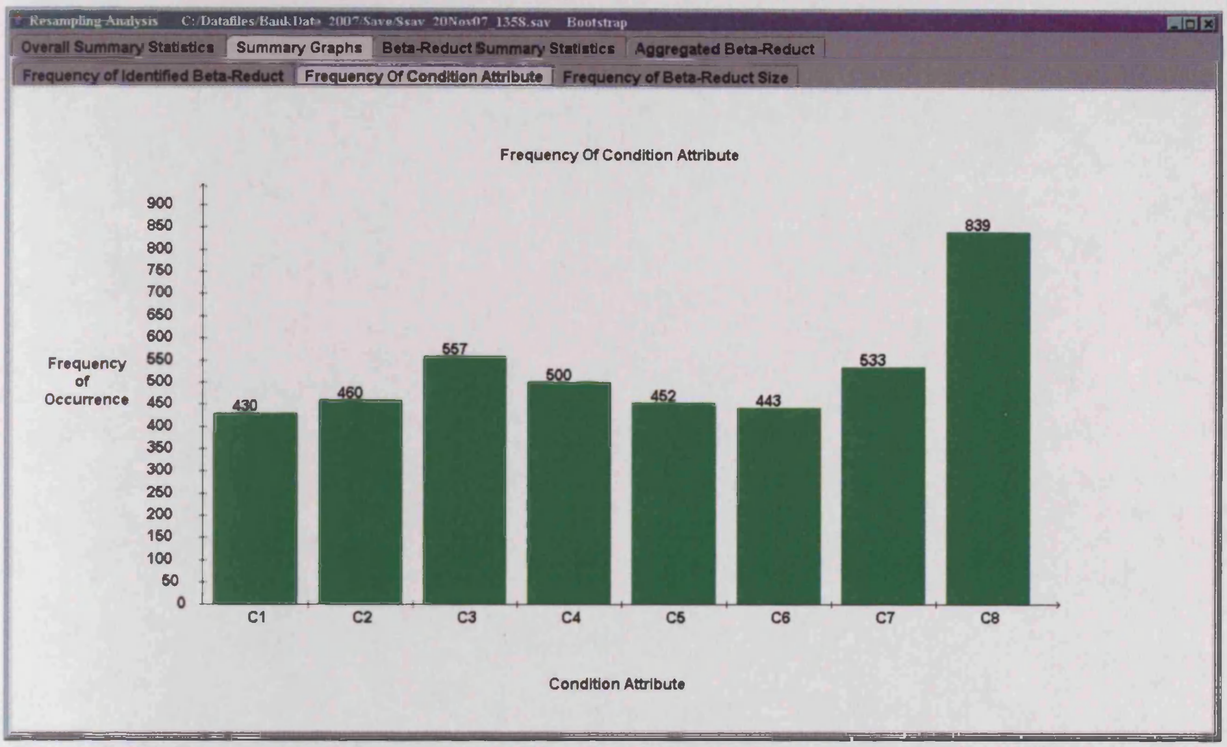


Figure 8.3.2: Frequency of Occurrence, of the Condition Attributes Associated with the Selected β-reducts, with Regards to the Bootstrap Analysis of the FIBR Training Set

Considering the β-reduct $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$ within Figure 8.3.1, it appears to be most dominant β-reduct (though less obviously so, than the most dominant β-reduct with regards to the leave-one-out analysis and 50-fold cross-validation analyses), and even though Figure 8.3.2 appears to support this fact, it can be shown these results do not express the information as fully as one may initially think. That is, by looking at the graphical distribution of the selected β-reduct sizes (number of condition attributes), shown in Figure 8.3.3, and the overall summary statistics shown in Figure 8.3.4, it can be seen that the average (mean) number of condition attributes associated with the selected β-reducts is closer to four than eight (note the most dominant β-reduct $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$ has eight condition attributes).

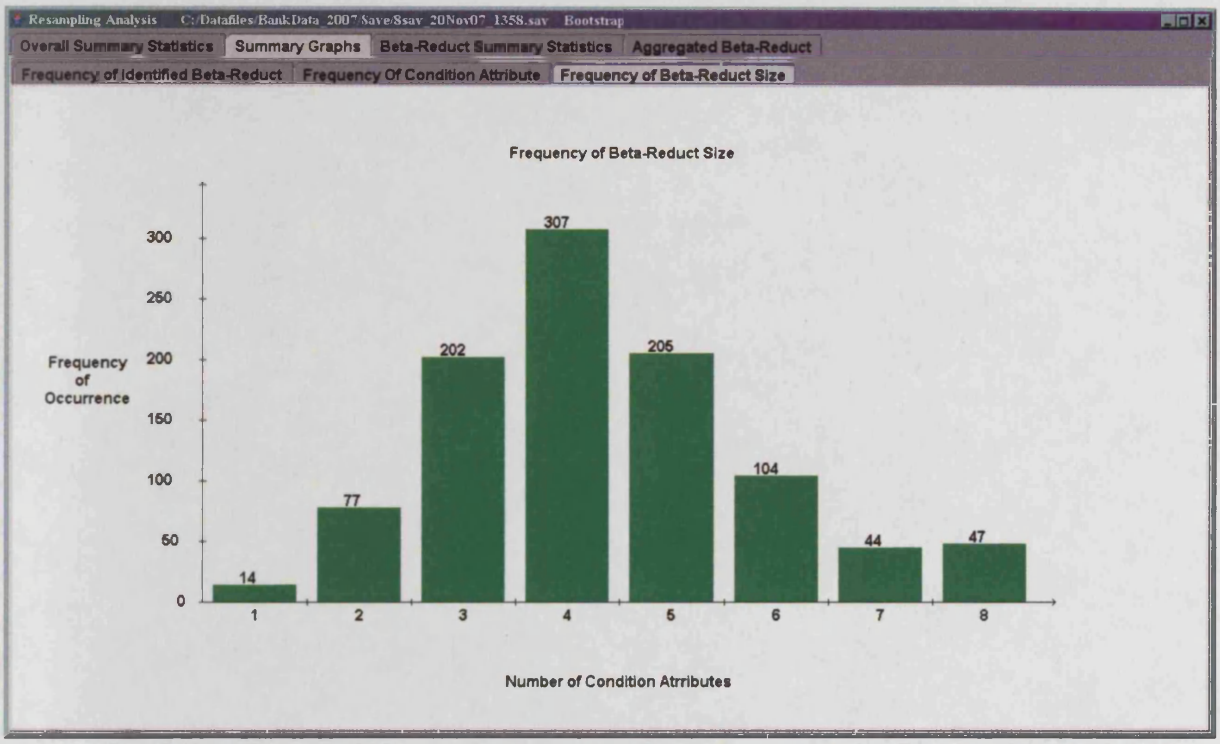
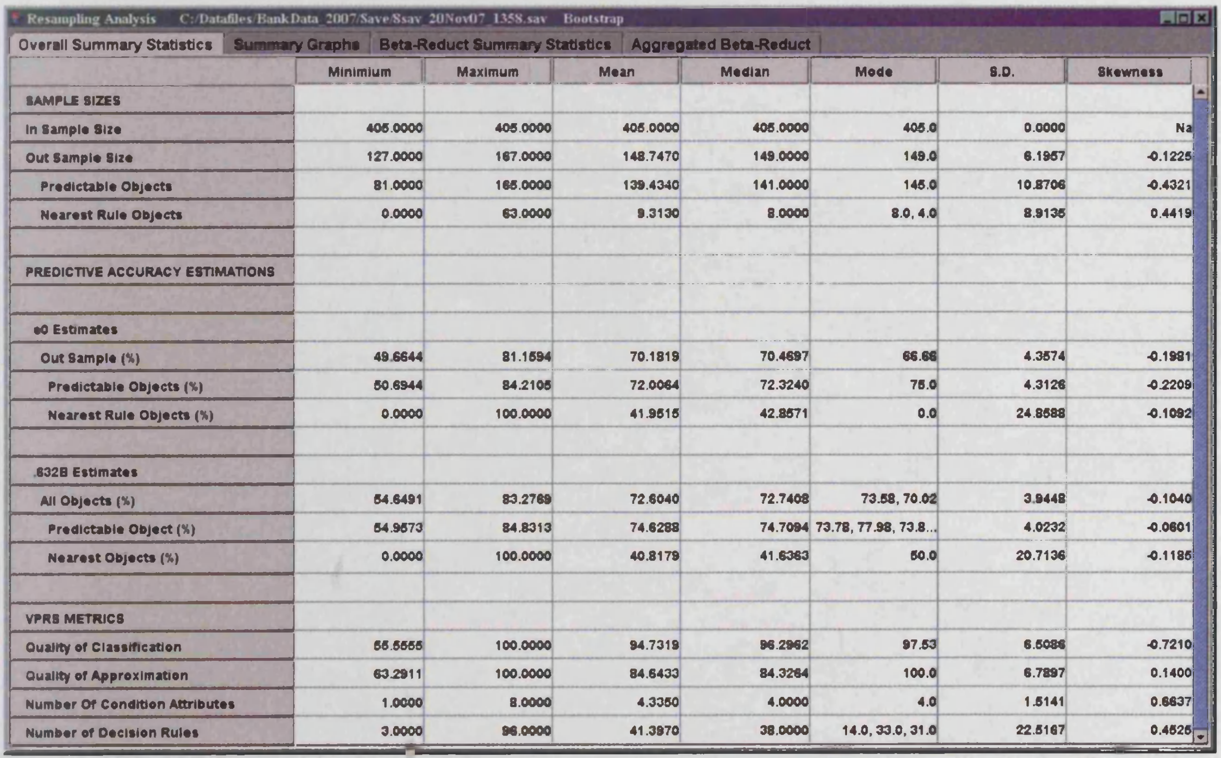


Figure 8.3.3: Frequency of Selected β-reducts Size, Associated with the Bootstrap Analysis of the FIBR Training Set


Resampling Analysis C:/Datafiles/BankData_2007/Save/8sav_20Nov07_1358.sav Bootstrap
Overall Summary Statistics
Summary Graphs
Beta-Reduct Summary Statistics
Aggregated Beta-Reduct

	Minimium	Maximum	Mean	Median	Mode	S.D.	Skewness
SAMPLE SIZES							
In Sample Size	405.0000	405.0000	405.0000	405.0000	405.0	0.0000	Na
Out Sample Size	127.0000	167.0000	148.7470	149.0000	149.0	6.1957	-0.1225
Predictable Objects	81.0000	165.0000	139.4340	141.0000	145.0	10.8706	-0.4321
Nearest Rule Objects	0.0000	63.0000	9.3130	8.0000	8.0, 4.0	8.9135	0.4419
PREDICTIVE ACCURACY ESTIMATIONS							
e0 Estimates							
Out Sample (%)	49.6644	81.1594	70.1819	70.4697	66.66	4.3574	-0.1981
Predictable Objects (%)	50.6944	84.2105	72.0064	72.3240	75.0	4.3126	-0.2209
Nearest Rule Objects (%)	0.0000	100.0000	41.9515	42.8571	0.0	24.8588	-0.1092
.632B Estimates							
All Objects (%)	54.6491	83.2769	72.6040	72.7408	73.58, 70.02	3.9448	-0.1040
Predictable Object (%)	54.9573	84.8313	74.6288	74.7094	73.78, 77.98, 73.8...	4.0232	-0.0601
Nearest Objects (%)	0.0000	100.0000	40.8179	41.6363	50.0	20.7136	-0.1185
VPRS METRICS							
Quality of Classification	55.5555	100.0000	94.7319	96.2962	97.53	6.5086	-0.7210
Quality of Approximation	63.2911	100.0000	84.6433	84.3264	100.0	6.7897	0.1400
Number Of Condition Attributes	1.0000	8.0000	4.3350	4.0000	4.0	1.5141	0.6637
Number of Decision Rules	3.0000	96.0000	41.3970	38.0000	14.0, 33.0, 31.0	22.5167	0.4526

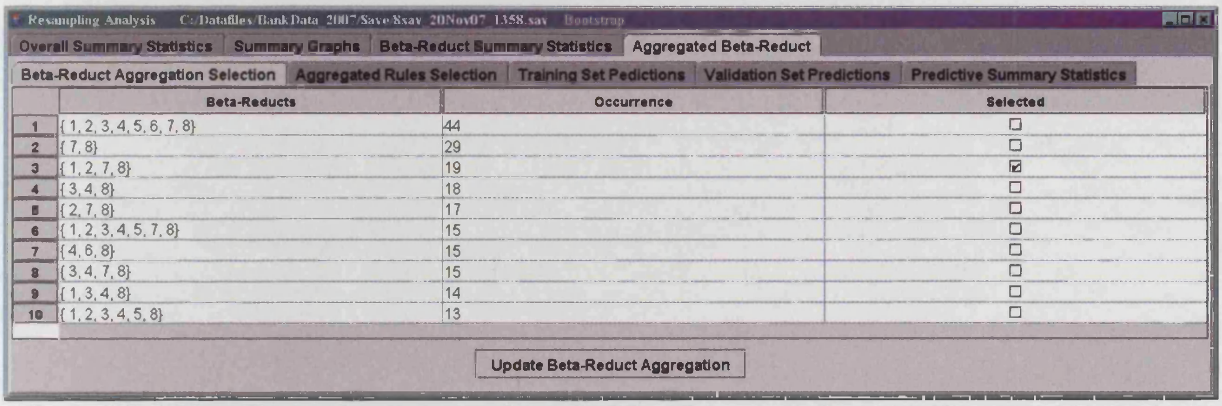
Figure 8.3.4: VPRS Metric Summary Statistics Associated with the Bootstrap Analysis of the FIBR Training Data, Illustrating that the Average Number of Condition Attributes Associated with the Selected β-reduct is Closer to Four than Eight

Additionally, Figure 8.3.3 shows that, β-reducts of size eight, that is the β-reduct $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$, are only selected by the automated selection criteria 47 out of the 1,000 bootstrap repetitions, whereas β-reducts of size four are selected 307 out of the 1,000 repetitions. Hence the analyst is presented with a choice between aggregating over the most frequently selected β-reduct $\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$, or selecting a β-reduct associated with (or nearer to) the most frequent size of β-reduct, i.e. four condition attributes in this case.

Interestingly, on closer inspection of the previously considered Figure 8.3.2, the top four most frequently occurring condition attributes c_3, c_4, c_7 and c_8 are those which appear in the β-reduct $\{c_3, c_4, c_7, c_8\}$, which was the joint sixth most frequently selected β-reduct in Figure 8.3.1. It should be noted that, the distribution associated with Figure 8.3.2 is slightly misleading because of the dominance of condition attribute c_8. To explain further, considering the second, third and forth most significant condition attributes, c_3, c_7 and c_4, respectively, they occur on, at least 48 more occasions than the next most significantly occurring condition attribute c_5 (c_4 occurs on 500 occasions, whereas c_5 occurs on 452 occasions). This difference in occurrence, would have appeared much more significant with regards to the magnitude of values within the graphs shown previously, associated with the leave-one-out and with 50-fold cross-validation analyses (more so with 50-fold cross-validation), see Figures 8.1.1.2 and 8.2.5, respectively.

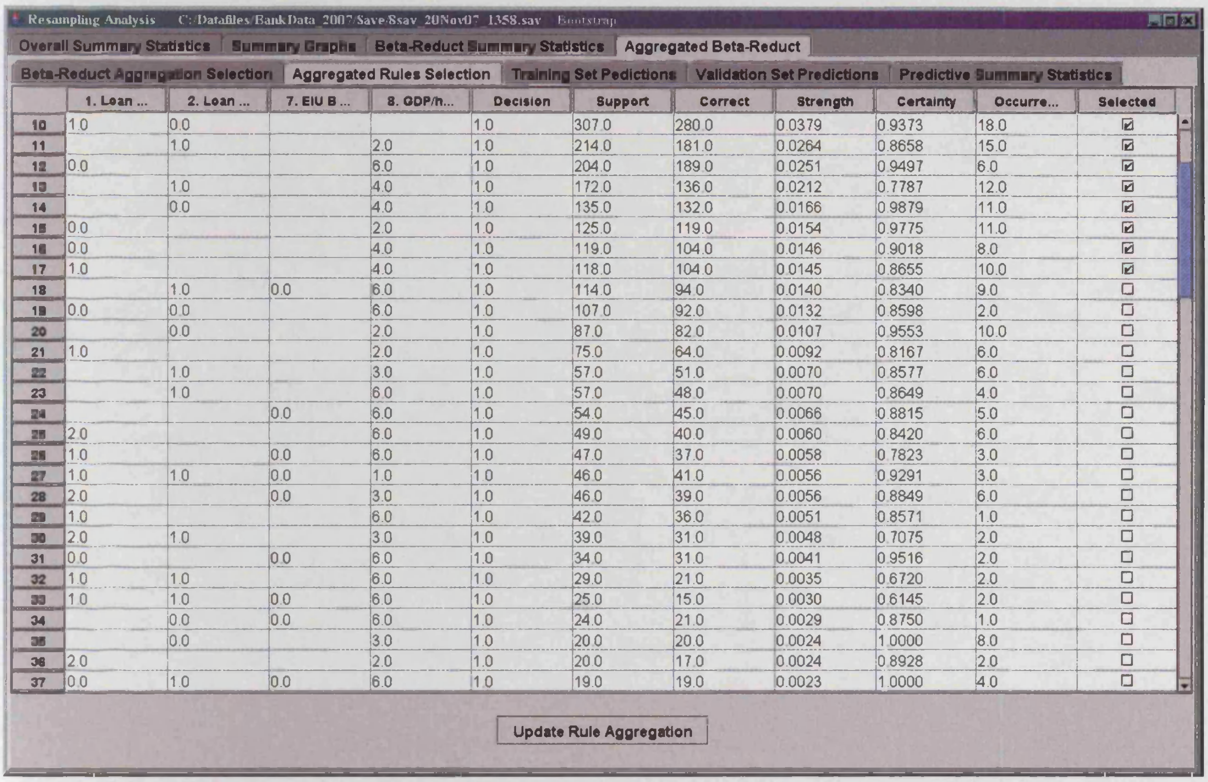
Hence, with regards to the selection of a β-reduct when employing VPRS bootstrap aggregation, the analyst can be guided into, potentially making a better choice, through considering the full spectrum of evidence presented within the developed VPRS re-sampling software (not just considering the graph in Figure 8.3.1). Moreover, the next section will demonstrate, that the estimated predictive accuracies based on the most frequently selected β-reduct, are not necessarily the best (for both the out-of-bag estimates and the results from applying the aggregated β-reduct to the respective validation set), and the QoC associated with the most frequently selected β-reduct is also markedly lower, than when compared to the less frequently occurring β-reducts (again for both the out-of-bag estimates and the validation set). Again it should be emphasised that, these findings and conclusions could not have been made without utilising the developed software.

In keeping with the order of analysis, with regards to leave-one-out and the *k*-fold cross-validation analyses, the final part of this section presents in Figures 8.3.5 to 8.3.7, the β-reduct aggregation of the selected β-reduct $\{c_1, c_2, c_7, c_8\}$, and the results of the associated aggregated rule set applied to the FIBR validation set. The β-reduct $\{c_1, c_2, c_7, c_8\}$, was selected considering the information described above, that is, β-reduct $\{c_1, c_2, c_7, c_8\}$ is the most frequently occurring β-reduct associated with four condition attributes (the most frequent size of selected β-reduct).


Resampling Analysis   C:/Datafiles/Bank Data 2007/Save/Ssav 20Nov07 1358.sav   Bootstrap
Overall Summary Statistics
Summary Graphs
Beta-Reduct Summary Statistics
Aggregated Beta-Reduct
Beta-Reduct Aggregation Selection
Aggregated Rules Selection
Training Set Pedictions
Validation Set Predictions
Predictive Summary Statistics
Update Beta-Reduct Aggregation

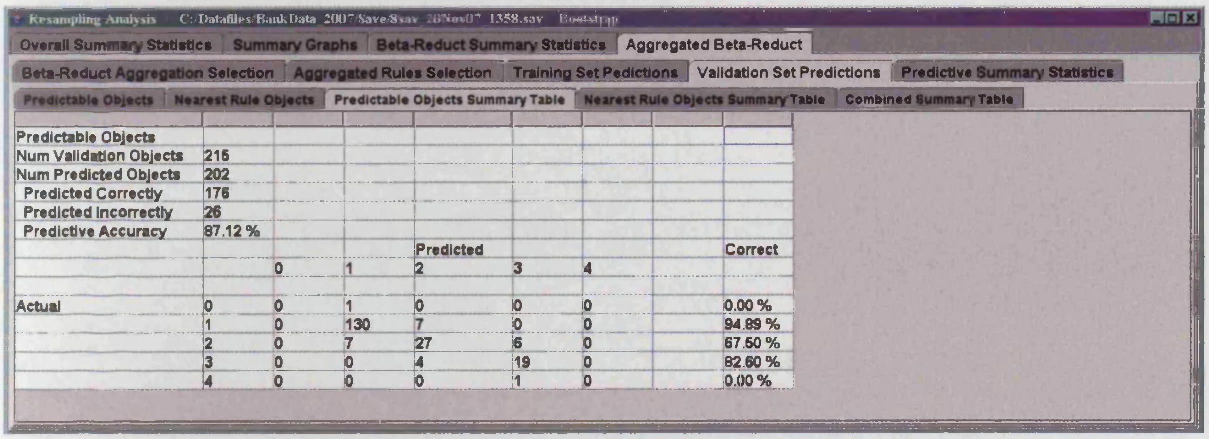
	Beta-Reducts	Occurrence	Selected
1	{ 1, 2, 3, 4, 5, 6, 7, 8}	44	☐
2	{ 7, 8}	29	☐
3	{ 1, 2, 7, 8}	19	☑
4	{ 3, 4, 8}	18	☐
5	{ 2, 7, 8}	17	☐
6	{ 1, 2, 3, 4, 5, 7, 8}	15	☐
7	{ 4, 6, 8}	15	☐
8	{ 3, 4, 7, 8}	15	☐
9	{ 1, 3, 4, 8}	14	☐
10	{ 1, 2, 3, 4, 5, 8}	13	☐

Figure 8.3.5: 'Beta Reduct Aggregation Selection' Panel Associated with the Bootstrap Analysis of the FIBR Training Set


Resampling Analysis C:/Datafiles/BankData_2007/Save/Ssav_20Nov07_1358.sav Bootstrap
Overall Summary Statistics
Summary Graphs
Beta-Reduct Summary Statistics
Aggregated Beta-Reduct
Beta-Reduct Aggregation Selection
Aggregated Rules Selection
Training Set Pedictions
Validation Set Predictions
Predictive Summary Statistics
Update Rule Aggregation

	1. Loan ...	2. Loan ...	7. EIU B ...	8. GDP/h...	Decision	Support	Correct	Strength	Certainty	Occurre...	Selected
10	1.0	0.0			1.0	307.0	280.0	0.0379	0.9373	18.0	☑
11		1.0		2.0	1.0	214.0	181.0	0.0264	0.8658	15.0	☑
12	0.0			6.0	1.0	204.0	189.0	0.0251	0.9497	6.0	☑
13		1.0		4.0	1.0	172.0	136.0	0.0212	0.7787	12.0	☑
14		0.0		4.0	1.0	135.0	132.0	0.0166	0.9879	11.0	☑
15	0.0			2.0	1.0	125.0	119.0	0.0154	0.9775	11.0	☑
16	0.0			4.0	1.0	119.0	104.0	0.0146	0.9018	8.0	☑
17	1.0			4.0	1.0	118.0	104.0	0.0145	0.8655	10.0	☑
18		1.0	0.0	6.0	1.0	114.0	94.0	0.0140	0.8340	9.0	□
19	0.0	0.0		6.0	1.0	107.0	92.0	0.0132	0.8598	2.0	□
20		0.0		2.0	1.0	87.0	82.0	0.0107	0.9553	10.0	□
21	1.0			2.0	1.0	75.0	64.0	0.0092	0.8167	6.0	□
22		1.0		3.0	1.0	57.0	51.0	0.0070	0.8577	6.0	□
23		1.0		6.0	1.0	57.0	48.0	0.0070	0.8649	4.0	□
24			0.0	6.0	1.0	54.0	45.0	0.0066	0.8815	5.0	□
25	2.0			6.0	1.0	49.0	40.0	0.0060	0.8420	6.0	□
26	1.0		0.0	6.0	1.0	47.0	37.0	0.0058	0.7823	3.0	□
27	1.0	1.0	0.0	1.0	1.0	46.0	41.0	0.0056	0.9291	3.0	□
28	2.0		0.0	3.0	1.0	46.0	39.0	0.0056	0.8849	6.0	□
29	1.0			6.0	1.0	42.0	36.0	0.0051	0.8571	1.0	□
30	2.0	1.0		3.0	1.0	39.0	31.0	0.0048	0.7075	2.0	□
31	0.0		0.0	6.0	1.0	34.0	31.0	0.0041	0.9516	2.0	□
32	1.0	1.0		6.0	1.0	29.0	21.0	0.0035	0.6720	2.0	□
33	1.0	1.0	0.0	6.0	1.0	25.0	15.0	0.0030	0.6145	2.0	□
34		0.0	0.0	6.0	1.0	24.0	21.0	0.0029	0.8750	1.0	□
35		0.0		3.0	1.0	20.0	20.0	0.0024	1.0000	8.0	□
36	2.0			2.0	1.0	20.0	17.0	0.0024	0.8928	2.0	□
37	0.0	1.0	0.0	6.0	1.0	19.0	19.0	0.0023	1.0000	4.0	□

Figure 8.3.6: Selection of Aggregated Rules Concomitant with the Aggregated β-reduct $\{c_1, c_2, c_7, c_8\}$, Associated with the Bootstrap Analysis of the FIBR Training Set


Resampling Analysis C:/Datafiles/BankData_2007/Save/Ssav_20Nov07_1358.sav Bootstrap
Overall Summary Statistics
Summary Graphs
Beta-Reduct Summary Statistics
Aggregated Beta-Reduct
Beta-Reduct Aggregation Selection
Aggregated Rules Selection
Training Set Pedictions
Validation Set Predictions
Predictive Summary Statistics
Predictable Objects
Nearest Rule Objects
Predictable Objects Summary Table
Nearest Rule Objects Summary Table
Combined Summary Table

Predictable Objects								
Num Validation Objects	215							
Num Predicted Objects	202							
Predicted Correctly	176							
Predicted Incorrectly	26							
Predictive Accuracy	87.12 %							
				Predicted				Correct
		0	1	2	3	4		
Actual	0	0	1	0	0	0		0.00 %
	1	0	130	7	0	0		94.89 %
	2	0	7	27	6	0		67.50 %
	3	0	0	4	19	0		82.60 %
	4	0	0	0	1	0		0.00 %

Figure 8.3.7: Application of the Aggregated Rules Concomitant with the Aggregated β-reduct $\{c_1, c_2, c_7, c_8\}$, Associated with Bootstrapping and Applied to the FIBR Validation Set

Compared to the leave-one-out and 50-fold cross-validation analyses, Figures 8.3.5 to 8.3.7 demonstrate a more diverse range of results. Figure 8.3.5 shows the selection of β-reduct $\{c_1, c_2, c_7, c_8\}$ for aggregation, and Figure 8.3.6 shows the selection of the aggregated rules concomitant with that aggregated β-reduct. Rule selection was based on the same selection principle used previously, utilising the average (mean) frequency of the rule occurrence over the five decision classes (these values can be found later in Table 8.4.2.5).

The 'Strength', 'Certainty' and 'Occurrence' columns within Figure 8.3.6, clearly show a much more diverse range of values compared to the equivalent leave-one-out analysis (see Figure 8.1.3.2). This could indicate that the rules associated with the aggregated β-reducts, with regards to bootstrapping, are less biased towards a specific rule set, and are more varied across the bootstrap repetitions. This is interesting because, it is in complete contrast to the theory relating to accuracy estimates compared between with the leave-one-out and bootstrapping analyses, which states that, leave-one-out is the least biased but has the highest variance (see Chapter 3 subsection 3.3.4).

Comparing the results of applying the aggregated rule set to the FIBR validation set, between the bootstrapping analysis shown in Figure 8.3.7 and the leave-one-out analysis shown in Figure 8.1.3.3 (n.b. the results of 50-fold cross-validation were similar to those of the leave-one-out analysis), it can be seen that, bootstrapping gives a classification to two more banks (202 compared to 200), and predicts correctly a greater proportion of the banks it has given a classification to (87.12% compared to 83.33%). Moreover, the breakdown of the predictive accuracies across the five decision classes with respect to bootstrapping is higher for the 'B' (1) grade (94.89% compared to 94.07%), 'C' (2) grade (67.50% compared to 62.22%) and 'D' (3) grade (82.60 compared to 68.18%) banks, with the 'A' (0) and 'E' (4) grade banks both having equivalent predictive accuracies (0%) to those shown in Figure 8.3.7. This could indicate that the results based on bootstrapping can improve on the results based on leave-one-out. It must also be kept in mind that, the bootstrapping results were based on an in-sample set which was on average, only 62.3% of the training set, whereas the leave-one-out results were based on 99.75% of the training set (404/405), which is impressive with regards to the bootstrapping results, since they are associated with higher predictive accuracies on the validation set.

The following section provides a full exposition and comparison of the out-of-bag estimates over the three re-sampling methods, and a comparison of the β-reduct aggregation with regards to bootstrapping and leave-one-out, applied to the FIBR validation set.

8.4 Comparison of Leave-one-out, *k*-fold Cross-validation and Bootstrapping Predictive Results

This section is split into two subsections, which presents in a number of tables, information collated from the software during a number of comparative analyses. Firstly, subsection 8.4.1 compares the out-of-bag predictive accuracy estimates between the β-reducts selected during the leave-one-out, *k*-fold cross-validation and bootstrapping analyses. Particular reference is made to the bias and variance between the three re-sampling methods. Secondly, subsection 8.4.2 compares the estimated predictive accuracies based on aggregating the selected β-reducts from the leave-one-out and bootstrapping results, and applying those aggregated β-reducts to the FIBR validation set. Subsection 8.4.2 does not consider *k*-fold cross-validation, as it was shown earlier in this chapter, for higher values of *k*, the results are asymptotic to the leave-one-out analysis, hence, it was only necessary to present the difference in results, between leave-one-out and bootstrap aggregation.

8.4.1 Out-of-bag Estimates

As stated, this subsection compares the out-of-bag estimates over the three re-sampling methods (two sets of results are presented for bootstrapping, i.e. the *e*0 and the 0.632B estimates), and presents them in four tables 8.4.1.1 to 8.4.1.4. These tables show the selected β-reducts within the first column and the frequency of occurrence in the second. The remaining columns display separately, the mean average predictive accuracy estimates and standard deviations of the selected β-reducts for: all banks in the out-of-sample, only those banks within the out-of-sample predictable by matching rules, and those banks within the out-of-sample predicted by nearest rules. The information was collated from the statistical breakdown of the 'Individual Beta-reducts Statistics' panel, see in Figure 8.1.2.1.1 subsection 8.1.2.1.

		Leave-one-out Estimated Predictive Accuracies (Out-of-bag Estimates)					
		All Banks		Predictable Banks by Matching Rule		Predictable by Nearest Rule	
β-reduct	Occurrence	Mean	S.D.	Mean	S.D.	Mean	S.D.
$\{c_1, c_3, c_4, c_6, c_8\}$	323	80.81%	39.38	81.79%	38.59%	50.00%	50.00%
$\{c_2, c_6, c_7, c_8\}$	16	50.00%	50.00	50.00%	50.00%	Na	Na
$\{c_3, c_4, c_5, c_6, c_7\}$	15	26.67%	44.22	26.67%	44.22%	Na	Na
$\{c_2, c_4, c_8\}$	10	0.00%	0.00	0.00%	0.00%	Na	Na
$\{c_1, c_2, c_3, c_4, c_5, c_8\}$	8	75.00%	43.30	75.00%	43.30%	Na	Na
$\{c_1, c_2, c_3, c_4, c_7, c_8\}$	5	0.00%	0.00	Na	Na	0.00%	0.00
$\{c_2, c_7, c_8\}$	4	0.00%	0.00	0.00%	0.00%	Na	Na
$\{c_1, c_3, c_4, c_6, c_7, c_8\}$	4	50.00%	50.00	Na	Na	50.00%	50.00%
$\{c_1, c_3, c_4, c_7, c_8\}$	3	100.00%	0.000	Na	Na	100.00%	0.00%
$\{c_2, c_3, c_4, c_5, c_7, c_8\}$	3	0.00%	0.000	Na	Na	0.00%	0.00%

Table 8.4.1.1: Occurrence and Average Predictive Accuracy (Out-of-bag) Estimates of the Top Ten Most Frequently Selected β-reducts, Associated with the Leave-one-out Analysis of the FIBR Training Set

		50-fold Cross-validation Estimated Predictive Accuracies (Out-of-bag Estimates)					
		All Banks		Predictable Banks by Matching Rule		Predictable by Nearest Rule	
β-reduct	Occurrence	Mean	S.D.	Mean	S.D.	Mean	S.D.
$\{c_1, c_3, c_4, c_6, c_8\}$	15	70.00%	16.96	70.56%	15.35	42.86%	49.49
$\{c_2, c_6, c_7, c_8\}$	7	82.54%	5.79	82.54%	5.79	Na	Na
$\{c_3, c_4, c_5, c_6, c_7\}$	5	80.56%	5.76	80.56%	5.76	Na	Na
$\{c_1, c_3, c_8\}$	4	75.00%	8.84	77.23%	6.11	0.00%	0.00
$\{c_1, c_3, c_4, c_6, c_7, c_8\}$	3	71.30%	16.09	72.49%	12.98	50.00%	50.00
$\{c_3, c_4, c_6, c_8\}$	3	63.89%	10.39	66.67%	11.79	0.00%	0.00
$\{c_2, c_7, c_8\}$	2	75.00%	0.00	75.00%	0.00	Na	Na
$\{c_1, c_2, c_3, c_4, c_6, c_8\}$	2	59.03%	3.47	59.03%	3.47	Na	Na
$\{c_2, c_4, c_8\}$	1	75.00%	0.00	75.00%	0.00	Na	Na
$\{c_4, c_7, c_8\}$	1	37.50%	0.00	37.50%	0.00	Na	Na

Table 8.4.1.2: Occurrence and Average Predictive Accuracy (Out-of-bag) Estimates of the Top Ten Most Frequently Selected β-reducts, Associated with the 50-fold Cross-validation Analysis of the FIBR Training Set

		$e0$ Bootstrap Estimated Predictive Accuracies (%) (Out-of-bag Estimates)					
		All Banks		Predictable Banks by Matching Rule		Predictable by Nearest Rule	
β-reduct	Occurrence	Mean	S.D.	Mean	S.D.	Mean	S.D.
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	44	65.943	5.969	72.584	4.402	48.157	16.481
$\{c_7, c_8\}$	29	71.214	3.071	72.747	3.231	34.985	36.480
$\{c_1, c_2, c_7, c_8\}$	19	69.817	3.084	72.090	3.084	24.226	23.619
$\{c_3, c_4, c_8\}$	18	70.973	3.133	72.373	3.462	52.166	22.065
$\{c_2, c_7, c_8\}$	17	71.404	2.602	73.126	1.865	11.006	14.302
$\{c_1, c_2, c_3, c_4, c_5, c_7, c_8\}$	15	70.925	4.011	74.409	2.450	48.919	13.346
$\{c_4, c_6, c_8\}$	15	69.743	2.160	70.47	2.503	46.678	29.617
$\{c_3, c_4, c_7, c_8\}$	15	71.523	2.129	73.015	2.074	44.778	23.552
$\{c_1, c_3, c_4, c_8\}$	14	70.961	3.135	72.213	3.200	42.943	17.825
$\{c_1, c_2, c_3, c_4, c_5, c_8\}$	13	70.309	4.369	71.49	4.134	47.895	28.239

Table 8.4.1.3: Occurrence and Average Predictive Accuracy (Out-of-bag) $e0$ Estimates of the Top Ten Most Frequently Selected β-reducts, Associated with the Bootstrap Analysis of the FIBR Training Set

		0.632B Bootstrap Estimated Predictive Accuracies (Out-of-bag Estimates)					
		All Banks		Predictable Banks by Matching Rule		Predictable by Nearest Rule	
β-reduct	Occurrence	Mean	S.D.	Mean	S.D.	Mean	S.D.
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	44	69.78	5.62	78.72	3.30	46.61	14.32
$\{c_7, c_8\}$	29	72.13	2.04	73.68	2.14	33.48	28.80
$\{c_1, c_2, c_7, c_8\}$	19	71.74	2.19	74.00	2.19	25.64	20.04
$\{c_3, c_4, c_8\}$	18	73.11	2.30	74.56	2.57	49.27	17.93
$\{c_2, c_7, c_8\}$	17	72.33	1.90	74.00	1.19	13.98	13.95
$\{c_1, c_2, c_3, c_4, c_5, c_7, c_8\}$	15	75.00	3.65	79.37	1.85	46.66	11.00
$\{c_4, c_6, c_8\}$	15	70.89	1.54	71.75	1.74	45.27	23.43
$\{c_3, c_4, c_7, c_8\}$	15	74.19	1.51	75.75	1.45	44.22	20.39
$\{c_1, c_3, c_4, c_8\}$	14	73.71	2.33	75.10	2.31	41.09	14.87
$\{c_1, c_2, c_3, c_4, c_5, c_8\}$	13	74.88	3.22	76.15	3.09	47.50	25.35

Table 8.4.1.4: Occurrence and Average Predictive Accuracy (Out-of-bag) 0.632B Estimates of the Top Ten Most Frequently Selected β-reducts, Associated with the Bootstrap Analysis of the FIBR Training Set

In general, the estimated predicted accuracy results, reflect the expected pattern with regards to bias and variance (shown here as standard deviation) (Chapter 3 subsection 3.3.4; Weiss and Kulikowski, 1991). That is, the mean estimated predictive accuracies of the bootstrapping estimates are more pessimistic than those estimates associated with the leave-one-out and *k*-fold cross-validation analyses. To re-iterate briefly, leave-one-out should be nearly an unbiased estimate, *k*-fold cross-validation should be slightly pessimistic and bootstrapping should show quite pessimistic estimates. Though it is difficult to draw too much inference from the leave-one-out results, given that for the less frequently selected β-reducts (those other than β-reduct $\{c_1, c_3, c_4, c_6, c_8\}$), they are only tested on a single bank for the few times they occur, hence confidence in their estimated predictive accuracy is not possible. This perhaps, reflects the position taken by Shao (1993), that leave-one-out can be deficient and the more general *k*-fold cross-validation can rectify its deficiencies. Han and Kamber (2006) shares the same sentiment, stating that in general, *k*-fold cross-validation is recommended for estimating accuracy even if computation power allows using more folds (with leave-one-out being the extreme limit).

With regards to the variance, here, by looking at the standard deviation values, again the re-sampling methods follow the expected pattern. That is, leave-one-out shows the highest variance, *k*-fold cross-validation shows markedly less variance, and bootstrapping generally shows the least variance.

The predictive accuracy estimates between the banks predictable by matching rules, and those predictable only by nearest rules, for all three re-sampling methods; indicates strong evidence for the conclusion that the nearest rule approach to predicting banks, where no matching rule can be found, performs relatively poorly. This was also observed with regards to the vein graph results shown in chapter 7 section 7.3, and will additionally be emphasised later in this chapter with regards to the full exposition of the β-reduct aggregation results on the validation set (also see aggregation results based on banks only predictable by nearest rule, in Appendix 8).

With regards to the *e*0 and 0.632B bootstrap estimates (Tables 8.4.1.3 and 8.4.1.4), again, the results fit the pattern as described in the related literature (see Weiss and Kulikowski, 1991), which suggested that the *e*0 estimates (only calculated on the out-of-sample banks), would be more pessimistic than the 0.632B estimates. These results were also reflected within the 'Overall

Summary Statistics' panel shown previously in Figure 8.3.4 (relating to the bootstrapping results), which indicate that the 0.632B estimates (denoted as '.632B Estimates' in the row heading) were slightly less pessimistic than the *e*0 estimates.

Interestingly, Figure 8.3.4 showed that the average number of banks within the out-of-samples is 148.7470, which as a proportion of the 405 banks within the training set is 0.367 (rounding down). Hence, the average proportion of banks taken from the training set and used in the in-sample, once or more, during bootstrapping process is 0.633 (rounded up). These values are almost exactly the proportions estimated by Efron and Tibshirani (1993) and Breiman (1996a). Furthermore, the 0.632B bootstrap uses the fixed proportional coefficients 0.632 and 0.368, but it is worth considering, if the analysis has access to the exact proportions (as the developed software does here), the coefficients of the 0.632B equation (see Chapter 3 Equation 3.3.4.1) could be adaptively assigned to utilise the exact proportions calculated at the end of the bootstrapping process, and this may yield better predictive accuracy estimates.

8.4.2 Comparison of Aggregated β-reduct Validation Set Results, Based on the Leave-one-out, Bootstrapping, and Vein Graph Analyses

This section compares the results of β-reduct aggregation, with regards to all selected β-reducts, associated with the leave-one-out and bootstrap re-sampling methods, applied to the validation set from the FIBR data. The results are also compared with the vein graph analysis of the FIBR data previously reported in Chapter 7.

Table 8.4.2.1 presents the results of the β-reduct aggregation process, for the top ten most frequently occurring β-reducts selected during the leave-one-out analysis. The aggregated β-reducts have been applied to the FIBR validation set, where the results shown, are only based on the banks that were predictable by matching aggregated rules, associated with the relevant aggregated β-reducts (for the results based on the banks predictable only by the nearest rule method, see

Appendix A sections A.5 and A.6)

Taking the aggregated β-reduct $\{c_1, c_3, c_4, c_6, c_8\}$ as an example, contained in Table 8.4.2.1, it is associated with 104 aggregated rules (column two), from which a subset of aggregated rules are selected (described next); from the subset of aggregated rules: 200 out of the 215 banks within the validation set were predictable by matching rules (column three), 168 of those predicted were predicted correctly (column four), with 32 being predicted incorrectly (column five), giving a predictive accuracy of 84.00% (column six).

β-reduct	Number of Aggregated Rules	Banks Predicted by Matching Rule	Banks Predicted Correctly	Banks Predicted Incorrectly	Predictive Accuracy (%)
$\{c_1, c_3, c_4, c_6, c_8\}$	104	200	168	32	84.00%
$\{c_2, c_6, c_7, c_8\}$	16	204	166	38	81.37%
$\{c_3, c_4, c_5, c_6, c_7\}$	66	202	165	37	81.68%
$\{c_2, c_4, c_8\}$	13	208	176	32	84.61%
$\{c_1, c_2, c_3, c_4, c_5, c_8\}$	101	206	170	36	82.52%
$\{c_1, c_2, c_3, c_4, c_7, c_8\}$	72	205	169	36	82.43%
$\{c_2, c_7, c_8\}$	14	208	169	39	81.25%
$\{c_1, c_3, c_4, c_6, c_7, c_8\}$	76	198	169	29	85.35%
$\{c_1, c_3, c_4, c_7, c_8\}$	65	195	168	27	86.15%
$\{c_2, c_3, c_4, c_5, c_7, c_8\}$	88	190	162	28	85.26%

Table 8.4.2.1: Aggregated β-reduct Results Associated with the Leave-one-out Analysis Applied to the FIBR Validation Set

As stated, the number of aggregated rules presented in Table 8.4.2.1 is not the final number of rules selected for the classification of the banks within the validation set. The number of aggregated rules selected from the 'Aggregated Rules Selection' panel (see Figure 8.1.3.2), was based on the average number of rules (rounded to the nearest integer) that predicted each individual decision class, discussed previously and shown in subsection 8.1.2.2. The average number of aggregated rules selected and associated with all the selected β-reducts with regards to the leave-one-out analysis, have been collated from the panel shown earlier in Figure 8.1.2.2.1 and are shown below in Table 8.4.2.2. Table 8.4.2.3 then presents the breakdown of the validation set predictive accuracies, across the five decision classes, for each aggregated β-reduct.

β-reduct	Average Number of Rules Predicting Each Individual Decision Class					Total
	0 (A)	1 (B)	2 (C)	3 (D)	4 (E)	
$\{c_1, c_3, c_4, c_6, c_8\}$	0.00	20.00	26.97	16.98	3.00	66.92
$\{c_2, c_6, c_7, c_8\}$	0.00	7.00	7.00	2.00	0.00	16.00
$\{c_3, c_4, c_5, c_6, c_7\}$	0.00	13.00	22.27	14.07	3.07	52.47
$\{c_2, c_4, c_8\}$	0.00	5.00	6.00	2.00	0.00	13.00
$\{c_1, c_2, c_3, c_4, c_5, c_8\}$	3.90	33.00	34.88	19.00	4.88	95.63
$\{c_1, c_2, c_3, c_4, c_7, c_8\}$	2.00	23.00	32.00	11.00	4.00	72.00
$\{c_2, c_7, c_8\}$	0.00	7.00	5.00	2.00	0.00	14.00
$\{c_1, c_3, c_4, c_6, c_7, c_8\}$	0.00	24.00	31.5	17.00	3.00	75.00
$\{c_1, c_3, c_4, c_7, c_8\}$	0.00	21.00	29.00	12.00	3.00	65.00
$\{c_2, c_3, c_4, c_5, c_7, c_8\}$	4.00	30.00	33.00	17.00	4.00	88.00

Table 8.4.2.2: Average and Total Number of Rules Predicting Each Individual Decision Class, Associated with the β-reducts Selected During the Leave-one-out Analysis of the FIBR Training Set

β-reduct	Individual Decision Class Predictive Accuracies				
	0 (A)	1 (B)	2 (C)	3 (D)	4 (E)
$\{c_1, c_3, c_4, c_6, c_8\}$	0.00%	95.45%	63.63%	63.63%	0.00%
$\{c_2, c_6, c_7, c_8\}$	0.00%	90.00%	59.52%	75.00%	0.00%
$\{c_3, c_4, c_5, c_6, c_7\}$	0.00%	94.07%	53.48%	63.63%	100.00%
$\{c_2, c_4, c_8\}$	0.00%	92.25%	65.90%	80.00%	0.00%
$\{c_1, c_2, c_3, c_4, c_5, c_8\}$	0.00%	94.11%	63.04%	56.52%	0.00%
$\{c_1, c_2, c_3, c_4, c_7, c_8\}$	0.00%	92.02%	67.44%	59.09%	0.00%
$\{c_2, c_7, c_8\}$	0.00%	92.14%	54.34%	75.00%	0.00%
$\{c_1, c_3, c_4, c_6, c_7, c_8\}$	0.00%	96.94%	62.79%	68.18%	0.00%
$\{c_1, c_3, c_4, c_7, c_8\}$	0.00%	96.15%	70.73%	63.63%	0.00%
$\{c_2, c_3, c_4, c_5, c_7, c_8\}$	0.00%	93.60%	68.29%	72.72%	100.00%

Table 8.4.2.3: Breakdown of the Validation Set Predictive Accuracies Across the Five Decision Classes, Associated with the β-reducts Selected During the Leave-one-out Analysis of the FIBR Training Set

Within Table 8.4.2.3, a zero percent predictive accuracy was recorded for all 'A' (0) grade banks, but as shown in Chapter 6, the validation set only contains one 'A' grade bank, hence no inference can really be extracted from the predictive accuracy results associated with the 'A' grade banks. Furthermore, as shown in Table 8.4.2.2, only three out of the ten aggregated β-reducts are associated with aggregated rule sets capable of predicting the 'A' grade banks.

Tables 8.4.2.4, 8.4.2.5 and 8.4.2.6 present similar information as those three tables described above, but associated with the aggregation of the β-reducts selected during the bootstrap re-sampling analysis of the FIBR training set, applied to the FIBR validation set.

β-reduct	Number of Aggregated Rules	Banks Predicted by Matching Rule	Banks Predicted Correctly	Banks Predicted Incorrectly	Predictive Accuracy (%)
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	565	163	139	24	85.27%
$\{c_7, c_8\}$	19	200	172	28	86.00%
$\{c_1, c_2, c_7, c_8\}$	102	202	176	26	87.12%
$\{c_3, c_4, c_8\}$	86	204	175	29	85.78%
$\{c_2, c_7, c_8\}$	37	206	172	34	83.49%
$\{c_1, c_2, c_3, c_4, c_5, c_7, c_8\}$	366	184	159	25	86.41%
$\{c_4, c_6, c_8\}$	46	199	166	33	83.41%
$\{c_3, c_4, c_7, c_8\}$	133	201	170	31	84.57%
$\{c_1, c_3, c_4, c_8\}$	169	209	170	39	81.33%
$\{c_1, c_2, c_3, c_4, c_5, c_8\}$	329	205	174	31	84.87%

Table 8.4.2.4: Aggregated β-reduct Results Associated with the Bootstrap Analysis Applied to the FIBR Validation Set

β-reduct	Average Number of Rules Predicting Each Individual Decision Class					Total
	0 (A)	1 (B)	2 (C)	3 (D)	4 (E)	
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	3.23	28.20	30.02	15.64	3.05	80.14
$\{c_7, c_8\}$	0.00	4.72	4.45	1.79	0.10	11.06
$\{c_1, c_2, c_7, c_8\}$	0.37	10.42	12.95	4.95	1.05	29.74
$\{c_3, c_4, c_8\}$	0.83	10.11	13.11	7.72	0.83	32.60
$\{c_2, c_7, c_8\}$	0.06	6.53	5.59	2.06	0.29	14.53
$\{c_1, c_2, c_3, c_4, c_5, c_7, c_8\}$	3.27	29.33	21.27	14.67	3.53	72.07
$\{c_4, c_6, c_8\}$	0.00	5.13	6.53	3.20	0.20	15.06
$\{c_3, c_4, c_7, c_8\}$	0.6	13.13	17.47	9.20	1.33	41.73
$\{c_1, c_3, c_4, c_8\}$	1.14	15.00	19.07	10.07	1.93	47.21
$\{c_1, c_2, c_3, c_4, c_5, c_8\}$	3.62	27.85	28.69	13.77	4.23	78.16

Table 8.4.2.5: Average and Total Number of Rules Predicting Each Individual Decision Class, Associated with the β-reducts Selected During the Bootstrap Analysis of the FIBR Training Set

	Individual Decision Class Predictive Accuracies				
β-reduct	0 (A)	1 (B)	2 (C)	3 (D)	4 (E)
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	0.00%	99.05%	57.89%	63.15%	0.00%
$\{c_7, c_8\}$	0.00%	92.75%	70.27%	78.26%	0.00%
$\{c_1, c_2, c_7, c_8\}$	0.00%	94.89%	67.50%	82.60%	0.00%
$\{c_3, c_4, c_8\}$	0.00%	92.70%	69.76%	77.27%	100.00%
$\{c_2, c_7, c_8\}$	0.00%	92.95%	59.52%	75.00%	0.00%
$\{c_1, c_2, c_3, c_4, c_5, c_7, c_8\}$	0.00%	97.58%	62.50%	68.42%	0.00%
$\{c_4, c_6, c_8\}$	0.00%	91.17%	64.28%	78.94%	0.00%
$\{c_3, c_4, c_7, c_8\}$	0.00%	92.08%	66.66%	71.42%	100.00%
$\{c_1, c_3, c_4, c_8\}$	0.00%	92.85%	56.52%	66.66%	0.00%
$\{c_1, c_2, c_3, c_4, c_5, c_8\}$	0.00%	96.35%	62.22%	63.63%	0.00%

Table 8.4.2.6: Breakdown of the Validation Set Predictive Accuracies Across the Five Decision Classes, Associated with the β-reducts Selected During the Bootstrap Analysis of the FIBR Training Set

The predictive accuracies displayed in Table 8.4.2.4, generally show an increase over the accuracies associated with the aggregated β-reducts shown previously, with regards to the leave-one-out analysis (see Table 8.4.2.1). This is further reflected in Table 8.4.2.6, which gives the breakdown across the individual decision classes for the banks that were predictable by a matching rule from one of the bootstrap aggregated β-reducts.

With regards to the number of rules capable of predicting each decision class associated with the β-reducts selected during the bootstrap re-sampling analysis, shown in Table 8.4.2.5, when compared to Table 8.4.2.2, there are more selected β-reducts in Table 8.4.2.5 that are capable of predicting the 'A' (0) grade banks. That is, for eight out of the ten selected β-reducts within Table 8.4.2.5, during one of the repetitions, the selected β-reducts' associated rule sets contained a rule, capable of predicting an 'A' grade bank.

From the information presented in Tables 8.4.2.4 to 8.4.2.6, in general, it appears that the aggregated β-reducts based on bootstrapping of the FIBR data, perform better than those of the leave-one-out analysis. Table 8.4.2.7 further supports this conclusion by comparing the average values of the predictive accuracies and average values of the breakdown of the predictive accuracies for the bootstrapping and leave-one-out predictive performances on the validation set; it also includes the average number of banks predicted from the 215 banks within the validation set. Table 8.4.2.7 additionally compares the results with the set of β-reducts shown in Chapter 7 section 7.4, associated with the vein graph analysis of the FIBR validation set. The results are compared to the vein graph analysis to asses whether, the extra processing required to produced aggregated β-reducts translates, into improved predictive performances.

Analysis Method	Reducts' Average Predictive Accuracy	Average Number of Banks Predicted from 215 Banks	Individual Decision Class Predictive Accuracies				
			0 (A)	1 (B)	2 (C)	3 (D)	4 (E)
Vein Graph β-reducts	84.35%	181.78	0.00%	94.58%	62.13%	73.67%	0.00%
Leave-one-out Aggregated β-reducts	83.46%	201.60	0.00%	93.67%	62.92%	67.74%	20.00%
Bootstrapping Aggregated β-reducts	84.83%	197.30	0.00%	94.24%	63.71%	72.54%	20.00%
Difference between bootstrapping and Vein Graph β-reducts	+0.48%	+15.52	0.00%	–0.34%	+1.58%	–1.13%	20.00%
Difference between bootstrapping and Leave-one-out Aggregated β-reducts	+1.37%	–4.30	0.00%	+0.57%	+0.79%	+4.8%	0.00%

Table 8.4.2.7: Averaged Predictive Accuracy Values, and Differences Between Bootstrapping Compared to the Leave-one-out and Vein Graph Predictive Performances on the FIBR Validation Set

Table 8.4.2.7 indicates that the bootstrap aggregated β-reducts show, on average, a 1.37% increase in predictive accuracy over the leave-one-out analysis. This is not strong evidence that bootstrap aggregation is better than leave-one-out, but on the breakdown of the individual decision class predictive accuracies, there is a 4.8% increase on the (C) grade banks. Other studies indicated that the increase of predictive accuracy associated with bootstrap aggregation was between zero and five percent (Breiman, 1996a; Stefanowski, 2004, 2007). Hence, an increase of almost five percent, if only on one decision class may be significant; it also illustrates again, the importance of reporting the predictive accuracies over all the decision classes individually.

When compared to the average vein graph β-reduct predictive accuracies on the validation set (the banks predictable by matching rules) shown in Chapter 7 section 7.4, there is less evidence that bootstrap aggregation has improved the predictive performance, with only a 0.48% increase. Though, the bootstrap aggregated β-reducts did, on average, predict 15.52 more banks from the validation set compared to the average of the vein graph β-reducts, with comparable levels of predictive accuracies. Additionally, the results of the bootstrap aggregated β-reducts within Table 8.4.2.4, contain the top two highest predictive accuracies (87.12% and 86.41%) compared to the vein graph results, and the top four (87.12%, 86.41%, 86.00% and 85.78%) when compared to the leave-one-out aggregated β-reducts.

Here, there is some evidence that the bootstrap aggregated β-reducts would perform better on any future unseen data, but it is perhaps not ideal comparing average predictive accuracy results between analyses that yield different reducts (β and aggregated β), and the results of the accuracies based on the validation set are not conclusive. Additionally, the results shown here, are comparable with the closest known study by Stefanowski (2004), which investigated the increase of predictive accuracy through the application of bootstrapping to their RST based rule construction process. However here, we have shown that, the predictive accuracy may actually improve particular decision class accuracies.

It should be kept in mind though, that the re-sampling analysis is not solely about improving the predictive accuracy results of the β-reducts through classifier aggregation, but also about better estimations of the β-reducts future predictive accuracy, through calculation of the out-of-bag estimates. Clearly, when compared to the out-of-bag estimates shown previously in subsection 8.4.1, the predictive accuracies based on the validation set for both the vein graph (see Chapter 7 section 7.4) and re-sampling analyses are over optimistic.

8.5 Further Benchmark Results

This final section provides further VPRS re-sampling and β-reduct aggregation results based on a number of benchmark data sets. It also includes the predictive accuracy results of another classification method, namely Multi Discriminant Analysis (MDA) (Altman, 1968; Beynon and Peel, 2001; Jingbo et al., 2005) based on those benchmark data sets. The purpose of this section is to allow comparisons to be made with other classification methods that exist in the extant literature.

Five benchmark data sets, that are regularly utilised in the related literature (Breiman, 1996a; Kotsianti and Kanellopoulos, 2007; Stefanowski, 2007), were selected here, and can be found at the UCI machine learning repository (http://archive.ics.uci.edu/ml/index.html). Table 8.5.1 describes these data sets, namely the breast-cancer (Wisconsin original version), iris, SPECT, wine, and zoo data sets.

Data Set	Number of Objects	Training Set	Validation Set	Number of Attributes	Attributes Selected for Analysis	Decision Classes
breast-cancer	699	363	336	10	7	2
iris	150	132	18	4	4	3
SPECT	267	198	69	22	6	2
wine	178	153	25	13	8	3
zoo	101	91	10	17	6	7

Table 8.5.1: Details Regarding the Selected Benchmark Data Sets

Describing Table 8.5.1 in more detail, the second column within the table states the number of objects associated with each data set, the third and fourth columns display the number of objects associated with the training and validation data sets, respectively (using the statistical sub-sampling method described in section 3.1.2). The fifth column displays the number of attributes associated with the respective data sets. The sixth column indicates how many of those attributes were selected for the VPRS re-sampling analyses[28], and the last column displays the number of decision classes associated with each of the data sets. Table 8.5.2 displays the information relating to the re-

[28] See Appendix A6 for further details on the attributes associated with the selected benchmark data sets.

sampling analyses and β-reducts selected for aggregation, based on the benchmark data sets.

Data Set	β Threshold Value Selected	Leave-one-out β-reduct Selected for Aggregation	Bootstrapping β-reduct Selected for Aggregation	Number of Bootstrap Repetitions
breast-cancer	0.69	$\{c_3, c_4, c_7\}$	$\{c_3, c_4, c_7\}$	400
iris	0.70	$\{c_1, c_3, c_4\}$	$\{c_3, c_4\}$	200
SPECT	0.52	$\{c_1, c_2, c_3, c_5, c_6\}$	$\{c_1, c_2, c_3, c_5, c_6\}$	200
wine	0.65	$\{c_1, c_7\}$	$\{c_5, c_6, c_7\}$	200
zoo	0.65	$\{c_3, c_4, c_5, c_6\}$	$\{c_3, c_4, c_5, c_6\}$	200

Table 8.5.2: Information Regarding Parameter Settings and β-reducts Selected for Aggregation, Associated with the Leave-one-out and Bootstrapping Analyses of the Benchmark Data Sets

To describe Table 8.5.2, taking the iris data set as an example, the first column indicates that a β threshold value of 0.7 was set during the parameter set-up stage (see section 6.1) (determined from a vein graph analysis of the iris data set), and that based on the leave-one-out analysis, the β-reduct $\{c_1, c_3, c_4\}$ was selected for aggregation, as shown in the third column. Furthermore, based on a bootstrap analysis (and a process considering the re-sampling results, similar to the process described throughout this chapter) the β-reduct $\{c_3, c_4\}$ was selected for aggregation, shown in the fourth column. The last column indicates that 200 bootstrap repetitions were undertaken with regards to the bootstrapping analysis of the iris data set. As has been the case with section 8.4, a k-fold cross validation analysis was not undertaken, as the results are asymptotic to leave-one-out for high values of k.

Referring now to the predictive results of the β-reducts selected for aggregation, associated with the leave-one-out and bootstrapping analyses of the benchmark data sets, Table 8.5.3 displays the out-of-bag (re-sampling) predictive accuracies, on all the objects given a classification, those objects only classified by matching rules, and the percentage of those objects given a classification that were given a classification by a matching rule.

Data Set	Leave-one-out (%)			Bootstrapping (%)		
	Out-of-Bag Predictive Accuracy	Objects Predicted by Matching Rule	Predictive Accuracy on Objects Predicted by Matching Rule	Out-of-Bag Predictive Accuracy	Objects Predicted by Matching Rule	Predictive Accuracy on Objects Predicted by Matching Rule
breast-cancer	96.58	100.00	96.58	95.77 (95.79)	99.32	95.74 (95.87)
iris	96.88	98.00	98.41	97.90 (97.37)	97.32	100.00 (100.00)
SPECT	71.88	93.00	73.83	63.42 (66.37)	88.97	63.61 (67.00)
wine	94.78	100.00	94.78	89.23 (90.28)	98.00	90.08 (91.01)
zoo	91.36	95.00	91.36	87.89 (87.14)	91.55	96.15 (95.91)

Table 8.5.3: Out-of-Bag Re-sampling Predictive Accuracies for the Leave-one-out and Bootstrapping Analyses of the Benchmark Data Sets

Describing Table 8.5.3 in more detail, and again referring to the iris data. Considering first the leave-one out analysis (columns two to four), there is an out of bag predictive accuracy on all objects given a classification (by matching or nearest rule) of 96.88%, of those objects 98.00% were predicted by matching rules, and of those objects predicted by matching rules, 98.41% were predicted correctly. With regards to the bootstrapping analysis of the iris data (columns five to seven), there is an out-of-bag e0 predictive accuracy on all objects given a classification (by matching or nearest rule) of 97.90% and a 0.632B estimate (in parenthesis) of 97.37%, of those objects 97.32% were predicted by matching rules, and of those objects predicted by matching rules, 100.0% were predicted correctly for both the e0 and 0.632B estimates.

More generally, the results within Table 8.5.3, indicate quite optimistic estimates of the predictive accuracies, with perhaps the SPECT data set being an exception. Though, this is not to say that the estimates are bias or over-optimistic; as described in the theory in section 3.3, leave-one-out should provide an almost unbiased estimate, and bootstrapping should typically be pessimistic. Though there is no clear trend here, with regards to the expected biases, with both the bootstrapping results for the breast-cancer and iris data sets appearing more optimistic than the leave-one-out results. What is perhaps interesting to see here though, is that the results based on the zoo data set, a relatively small data set (101 objects) with seven decision classes (thus on average 14 objects per decision class) still achieves re-sampling predictive accuracies of over 87.89%.

As a comparison, the results based on the iris, wine, zoo and breast-cancer data sets are now compared to a similar study by Kotsiantis and Kanellopoulos (2007), who implemented a number of different ensemble[29] versions of the Decision Stump classifier (DS) (a type of decision tree classifier, see Murthy, 1998) and applied them to a number of benchmark datasets, including the four named data sets (they did not use the SPECT data set). The comparisons are shown in Table 8.5.4.

Data Set	DS	Bagging DS	Dagging DS	Adaboost DS	Multiboost DS	Decorate DS	VOTE DS	Leave-one-out VPRS	Bootstrapping VPRS
breast-cancer	69.27	73.44	72.53	71.55	71.76	75.18	73.90	**96.58**	95.77 (95.79)
iris	66.67	70.33	78.13	95.07	94.73	93.93	95.07	96.88	**97.90** (97.37)
wine	57.91	85.16	71.21	91.57	91.17	**96.45**	91.74	94.78	89.23 (90.28)
zoo	60.43	60.63	39.51	60.43	60.43	61.96	60.43	**91.36**	87.89 (87.14)

Table 8.5.4: Comparison Between the Leave-one-out and Bootstrap Predictive Accuracies (%) Compared to a Number of DS Ensemble Classifiers

It is clear form Table 8.5.4 that the predictive accuracies, estimated by the leave-one-out and bootstrapping VPRS analyses, generally outperform those based on Kotsiantis and Kanellopoulos' (2007) work. Indeed, the re-sampling VPRS results outperform the DS ensemble classifier results in three out of the four data sets (highest predictive accuracies highlighted in bold).

As a further comparison, taking the most recent similar work within the RST literature, namely Stefanowski (2007), based on the iris and the zoo data sets, the best predictive accuracies presented by *ibid* with regards to their n^{2}MODLEM classifier were 95.53% and 94.64%, respectively. Here,

[29] The implement ensemble methods include, Bagging, Dagging, Adaboost, Multiboost, Decorate and VOTE, see Kotsiantis and Kanellopoulos (2007) for further detail.

the re-sampling accuracies with regards to the iris data were slightly higher at 96.58% (leave-one-out), 95.77% (e0) and 95.79% (0.632B), and with regards to the zoo data they were marginally lower at 91.36% (leave-one-out), 87.89% (e0) and 87.14% (0.632B). Thus, given the results in Table 8.5.4 and the comparisons made with Stefanowski (2007), it is possible to conclude that the predictive accuracy estimates with regards to applying re-sampling within the VPRS framework considered within this dissertation, appear to provide very competitive results (high accuracies).

Considering next, the results of the aggregated β-reducts applied to the set aside validation sets, Tables 8.5.5 and 8.5.6 displays the results with regards to the aggregation of the β-reducts selected from the leave-one-out and bootstrapping analyses of the benchmark data sets, respectively (selected β-reducts shown previously, in table 8.5.2).

Data set	Number of Aggregated Rules	Objects Predicted by Matching Rule	Objects Predicted Correctly	Objects Predicted Incorrectly	Predictive Accuracy (%)
Iris	4	18 (18)	16	2	88.88
Wine	10	25 (25)	25	0	100.00
Zoo	5	8 (10)	8	2	100.00
Cancer	8	333 (336)	315	18	94.59
SPECT	9	57(69)	49	8	85.96

Table 8.5.5: Results of Applying the Aggregated β-reducts Selected from the Leave-one-out Analysis of the Benchmark Data Sets, to the Respective Validation Set

Data set	Number of Aggregated Rules	Objects Predicted by Matching Rule	Objects Predicted Correctly	Objects Predicted Incorrectly	Predictive Accuracy (%)
Iris	3	18(18)	16	2	88.88
Wine	13	25 (25)	21	4	84.00
Zoo	5	8 (10)	8	2	100.00
Cancer	9	334 (336)	316	18	94.61
SPECT	11	63 (69)	53	10	84.12

Table 8.5.6: Results of Applying the Aggregated β-reducts Selected from the Bootstrap Analysis of the Benchmark Data Sets, to the Respective Validation Sets

Again, both Tables 8.5.5 and 8.5.6, indicate a high level of predictive accuracy (under the 'Predictive Accuracy' column), based here though, on the validation sets. It is also perhaps interesting to see, that a high proportion of the objects within all the data sets were predicted by matching rules, and perhaps even more interesting is the small size of the rule sets involved. Indeed, with regards to Table 8.5.6 displaying the results based on the bootstrap aggregated β-reducts, the 18 objects associated with the iris validation set, were classified based only on three simple rules (simple because the associated bootstrap aggregated β-reduct only consisted of two attributes $\{c_3, c_4\}$, see Table 8.5.2). Table 8.5.7 gives a further break down of the validation set results, based on the confusion matrices associated with the analysess of the benchmark data sets.

Data Set	breast-cancer		iris		SPECT		wine		zoo	
Decision Class	1	2	1	2	1	2	1	2	1	2
1	93.85	93.87	100.00	100.00	0.91	0.82	100.00	100.00	100.00	100.00
2	96.62	96.62	100.00	63.63	0.77	0.75	100.00	0.83	100.00	100.00
3	NA	NA	100.00	100.00	NA	NA	66.66	0.83	0.00	0.00
4	NA	NA	NA	NA	NA	NA	NA	NA	100.00	100.00
5	NA	NA	NA	NA	NA	NA	NA	NA	0.00	0.00
6	NA	NA	NA	NA	NA	NA	NA	NA	100.00	100.00
7	NA	NA	NA	NA	NA	NA	NA	NA	100.00	100.00

Table 8.5.7: Breakdown of Leave-one-out (1) and Bootstrap (2) Predictive Accuracies (%) over the Decision Classes Associated with the Benchmark Validation Data Sets

Table 8.5.7 gives the predictive accuracies for both the leave-one-out and bootstrap aggregated β-reducts, displaying the predictive accuracies associated with each individual decision class of the associated benchmark validation sets, for those objects predicted by matching rules[30], (the shaded grey cells indicate no further decision classes are associated with the considered data set). For example, the wine data has three decision classes (labelled in the left most column) with predictive accuracies based on the associated leave-one-out aggregated β-reduct $\{c_1, c_7\}$ of 100%, 100% and 66.66% on the decision classes '1', '2' and '3', respectively.

Within Table 8.5.7 it is interesting to see, that there does not appear to be any bias (in terms of

[30] Results based on those objects predicted by nearest rules were very limited, as most objects were predicted by matching rules. Hence, the results based on nearest rule classification have been omitted as they were felt superfluous, and did not add any value to the analysis being considered.

higher predictive accuracies) towards any particular decision class associated with any of the data sets, in contrast to, the results observed with regards to the FIBR data set. This is most likely because there is a more even distribution of the objects associated with the decision classes of the benchmark data sets, that is, they are more balanced. Furthermore, the high predictive accuracies observed within Tables 8.5.5 to 8.5.7 could be an indication that the benchmark data sets, do capture the condition attributes necessary to discern between the associated decision classes (less so for the SPECT data set), and that this, effects more accurate classifications across all decision classes, and hence, the overall predictive accuracies.

There is an interesting point to be made with respect to the SPECT data set, that is, although it performed poorly with regards to the re-sampling results (with out-of-bag predictive accuracies of 71.88%, 63.42% and 66.37% associated with the leave-one-out, e0 and 0.632B bootstrap out-of-bag predictive accuracies respectively, see Table 8.5.3), it performed much more favourably with regards to the predictive accuracies based on the application of the aggregated β-reducts to the validation sets (with predictive accuracies of 85.96% and 84.12% associated with the leave-one-out and bootstrap aggregated β-reducts, respectively, see Tables 8.5.5 and 8.5.6). It is perhaps not unexpected to have results on the validation set that may be biased (here optimistically), but what is interesting is that the original works by Cios et al (1997), Kurgan et al. (2001) and Cios and Kurgan (2001), from where this data set originates, reported predictive accuracies on their validation set of between 84.00% and 90.40%[31]. These results raise the question, whether their estimates of the predictive accuracies are also biased optimistically, and would the leave-one-out, *k*-fold cross validation or bootstrap out-of-bag estimates of their CLIP3/CLIP4 method be more pessimistic (n.b. the analyst can be more confident that the re-sampling results are more accurate estimates of the true predictive accuracy, see Chapter3 and section 3.3).

As a final benchmark, the results of the leave-one-out and bootstrap analyses from Table 8.5.3,

[31] Interestingly, the reported predictive accuracy of 90.40% was based on an ensemble of their CLIP4 classifier method.

are now compared to a MDA analysis of the benchmark data sets. The MDA analysis was performed using SPSS (http://www.spss.com/), which allowed for estimates of the predictive accuracy to be taken on the training set (i.e. the apparent predictive accuracy) and for a leave-one out analysis to be performed. These comparative results are shown in table Table 8.5.8

Data Set	Leave-one-out VPRS	Bootstrapping e0 (0.632B) VPRS	Apparent Predictive Accuracy, MDA	Leave-one-one MDA
breast-cancer	96.58	95.77 (95.79)	96.10	96.00
iris	96.88	97.9 (97.37)	98.00	98.00
SPECT	71.88	63.42 (66.37)	76.40	68.90
wine	94.78	89.23 (90.28)	98.30	96.60
zoo	91.36	87.89 (87.14)	98.00	93.10

Table 8.5.8: Comparison Between VPRS Re-sampling Predictive Accuracies (%) and MDA

Comparing the results between the leave-one-out VPRS predictive accuracies and the leave-one-out MDA predictive accuracies from Table 8.5.8, it can be seen that MDA outperformed VPRS on three out of the five data sets, namely, iris, wine and zoo. The apparent predictive accuracies associated with the MDA analysis are perhaps, as expected according the the theory in section 3.1, biased optimistically. What should be kept in mind though, is that the results of all the VPRS analyses are based on β-reducts which contain at most only five attributes associated with 11 rules (with regards to the SPECT data set and the β-reduct $\{c_1, c_2, c_3, c_5, c_6\}$) or as little as two attributes associated with three rules (with regards to the iris data set and the β-reduct $\{c_3, c_4\}$). The point here being, that the interpretability, or the simplicity, of the final classifier is, as has been suggested throughout this dissertation, also an important factor which allows the analyst to take confidence in the constructed classifier.

8.6 Summary

The developed VPRS software has shown a high level of predictive performance based on the FIBR data, comparable to, and in some cases better than, other related studies (Poon and Firth, 1999, 2005; Zopoundis and Doumpos, 2002; Doumpos and Pasiouras, 2005). As this dissertation is the first attempt at assessing the predictability of the FIBR data set through a prototype decision support system, it has shown strong evidence that it is possible to design a VPRS system capable of tackling and predicting a difficult data set (i.e. imbalanced with missing data) associated with a level of uncertainty. Predictive accuracy estimates could be improved further, by developing a better strategy to retain more banks from the under represented decision classes during the data collection phase, where many were removed due to missing data values. This would improve the predictability on the under represented decision classes (here the 'A' and 'E' grade banks).

The predictive performance results based on the introduced aggregated β-reduct method, indicated that there may be some improvement based on bootstrap re-sampling (compared to the single run vein graph), particularly with regards to predictive accuracies of individual decision classes. This point reinforces the importance of a transparent breakdown of predictive accuracies over all the decision classes. However, the out-of-bag estimates indicated that the predictive performance based on applying an aggregated β-reduct to the validation set, may be optimistic.

With respect to the FIBR data, the developed software could have been tested and demonstrated on a simpler target problem, that is an application associated with a balanced complete data set, however, the FIBR data has helped bring out more of the real challenges that would be faced by a modern financial analyst. Moreover, as not to mislead the analyst with regards to possible future predictive performance, the transparency within the system combined with the ability of the analyst to have control over the analysis, allows them to have more confidence in the results.

The final section of this chapter indicated that the developed software had a high level of predictive accuracy on a number of benchmark data sets, for both the out-of-bag and validation set predictive accuracy estimates. The results also indicated that the developed software was capable of producing a classifier (aggregated β-reduct with associated rules) that could compete with a widely used and established competitor method, namely, Multi Discriminant Analysis.

Chapter 9

Conclusion and Future Developments

The purpose of this dissertation was to develop a prototype Variable Precision Rough Sets (VPRS) Desicion Support System (DSS). The system encapsulates four of the five stages of knowledge discovery as outlined in Chapter 1, namely, pre-processing, feature selection, data mining, and evaluation. The system was envisaged as being as user friendly as possible, relying heavily on an intuitive point and click interface. The objective of this software, was to make VPRS analysis accessible to an analyst, who wishes to use VPRS as a data mining method, but is not an expert in VPRS.

Although developments within RST and VPRS are ongoing, little research has been done into the utilisation of re-sampling and classifier aggregation within the RST/VPRS framework. Hence, in its role as a DSS, the software developed here, implemented and expanded on re-sampling methods as a means of evaluating the future predictive performance of a classifier (here a list of decision rules), and additionally implemented a system enabling ensemble classification through a novel method of β-reduct aggregation, which has the potential to improve classifier stability and predictive accuracy.

The software was demonstrated mainly through the analysis of the Fitch Individual Bank

Strength Ratings (FIBR), using the CAMELS model (Feldman et al., 2003; Derviz and Podpiera, 2004), a model that has been used extensively throughout the literature relating to bank rating prediction, and is used in reality by the U.S. Federal Reserve Banks as part of an early warning system to identify failing or struggling banks.

The first section of this final chapter summarises the work presented within this dissertation and makes some final conclusions. Section 9.2 discusses the future direction with regards to the role of VPRS within the world of analysing real data sets, and proposes a scheme that has less emphasis on semantic preserving, where semantic preserving may not be appropriate or applicable. It also outlines a number of more specific developments of certain methods employed within this dissertation.

9.1 Summary

This section summaries the work undertaken throughout this dissertation under four subsections. The first three subsections reflect the three main phases of the work, that is, pre-processing and feature selection, the VPRS vein graph analysis, and the VPRS re-sampling analysis; the fourth section relates the software based results associated with the FIBR data set, to the bank rating prediction problem.

9.1.1 Pre-processing and Feature Selection

The theoretical background relating to pre-processing and feature selection, utilised within this dissertation, was presented within Chapter 3. The practical application of the theory implemented within the software was presented in Chapter 6. A number of methods for discretisation of continuous valued data, feature selection, data balancing and missing value imputation were implemented.

Four methods of discretisation were considered, two unsupervised methods, namely, equal-width and equal-frequency, and two more advanced supervised methods, namely, FUSINTER and Minimum Class Entropy (MCE) (Zighed et al., 1998; Fayyad and Irani, 1992). FUSINTER outperformed the other three methods (because of constraints on the size of this dissertation only the results with regards to FUSINTER where shown). It is likely that FUSINTER outperformed the similar MCE method, because FUSINTER is a global discretisation method, that considers all intervals associated with an attribute, whereas MCE only considered adjacent intervals whilst discretising (see Chapter 4 subsection 4.1.1.3).

The developed software implemented two recent feature selection methods, namely ReliefF (Kononenko, 1994), and a method based on RST proposed by Beynon (2004). ReliefF was applied to the attributes both prior to, and after discretisation, the rankings of which where noticeably different (though a high level of correlation existed between them). The software augmented the feature ranking results of ReliefF, with three novel graphs illustrating how the attribute rankings and weightings changed over the number of iterations undertaken (the value *m*, see Chapter 4 subsection 4.2.3.1). It was found that, the random sampling approach to the original Relief algorithm (ReliefF being an extension) suggested by Kira and Rendell (1992), may not be the best approach when dealing with large real world data sets, and that an iterative deterministic approach, where the number of iterations equalled the number of objects in the training set, would be a more efficient alternative, and produce more consistent results (see subsection 6.3.1 for reasoning).

Beynon's (2004) feature selection method, which can be described as a step-wise method (see Chapter 4 subsection 4.2.1.3), appeared to be too impacting as a method, because in a sense, it tended to find a reduct/β-reduct of the full data set, where no further reduction could be achieved within the subsequent data mining stage. Additionally, the method was a suboptimal approach to feature selection, where the selection of an attribute was very much determined by the attributes selected prior to its selection (an extension of this method is described in the future work, subsection 9.1.3). The method was also augmented by two graphs, showing the difference in Quality of Classification (QoC) of a selected attribute subset and the full set of attributes, with regards to the successive inclusion of attributes into the selected subset during the selection process. The results illustrated that, typically, the initially selected attributes were the most impacting, and that the impact on the QoC of the successive attributes, diminished as the process continued. ReliefF had an advantage over Beynon's (2004) method, in that it gave a ranking to all attributes, whereas Beynon's method stopped after a certain criteria was met (where the subset's QoC was an exactitude of the full set's QoC over the whole range of β). However, as a by-product of the first stages of Beynon's method, where basic rankings were calculated for individual attributes, a simpler feature selection method (termed RST_PH1) was demonstrated that was highly correlated with the results of ReliefF.

The ranking results of ReliefF and Beynon's (2004) based approaches were collated into a point and click table that allowed the analyst, based on the information presented to them (graphs, rankings), to select a final set of attributes for the subsequent VPRS analysis. Encouragingly, Harnett and Young (2007) commented that, the flexibility of the developed software that allowed them the final selection of the attributes was important, as they retained their authority, and the additional information available would help them make informed decisions.

Imbalanced data sets and missing value imputation received less focus than the other methods, as they were, initially, less pressing issues. Further, the methods for handling imbalanced data sets and missing values, within the related literature, are less developed than the methods included to facilitate discretisation and feature selection (Weiss and Indurkhya, 1998). Three methods of balancing were implemented, but none appeared affective at tackling the issue of imbalanced data sets. Balancing tended to be detrimental to predictive accuracy results. With regards to the up-balancing method specifically, it greatly affected the performance of the subsequent data mining analysis (in terms of processing time and memory requirements). Two methods for imputing missing data values were implemented, a straight forward mean imputation and a *k*-nearest neighbour method. There was no noticeable difference between the results of the two methods (a full exposition of the results was not shown).

Overall, the pre-processing software, though not the main focus of this dissertation, performed very well (in terms of its impact on later results i.e. increased predictive accuracies). At the initial stages of the software development (first version of the VPRS vein graph software), the pre-processing requirements were undertaken through a number of disjoint programs. The development of the pre-processing software, greatly enhanced the effectiveness of the system, allowing many analyses to be efficiently undertake in a fraction of the time it had previously taken, with improved consistency, and with the ability to save work. The pre-processing software in itself, with the modern approaches it encompasses, is worthy of development into an independent package that can support a wider range of data mining methods.

9.1.2 VPRS Vein Graph Software Analysis

The VPRS vein graph software implemented a graphical interface system, based on Beynon's (2001) vein graph, that allowed the analyst to select from a number of identified β-reducts displayed as "veins" within a graphical panel, using a simple point and click system. Within Chapter 7, the system was initially demonstrated on a small example data set (following the theoretical exposition in Chapter 2), and then a full analysis based on the FIBR data set was presented. With regards to the FIBR data set, evaluation of the classifier's (a rule set induced from a selected β-reduct) performance was presented through the application of the induced rules, on a training set and a validation set derived from the full set of data; where a breakdown of the results were presented separately in a number of information panels

Conforming with the work outlined in Beynon (2001), β-reducts associated with high β-values were concomitant with high Qualities of Approximation (QoA), but lower Qualities of

Classification (QoC), thus demonstrating the inversely proportional relationship (*ibid*). Although, more importantly, this relationship was not demonstrated with regards to the validation set. That is, in contrast to what was expected, β-reducts associated with lower values of β, tended to be associated with higher predictive accuracies on the validation set. They also provided simpler rules and smaller rule sets. β-reducts associated with higher values of β appeared to overfit the training set, being able to achieve high predictive accuracies on the training set but relatively poorer predictive accuracies on the validation set (though still achieving a respectable level of predictive accuracy).

These findings were made possible by the ability to select the β-reducts for comparison, and through the range of results from the evaluation methods implemented within the software. Of the evaluating methods, perhaps the most indispensable tool, providing the analyst with a transparent incite into the true nature of the predictive performance, was the confusion matrix (see Chapter 3 subsection 3.2.1). Implemented within the summary tables (see Chapter 7 section 7.3), the confusion matrix highlighted the despondency between the predictive performance across the decision classes, and aided in the search for a better solution that reflected a better distribution of predictive accuracies.

It is worth mentioning here, that the affect of the validation set selection method, chosen during the initial set-up screen (prior to pre-processing, see Chapter 6 Figure 6.1.1), namely the statistical sub-sampling method, was most recognisable within the confusion matrix (presented in the summary tables). It was found that the statistical sub-sampling method for validation set selection (see Chapter 3 section 3.1.2) performed well (better than stratified and non-stratified random sampling) and to some extent, mitigated the affect of the imbalanced data set (the FIBR data set).

Another factor that aided in the transparency of the presented results, was the breakdown between objects predictable by a matching rule and those predictable using Słowiński's (1992) nearest rule method. This highlighted that, those objects being predicted by the nearest rule were often incorrectly predicted, and that the nearest rule method was performing little better than a random guess. This was an important point, because, had this not been recognised, the predictive accuracy results based on all classifications, may have affected the analyst's confidence in the VPRS model as a data mining method. Other nearest rule type methods exist, which if implemented, may perform better. Such as Słowiński's (1993) method, based on 'valued closeness relation', however this method requires the analyst to place a subjective assessment on the importance of each attribute, hence less suitable for the system implemented here. The further breakdown of the predictions into correctly predicted and incorrectly predicted objects displayed in panels showing; the rules used to predict each object, rules that could correctly predict objects, and rule distances to the objects; although useful during the development of the software, and potentially useful to the analyst seeking an in-depth interpretation of the results (comparison between rules), were rarely required during the FIBR analyses. However, if additional analysis methods were implemented to augment the information, such as individual predictive accuracies for each rule, this would then be useful to the analyst. That is, the analyst could be given the option to exclude "serial offending" rules, i.e. those that were performing badly (consistently classifying objects incorrectly).

The VPRS vein graph software performed satisfactorily, and met our original aim of developing a user interface that was simple to use by the analyst. Although realistically, the VPRS vein graph software was developed as a prototype, it has demonstrated that with more development, VPRS is potentially viable as a data mining solution. Here, the software also played an important role in determining the parameter settings for the subsequent VPRS re-sampling analysis of the FIBR data, such as the β value, and the β-reduct selection criteria's order.

9.1.3 VPRS Re-sampling Software Analysis

The VPRS re-sampling software, constituted the more advanced approach to VPRS data mining, developed within this dissertation. There has been minimal attention given to re-sampling and classifier aggregation within the related RST/VPRS literature, hence this dissertation is the first comprehensive approach to incorporating re-sampling and classifier aggregation within the VPRS framework, and moreover as part of a developed VPRS software package. The VPRS re-sampling software implemented a novel approach to β-reduct selection, that effectively allowed the automation of the vein graph analysis for application within a re-sampling environment (i.e. through the described criteria selection process). An original method for β-reduct aggregation was also described and implemented, with the intention of stabilising and optimising the decision rules.

The software implemented three re-sampling methods to enable improved estimations of a set of decision rules predictive performance (predictive accuracy). In terms of bias and variance, the re-sampling methods reflected the trends as seen within the related literature, when applied to other data mining methods. That is, leave-one-out was unbiased, but was associated with high variance; *k*-fold cross-validation was slightly more biased, but was associated with less variance; the *e*0 bootstrap was biased pessimistically (when compared to leave-one-out) but had low variance; and the 0.632B bootstrap, a convex combination based on the *e*0 bootstrap value and the apparent predictive accuracy, as expected, was less pessimist than the *e*0 but still maintained a low variance.

The introduced β-reduct aggregation process, allowed the analyst to aggregate β-reducts associated with equivalent subsets of attributes (β-reducts with identical condition attributes). Three graphs were developed to aid the analyst's choice of β-reducts to aggregate. The first graph showed the distribution frequency of occurrence of the top ten most frequently occurring β-reducts (selected by the criteria during the re-sampling process). The second graph showed the frequency of occurrence of β-reduct size (number of attributes associated with the β-reducts). The third graph showed the frequency of occurrence of each individual attribute, associated with the β-reducts selected by the criteria during the re-sampling process. Additionally, a summary table provided the analyst with summary statistics relating to all the selected β-reducts, and a second more detailed summary table displayed summary statistics for the top ten most frequently occurring β-reducts.

The graphs and the associated summary tables indicated that, selection of the "best" set of *β*-reducts to aggregate, did not necessarily imply the most frequently occurring *β*-reduct should be selected. That is, with regards to the bootstrap analysis, the size (number of condition attributes) of the most frequently occurring *β*-reduct did not reflect in the most frequently occurring *β*-reduct size. Hence, the full range of results had to be considered during the *β*-reduct aggregation process. Additionally, the graphs highlighted the asymptotic nature between the results of the Leave-one-out analysis and the *k*-fold Cross-validation analyses (for stratified *k*-folds, $k = 10, 20, 30, 40$ and 50), but highlighted respectively, that the results associated with bootstrapping were more diverse.

The end product of *β*-reduct aggregation was a list of aggregated decision rules. The analyst, through a table and tick box interface, had the final choice of which rules they wished to be used for classification. The analyst's choice was aided by metrics associated with the decision rules, such as aggregated *β*-reduct strength, and by a summary table which listed summary statistics of the rules associated with the top ten most frequently occurring *β*-reducts. On this matter, Harnett and Young (2007) commented on the importance of the analyst being able to choose the final set of rules, allowing the flexibility to remove erroneous rules or tailor the set to their requirements, e.g. taking a small set of the most general rules.

Consistent with the VPRS vein graph analysis, where the set of rules associated with a selected *β*-reduct were applied to a validation set, the selected aggregated *β*-reduct and its associated aggregated rules were also applied to the validation set, and the results were evaluated using a number of summary tables and evaluation methods.

In summary, the VPRS re-sampling software, performed well. In particular, the re-sampling predictive accuracy estimates provided a more transparent indication of future predictive performance. Used as out-of-bag estimates for *β*-reduct aggregation, they indicated that the predictive accuracy associated with the validation set was overly optimistic, perhaps because the validation set did not contain many objects from the under represented decision classes. The performance of β-reduct aggregation in terms of stabilisation and optimisation of the β-reducts and their associated decision rules was inconclusive. Though there was some evidence, comparable with the closest related work by Stefanowski (2004, 2007), that suggested there could be some performance improvement, particularly when considering the decision classes' individual predictive accuracies (as shown in the confusion matrix, see Chapter 8 section 8.4). Again, this supports the benefit and effectiveness of the confusion matracies within the summary tables, to identify possibly unforeseen aspects of classifier performance.

9.1.4 Inference on the Bank Credit Rating Problem

The focus of application and research for the software developed within this dissertation was the credit rating prediction problem, more specifically prediction of bank ratings. An analyst may wish to predict bank ratings, for a number of different reasons. Firstly, a number of agencies, such as the U.S.A.'s Federal Insurance Deposit Company (FIDC), or central banks, are interested in developing early warning systems to identify struggling banks, before they require outside assistance (Feldman et al., 2003; Krainer and Lopez, 2003; Derviz and Podpiera, 2004). Secondly, investment companies may be interested in predicting the ratings of banks, that have not solicited ratings from the major rating agencies or, have ratings given to them by potentially biased or less reputable agencies (such is the case in China) (Kennedy, 2003; Poon, 2003; Poon and Chan, 2008). Thirdly, the main rating agencies do not disclose their rating process, and moreover claim that quantitative models cannot capture the qualitative aspects of the rating process. However, any incite into the rating process could provide companies with a basis to improve their operations and encourage a more favourable rating (investment grade or above) (Shin and Han, 2001; Kim and Sohn, 2008).

The specific type of bank rating investigated here, was Fitch's Individual Bank strength Rating (FIBR). The FIBR is a rating which is issued to banks globally. Hence, whilst developing our attribute rationale (initial attribute selection), based on the established CAMELS model (Gilbert et al., 2000; Krainer and Lopez, 2003), a number of attributes were included to capture this global factor. Indeed, during the pre-feature selection (ReleifF, and RST_FS), and during the data mining process (vein graph and re-sampling analyses), it was evident that GDP/head was a strong factor in determining a bank's rating, acting almost as a global discriminant. Of the other attributes, Impaired Loans/Gross Loans appeared to act as another strong discerning factor, where a pertinent example was given with regards to the U.S.A.'s recently failed IndyMac bank (Chapter 7 section 7.2), which three months prior to its collapse reported that its impaired loans had reached $1.85 billion, an increase 40.56% from the previous quarter.

In hindsight, it would have been of value to seek the opinion of a third party who is an expert in the field of bank credit rating prediction, to give feedback on the findings of the VPRS analysis of the FIBR data set undertaken within this dissertation. Perhaps the best course of action would have been to seek the opinion of a government agency who utilises such a system, such as, one of the U.S.A.'s Federal Reserve Banks, who regularly publish work relating to bank rating prediction (see section 5.3), or the U.K.'s Financial Services Authority who use the RATE system (Risk Assessment, Tools of Supervision and Evaluation) to monitor Bank Strength. Sahajwala and Van den Bergh (2000), give an overview of some of the most prominent early warning systems used by governments and other bodies worldwide.

9.2 Future Work

The future work presented here, provides a number of potential developments that could be implemented within the software, and some theoretical considerations with regards to the development of RST\VPRS. With the many facets currently involved within the developed software, additional functionality would have increased the development and testing times almost exponentially (i.e. more analysis paths requiring testing), and would be beyond the scope which could have feasibly been undertaken and developed during this dissertation, and hence are only considered here, as future work. Additionally, the potential of some aspects of the future work, was only realised or considered during the development and implementation of the developed software.

This section is split into two subsections. Subsection 9.2.1, discusses some issues relating to the reality of attempting to apply the RST/VPRS concept of semantic preserving, that is reducts/β-reducts, to real world data sets. Subsection 9.2.2 goes on to describe a number of specific developments that could potentially enhance the system, and improve the approaches to KDD implemented within the developed software.

9.2.1 The Reality of VPRS Semantic Preserving with Regards to Real World Data

Within Chapter 2, reducts and β-reducts were described as subsets of attributes which maintain the meaning, or semantics of the full set of attributes, by maintaining the dependency (the QoC) between the condition attributes and the decision attributes (over a range of β for β-reducts). Having undertaken a VPRS analysis on a "real world" data set, it has raised potential doubts over the true meaning of semantic preserving in terms of the scaled up problem. That is, does the system truly preserve the "meaning" of the data.

Beynon's (2001) description of a hidden β-reduct, that is, a subset that had a higher QoC than the full set of attributes over a range of β, essentially indicated, that a subset of attributes could be found that performed better in terms of possible future predictive accuracy than the full set of attributes. However, for certain values of β (in the hidden range), the subset of attributes would be rejected as a β-reduct under Ziarko's (1993a) definition of a β-reduct, i.e. it did not maintain the QoC.

With regards to the notion of maintaining the dependency (QoC) (semantic preserving), Mi et al. (2003), using the example originally given in Beynon (2001), showed that the derived rules from a

β-reduct may be in conflict with the original system. Essentially, the conflict arises whereby, a β-reduct maintains the QoC with the full set of attributes but does not maintain the condition class distribution (number of condition classes and number of objects within those decision classes), which can result in a different set of derived decision rules to the original system. Hence, in contrast with the original concept of a reduct, the β-reduct may not preserve the semantics of the original data set, at least in terms maintaining the decision rule structure.

It could be augured that by maintaining the QoC, the system is insuring that the subset of attributes does not do worse than the full set of attributes in terms of possible future predictive performance (Liu and Motoda, 2000, describes this as consistency). However, based on that argument, a response may be, then why not find a subset that increased the QoC? effectively finding a subset that could potentially improve the future predictive performance. Indeed, this approach is much more in line with the conventional wisdom of feature section within data mining, that seeks to find the best subset of attributes in terms of improving the future predictive performance (Liu and Motoda, 2000) (there are other issues that would need consideration, such as, overfitting and decision rule interpretability). This essentially epitomises, the core problem feature selection seeks to solve, that is, not knowing which attributes to select from the source data set. Han and Kamber (2006, pp. 75) state quite succinctly:

> "...keeping in irrelevant attributes may be detrimental, causing confusion for the mining algorithm employed. This can result in discovered patterns of poor quality."

Relating the above quote to the issue of reducts and β-reducts, it implies that, where we are unsure of the relevance of all the selected attributes, one cannot assume that a subset of attributes (a reduct) cannot do better than the full set. Hence, with this in mind, it would be worth reconsidering the definition of a β-reduct, with regards to real world data (specifically with relation to data mining), where the relevance of the selected attributes is unknown. That is, it would be interesting to investigate as future work, a β-reduct defined as having a QoC equal to, or where possible, greater than the full set of attributes.

9.2.2 Other Future Developments

The issue of reducts/β-reducts and semantic preserving within real world data sets, outlined in the previous subsection, is perhaps here, the most impacting issue with regards to classifier performance, which if addressed, could potentially improve performance. This subsection outlines a number of other future developments, that may enhance the performance or provide additional support to the analyst.

- **Re-sampling Extension of the RST Feature Selection Method**. As stated in Chapter 4, the RST based feature selection method proposed by Beynon (2004) was a sub-optimised approach to feature selection. Unfortunately, it was found to be too impacting as a feature selection method, finding a subset of the data set's condition attributes that was akin to being a reduct. The method could be improved by wrapping it within a re-sampling framework, and recording average results, such as feature subset size, occurrence of condition attributes etc. This would represent a heuristic approach, as the results may differ for each independent application of the method on the same data set, and would also constitute an increase in pre-processing time.
- **Re-sampling Extension of the Confusion Matrix**. The confusion matrix proved an indispensable evaluation asset to the VPRS data mining software developed here. However with regards to the VPRS re-sampling software, although the re-sampling predictive accuracy estimates (out-of-bag estimates) appeared more credible than the validation set's predictive accuracy estimates, it would have been beneficial to have seen a breakdown of re-sampling predictive accuracy estimates over all decision classes. This would allow the analyst some much needed incite into the possible predictive performance on the under represented decision classes. This extra level of transparency (breakdown of the out-of-bag estimates), has not been implemented in any previous studies (as far as we are aware), and may prove an interesting new addition to the field of predictive accuracy estimation. In essence, the idea constitutes the combining of the confusion matrix (as described in Chapter 3 section 3.2), with the ideas of re-

sampling for predictive accuracy estimation.

- **Handling Imbalanced Data.** The issue of imbalanced data has been prominent throughout the analyses undertaken within this dissertation, but the methods implemented in the developed software for dealing with imbalanced data performed poorly. Perhaps the main problem afflicting the balancing methods is the arbitrary affect that they have on the underlying distribution of the decision classes. That is, the balancing methods artificially and arbitrarily alter the decision class distribution, with no regard to the importance attached to any one decision class. The medical diagnoses example is often quoted (Coppin, 2004), as an application of balancing where the data is artificially altered to improve prediction of patients at risk of being ill, at the expense of predicting healthy patients as being ill. However, this poses the question, how can the analyst select the amount of balancing that is required? Furthermore, this approach further breaks down when considering multiple decision class problems, where it is difficult to place an importance value (how much to up or down-balance) on different decision classes.

The answer perhaps, is not a question of balancing the data, but the ability to assign risk and cost analyses to the decision classes. This is very much related to the FIBR analysis, and indeed Harnett and Young (2007) suggested that an important issue for them was a break down of the risks, costs, and profits associated with decision rules incorrectly predicting banks to the other decision classes. Risk could be defined through the associated rule measurements such as rule strength and certainty, whereas costs could be factored in by the analyst who would know the cost of incorrectly predicting a bank to another decision class, and the profit associated with predicting it correctly. This system would then allow the analyst to "hedge their bets".

Dealing with the problem of imbalanced data, is often the reality with respect to many large real world data sets, and perhaps the best approach is to develop a robust KDD system (good pre-processing, feature selection, evaluation, optimisation), which mitigates the affects of imbalanced data. As stated, it may be the reality that not enough data exists to perform credible predictions on under represented decision classes, but this is also the reality for the analyst who themselves may struggle to make credible predictions under such circumstances.

- **Utilising Basic Financial Attributes as Opposed to Final Ratios**. As stated in Chapter 5, financial ratios have been used since the early 1890's as a method of comparison between financial institutions. What is proposed here, is that, as modern data mining methods such as VPRS can recognise dependency between attributes, then it may be plausible to use the basic attribute values. For example, using the attributes net income and net loans independently, as opposed to the ratio net income/net loans, because in theory the system would be able to recognise the dependency between the attributes and the final rating classification. The advantage of this system, would be to reduce the potential set of attributes to select from, at the initial data selection stage of KDD.
- **Implementing Dominance Based RST to Mitigate the Requirement of Discretisation, and as an Approach to Handling Missing Data**. Dominance Based Rough Set Approach (DRSA) is an extension of RST to accommodate the ordinal properties of data with regards to decision problems (Greco et al., 2001, 2005). It considers the monotonicity relation between attributes, that is, the mutual relationship between the attributes (*ibid* gave inflation rate and interest rate as an example of this relationship). DRSA substitutes the idea of the indiscernibility relation in RST, by a dominance relation. Within DRSA, an object is not just categorized by its condition values presence or absence within a concept (condition class), but by its degree of presence or absence. The decision rules from DRSA have a more informative syntax to those presented by RST, and would read, for example, "If object y presents attribute a_1 in degree at least h_1, and attribute a_1 in degree at least h_2, then the object y belongs to set X in degree at least α".

The advantage of DRSA is that it operates on highly granular (continuous) data, and mitigates the need for discretisation (other extensions to RST also have these properties, such as Fuzzy-

Rough sets, see Jensen and Shen, 2004, 2008). Recent work by Dembczyński et al. (2007), proposed a model for combining aspects of VPRS and DRSA, and Yang et al. (2008) has proposed a system for data reduction within the DRSA framework. The consideration here, is, the development of DRSA in both the vein graph and re-sampling environments would be a natural progression to the developed software.

With regards to the issue of missing data, the methods implemented for handling missing data within the FIBR data set were adequate. However, a proportion of the objects (banks) available within the source FIBR data base, were removed from the analyses because they contained too much missing data (attribute values). When dealing with under represented decision classes, this loss of data can and was, detrimental to the data mining process (i.e. affecting the predictability of the under represented decision classes). Greco et al. (1999) proposed an involved approach to handling missing data, an extension to the original RST and their DRSA. Their system described the use of exact and approximate rules, depending on whether they were supported by consistent or inconsistently classified objects. They also state that rules are robust, were they are supported by one or more objects with no missing values associated with the condition attributes.

- **Implementation of Alternative Nearest Rule Prediction Methods.** There were, a number of occasions, with regards to the VPRS analysis considered in this dissertation, where objects had to be predicted by the nearest rule method, specifically Słowiński's (1992) method, because there was no rule associated with a selected β-reduct/aggregated β-reduct's rule set that had exact matching condition attributes values. The results based on the implemented nearest rule method were typically quite poor. Hence, a number of other approaches could be considered as future alternatives, such as Manhattan distance (Han and Kamber, 2006), or a method based on interpolation (Huang and Shen, 2006, 2008).

9.3 Closing Remarks

The software developed here, and theoretical aspects introduced within this dissertation, have provided the first comprehensive foundation for the integration of re-sampling and classifier aggregation within the VPRS framework. It has also illustrated the potential for developing fully integrated, intuitive, data mining software, utilising VPRS as the data mining solution. It is envisaged that the work undertaken here can act as a benchmark for future studies, and can facilitate further research within the field of VPRS; possibly through expansion of the developed software.

The introduction of this dissertation initially opened with the link between the origin of information and the beginning of recorded history. It is perhaps pertinent then, to close with some remarks on the future of the information industry, more particular, exploitation of information with regards to the financial markets.

The opening quote in Chapter 1, and the comments made by Aburdene (2005), suggested that if businesses were to remain competitive, then the future was in the hands of the concious individual, who could design the "killer app" (*ibid*) to exploit the abundance of information, and like the designers of successful applications that have gone before them, would launch a multi billion dollar industry. Indeed, a search of the web for quantitative analysis related occupations, will show the disparity between the supply of individuals capable of designing such systems and the huge demand within the financial services industry to employee them. This demand and the desire for competitive edge, is understandable when considering the observations made by Hull (2006), who stated that, the exchange traded and the over the counter markets combined, were worth around $260 trillion. To put this value into perspective, Hull observes that this figure is over five times the world's gross domestic product.

Perhaps software, such as that developed within this dissertation, or at the very least, the issues raised within the development of the software, will in some sense, show the way forward for the future. Giving those industries that are willing to risk and exploit new ideas and technology, the competitive edge to keep, or even expand their share of that $260 trillion industry.

Appendix A

This appendix presents in sections A1 to A4, a number of graphs expositing the convergence of the *k*-fold cross-validation analyses, to the leave-one-out analysis; an expansion of the results shown in Chapter 8 sections 8.1.1 and 8.2. Here, the graphical analyses have been expanded in two directions, firstly an increased range of folds are considered (*k* = 10, 20, 30, 40, 50), and secondly, consideration is given to both stratified and non-stratified *k*-fold cross-validation. The result illustrate that stratified *k*-fold, converges relatively quicker than non-stratified *k*-fold (as stated in section 8.2). Section A5 presents further tables of results relating to the selected *β*-reducts associated with the leave-one-out and bootstrapping analyses.

A breakdown of the sections within this appendix is provided below:

- Section A1. This section presents the non-stratified *k*-fold cross validation graphs, associated with the frequency of occurrence of the *β*-reducts selected during VPRS re-sampling analyses, for *k* = 10, 20, 30, 40 and 50 folds.
- Section A2. This section presents the non-stratified *k*-fold cross validation graphs, displaying the frequency of occurrence of the condition attributes associated with the *β*-reducts selected during VPRS re-sampling analyses, for *k* = 10, 20, 30, 40 and 50 folds.
- Section A3. This section presents the stratified *k*-fold cross validation graphs, associated with the frequency of occurrence of the *β*-reducts selected during VPRS re-sampling analyses, for *k* = 10, 20, 30, 40 and 50 folds.
- Section A4. This section presents the stratified *k*-fold cross validation graphs, displaying the

frequency of occurrence of the condition attributes associated with the β-reducts selected during VPRS re-sampling analyses, for k = 10, 20, 30, 40 and 50 folds.

- Section A5. This section present tables of information relating to the prediction of banks within the FIBR validation set, by nearest rule method; based on all aggregated β-reducts associated with the leave-one-out and bootstrapping analyses.

A.1 Non-stratified *k*-fold Cross-validation Graphs Associated with the Frequency of the Selected β-reducts, with Regards to the FIBR Data Set

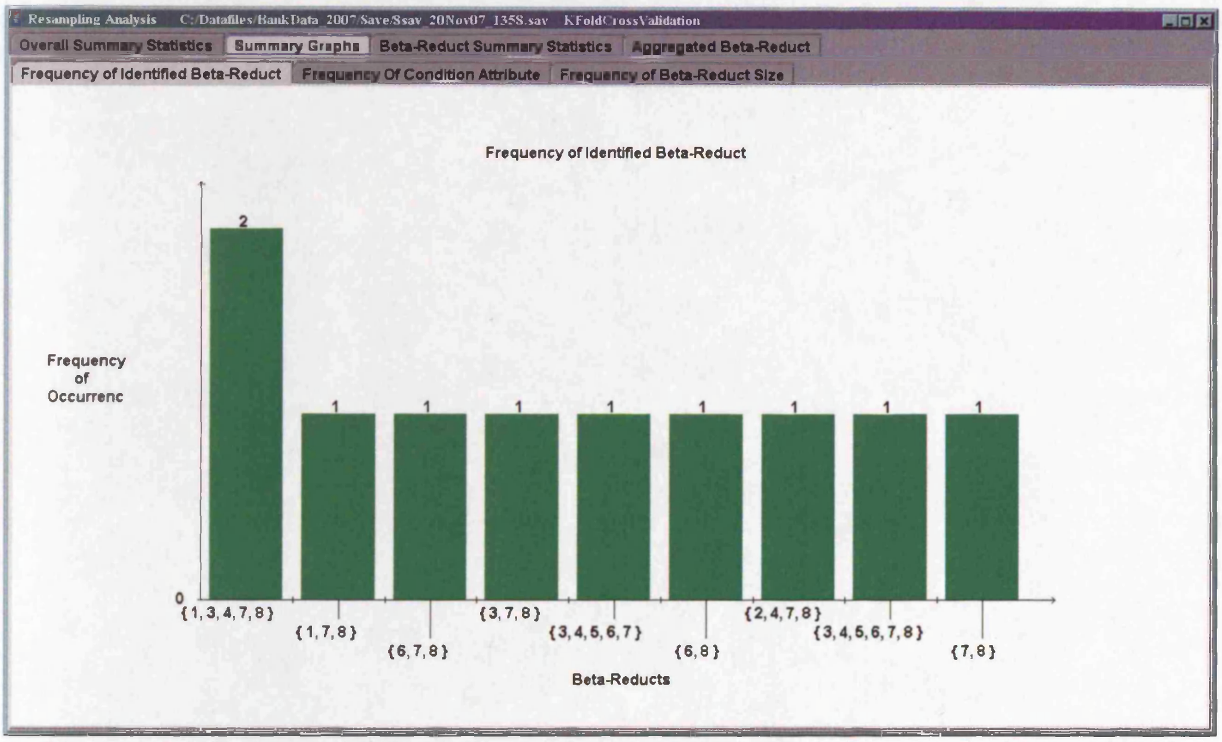

Resampling Analysis C:/Datafiles/BankData_2007/Save/Ssav_20Nov07_135S.sav KFoldCrossValidation
Overall Summary Statistics
Summary Graphs
Beta-Reduct Summary Statistics
Aggregated Beta-Reduct
Frequency of Identified Beta-Reduct
Frequency Of Condition Attribute
Frequency of Beta-Reduct Size
Frequency of Identified Beta-Reduct
Frequency of Occurrenc
0
2
1
1
1
1
1
1
1
1
{1,3,4,7,8}
{1,7,8}
{6,7,8}
{3,7,8}
{3,4,5,6,7}
{6,8}
{2,4,7,8}
{3,4,5,6,7,8}
{7,8}
Beta-Reducts

Figure A.1.1: Frequency of Occurrence of the Selected β-reducts, Associated with the 10-fold Cross-validation Analysis of the FIBR Training Set

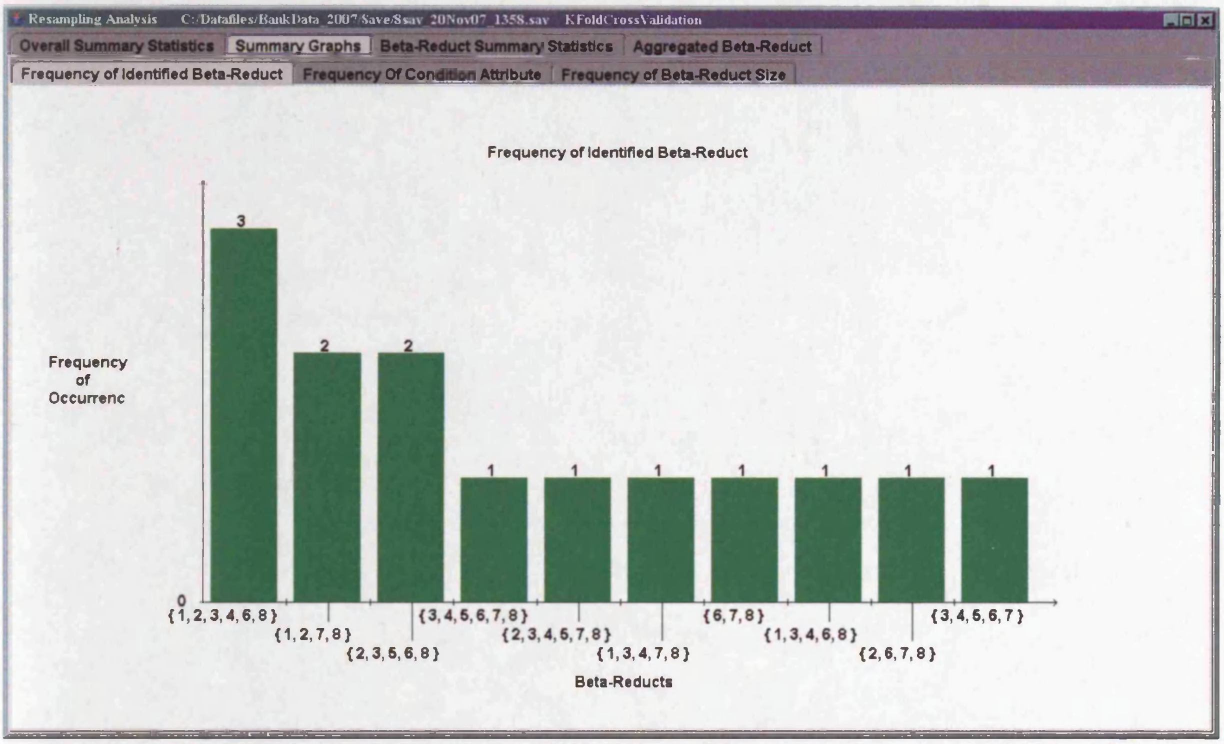


Figure A.1.2: Frequency of Occurrence of the Selected β-reducts, Associated with the 20-fold Cross-validation Analysis of the FIBR Training Set

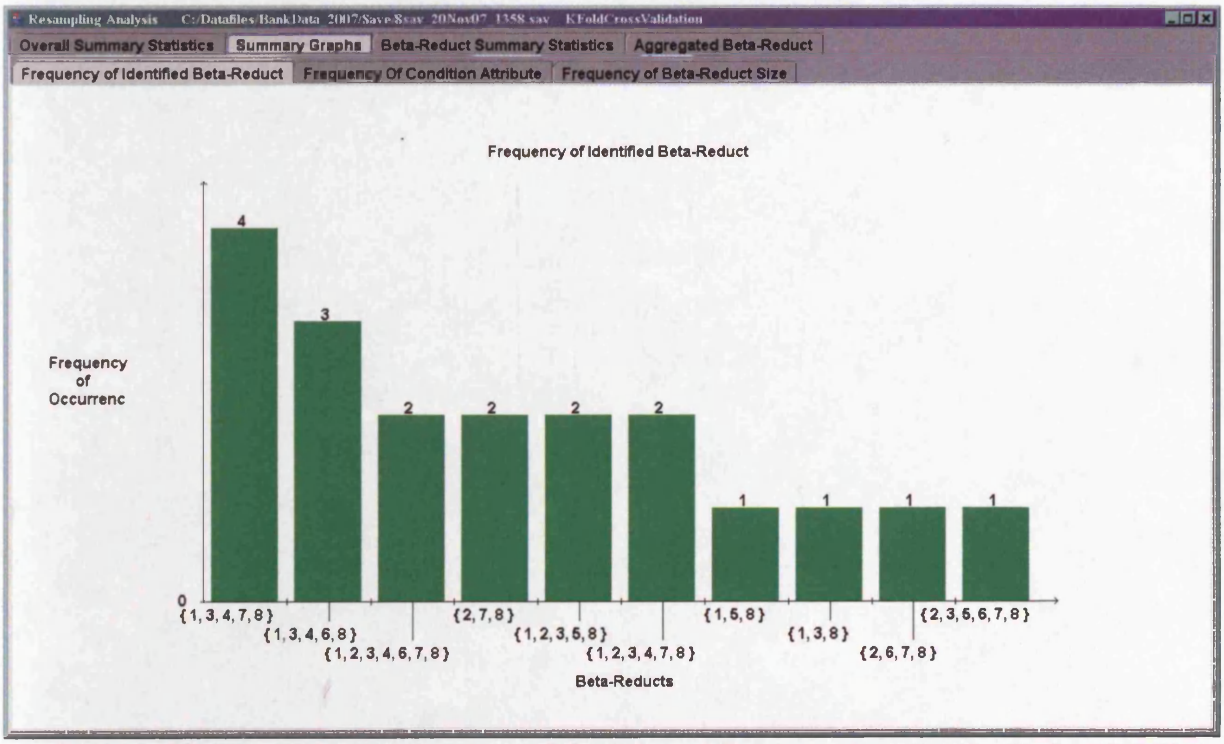


Figure A.1.3: Frequency of Occurrence of the Selected β-reducts, Associated with the 30-fold Cross-validation Analysis of the FIBR Training Set

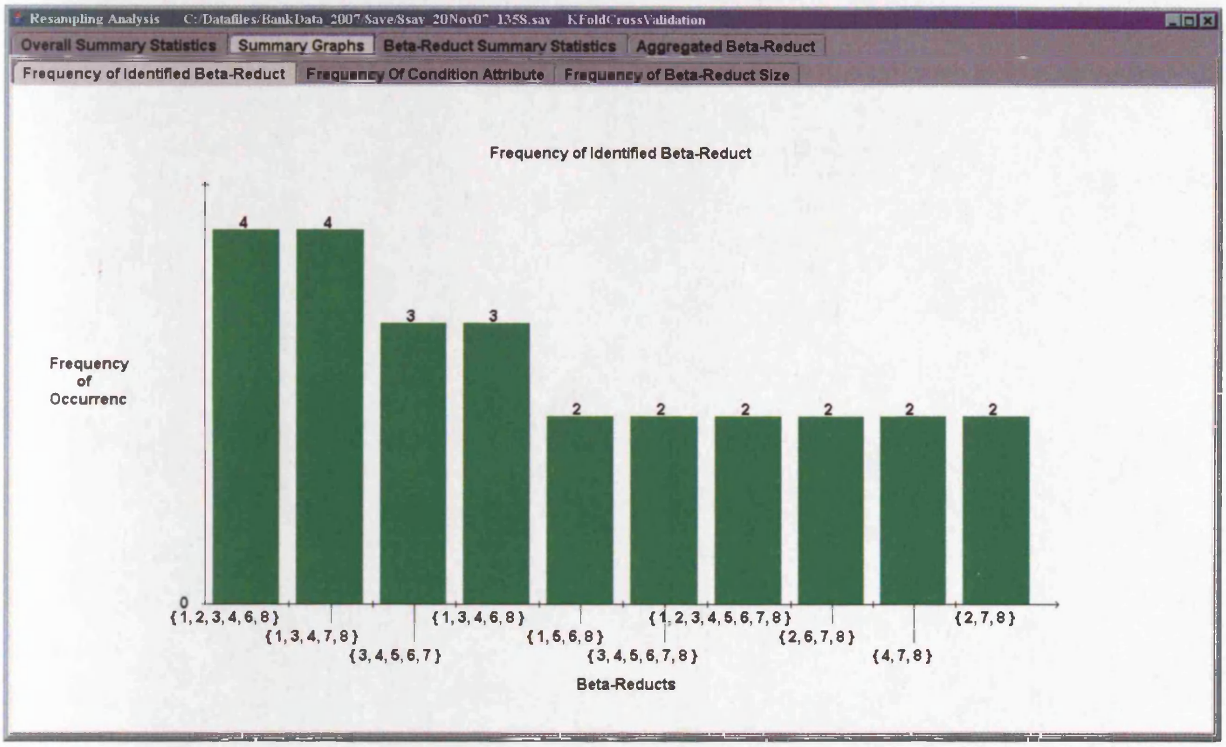


Figure A1.4: Frequency of Occurrence of the Selected β-reducts, Associated with the 40-fold Cross-validation Analysis of the FIBR Training Set

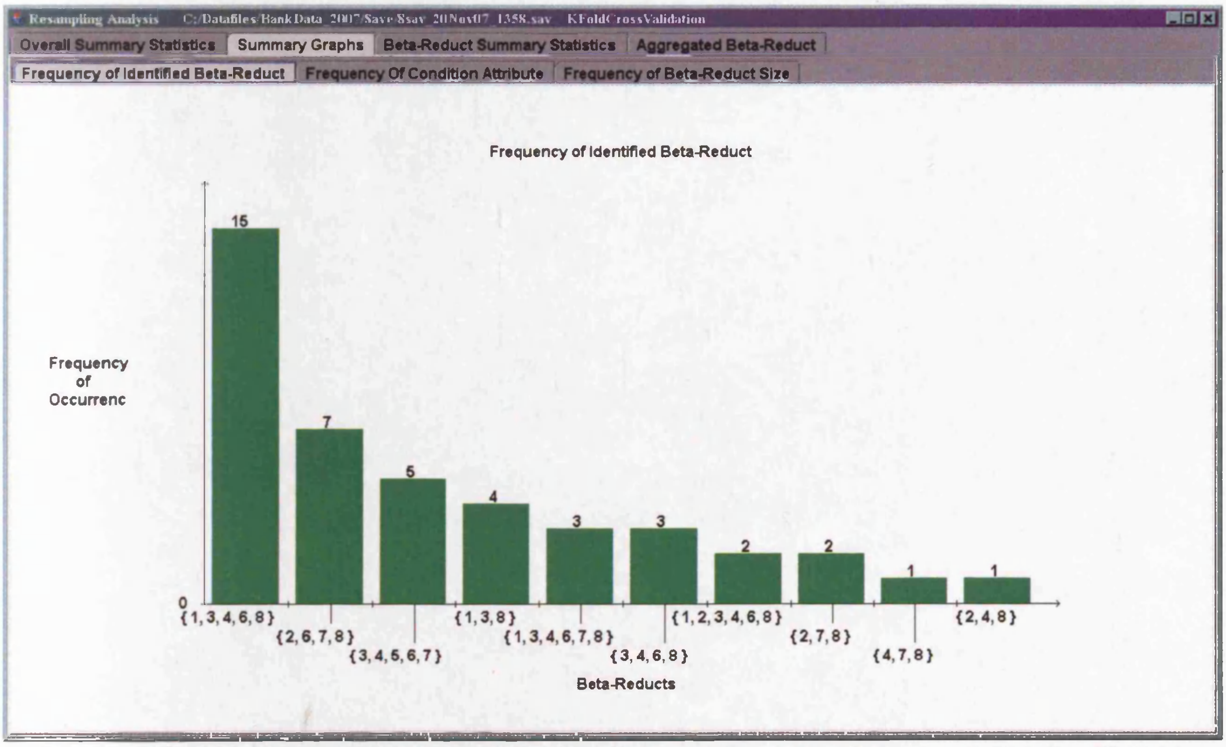


Figure A.1.5: Frequency of Occurrence of the Selected β-reducts, Associated with the 50-fold Cross-validation Analysis of the FIBR Training Set

A.2 Non-stratified *k*-fold Cross-validation Graphs Associated with the Frequency of Occurrence of Condition Attributes, with Regards to the FIBR Data Set

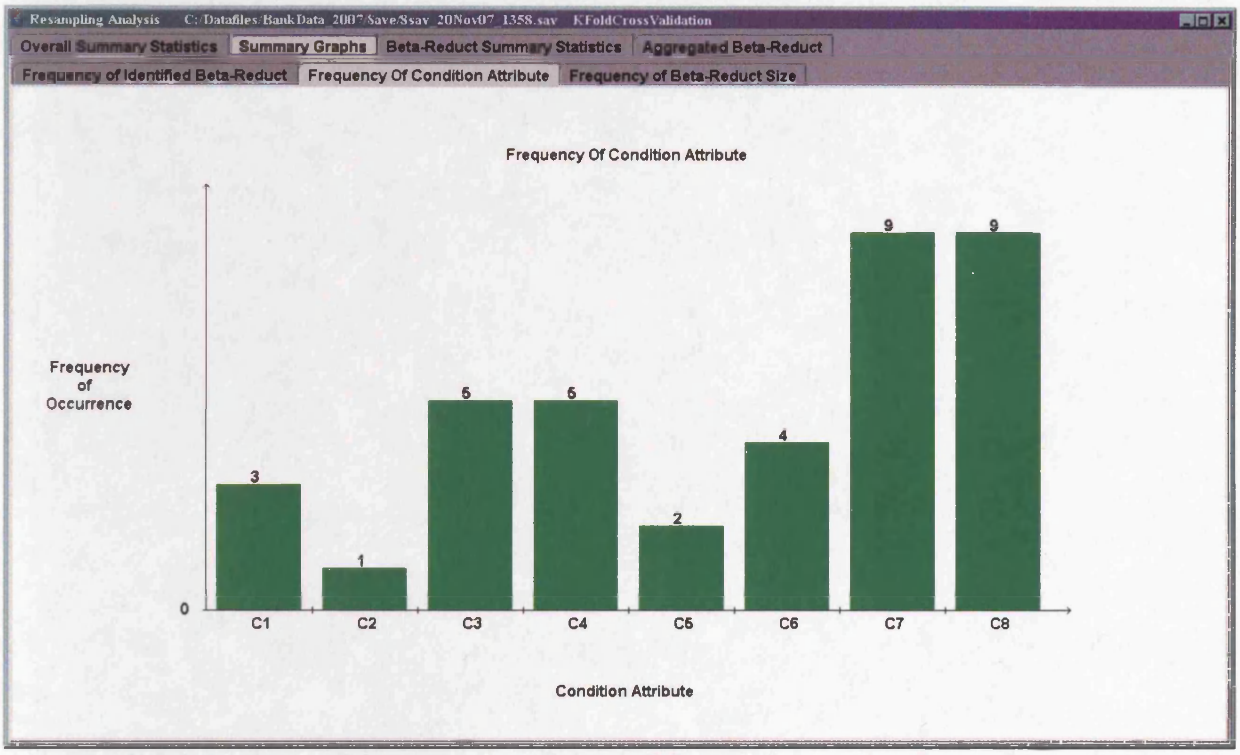


Figure A.2.1: Frequency of Occurrence, of the Condition Attributes Associated with the Selected β-reducts, with Regards to the 10-fold Cross-validation Analysis of the FIBR Training Set

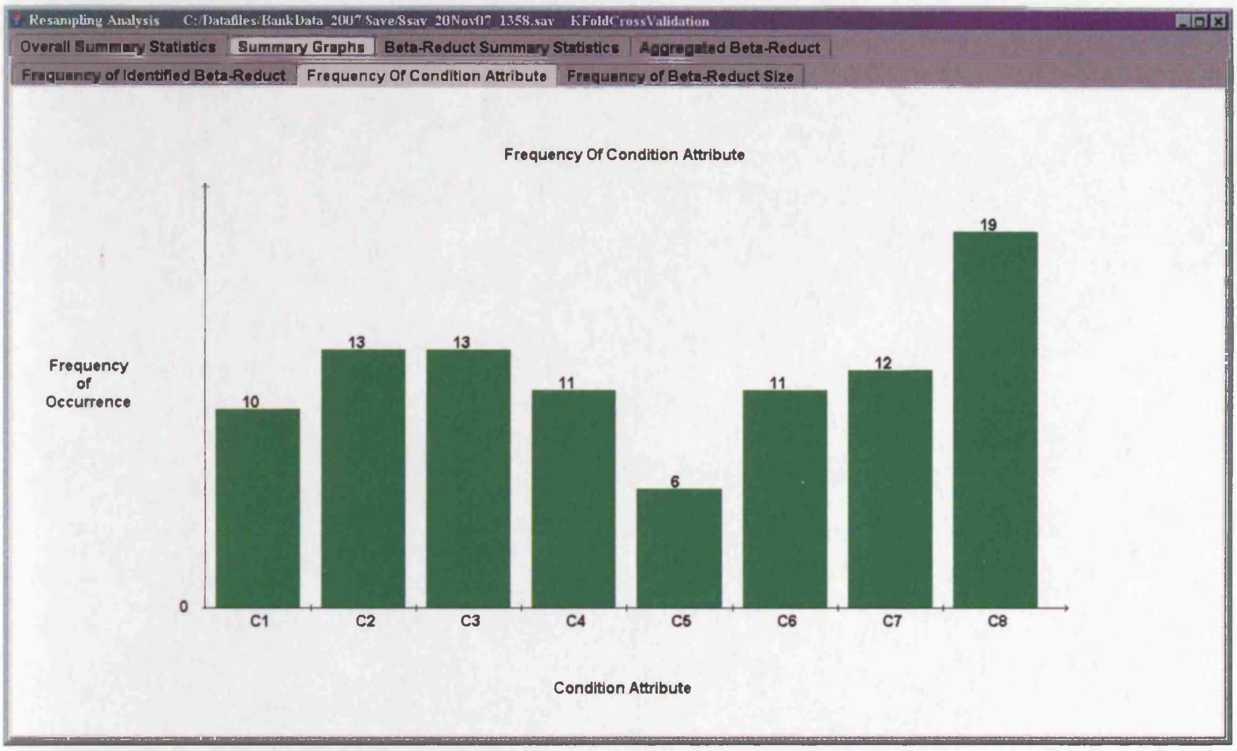

Resampling Analysis C:/Datafiles/BankData 2007/Save/Ssav 20Nov07 1358.sav KFoldCrossValidation
Overall Summary Statistics
Summary Graphs
Beta-Reduct Summary Statistics
Aggregated Beta-Reduct
Frequency of Identified Beta-Reduct
Frequency Of Condition Attribute
Frequency of Beta-Reduct Size
Frequency Of Condition Attribute
Frequency of Occurrence
0
10
13
13
11
6
11
12
19
C1
C2
C3
C4
C5
C6
C7
C8
Condition Attribute

Figure A.2.2: Frequency of Occurrence, of the Condition Attributes Associated with the Selected β-reducts, with Regards to the 20-fold Cross-validation Analysis of the FIBR Training Set

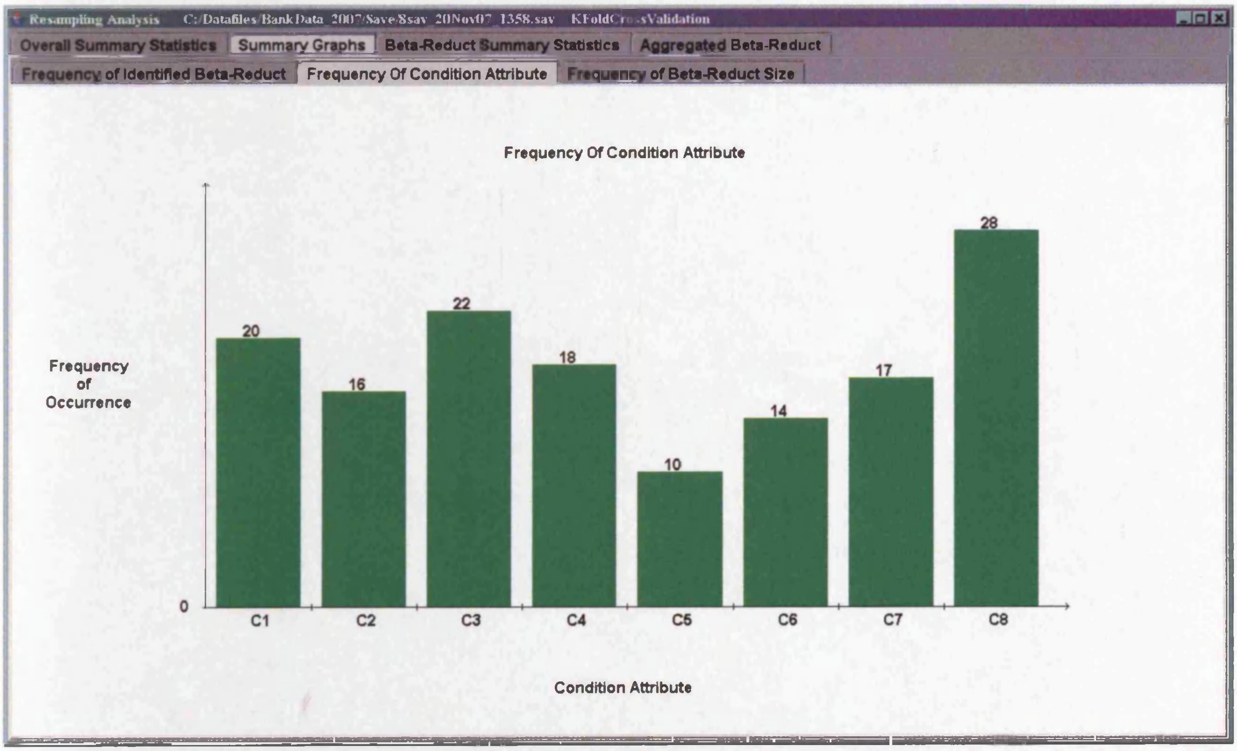

Resampling Analysis C:/Datafiles/BankData 2007/Save/Ssav 20Nov07 1358.sav KFoldCrossValidation
Overall Summary Statistics
Summary Graphs
Beta-Reduct Summary Statistics
Aggregated Beta-Reduct
Frequency of Identified Beta-Reduct
Frequency Of Condition Attribute
Frequency of Beta-Reduct Size
Frequency Of Condition Attribute
Frequency of Occurrence
0
20
16
22
18
10
14
17
28
C1
C2
C3
C4
C5
C6
C7
C8
Condition Attribute

Figure A.2.3: Frequency of Occurrence, of the Condition Attributes Associated with the Selected β-reducts, with Regards to the 30-fold Cross-validation Analysis of the FIBR Training Set

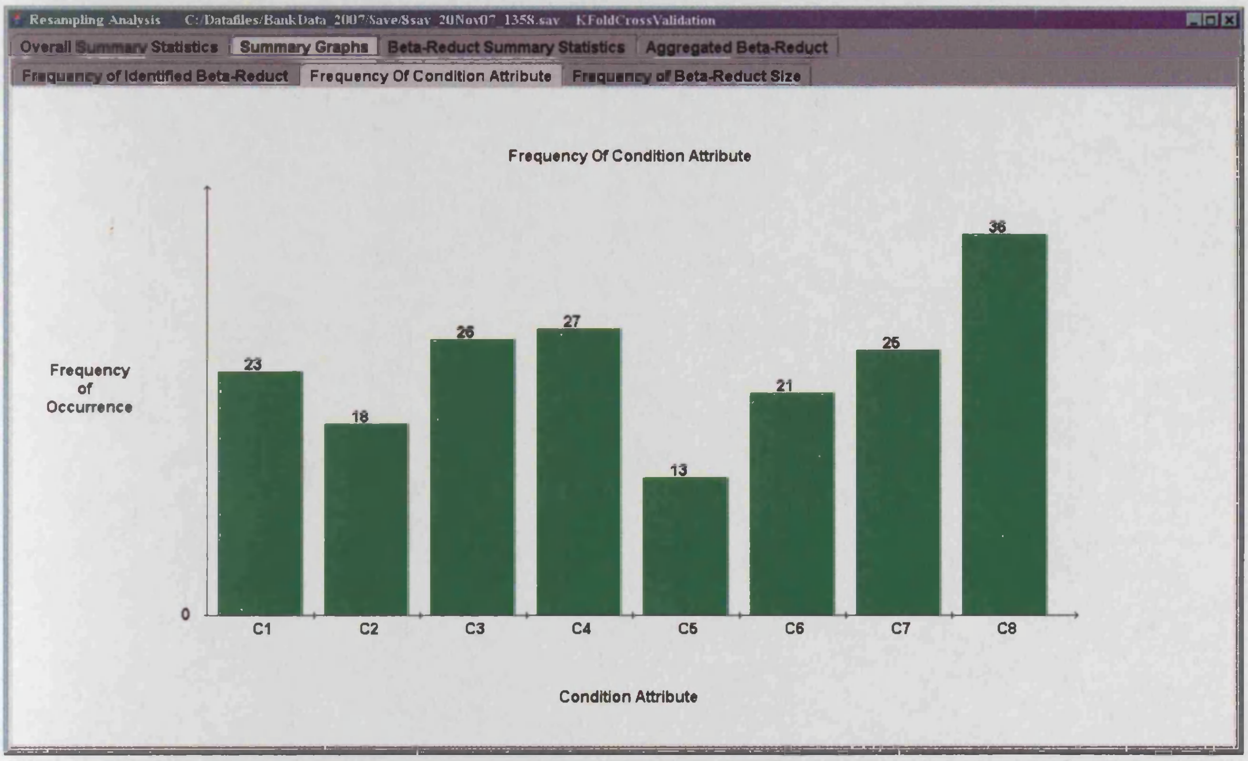


Figure A.2.4: Frequency of Occurrence, of the Condition Attributes Associated with the Selected β-reducts, with Regards to the 40-fold Cross-validation Analysis of the FIBR Training Set

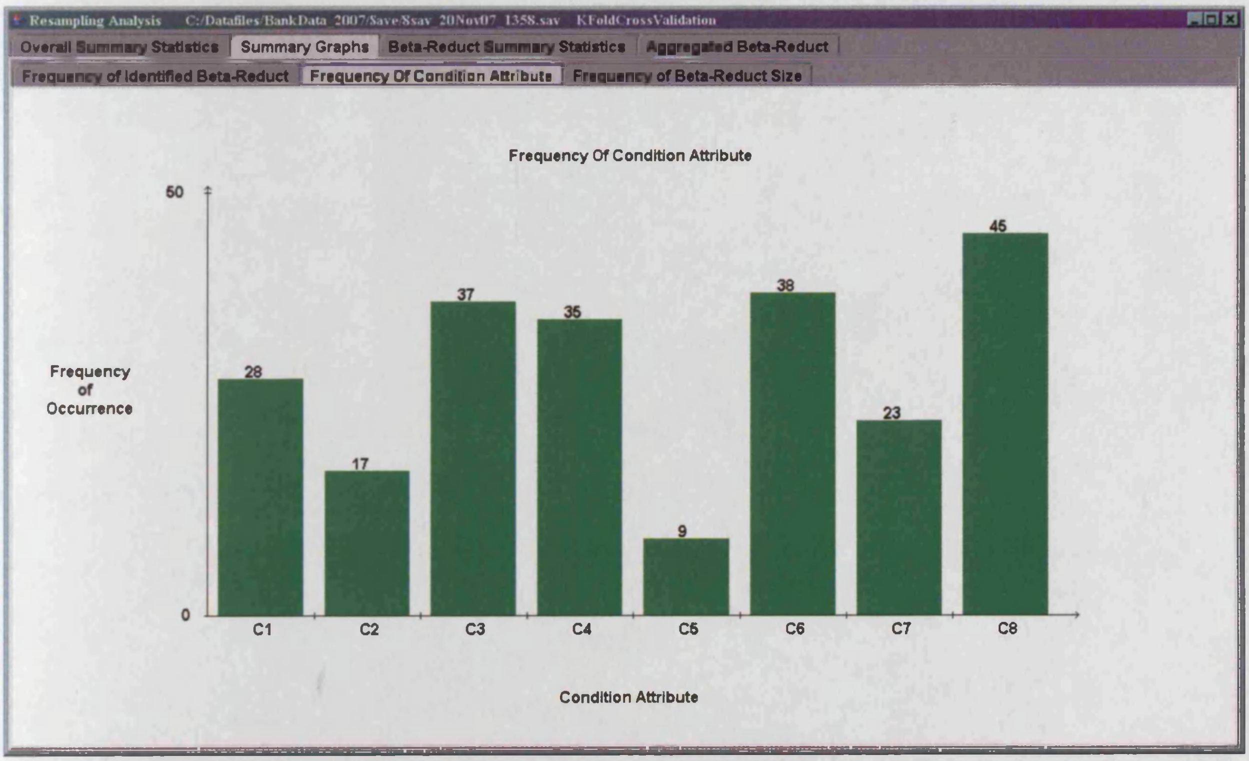


Figure A.2.5: Frequency of Occurrence, of the Condition Attributes Associated with the Selected β-reducts, with Regards to the 50-fold Cross-validation Analysis of the FIBR Training Set

A.3 Stratified *k*-fold Cross-validation Graphs Associated with the Frequency of the Selected β-reducts, with Regards to the FIBR Data Set

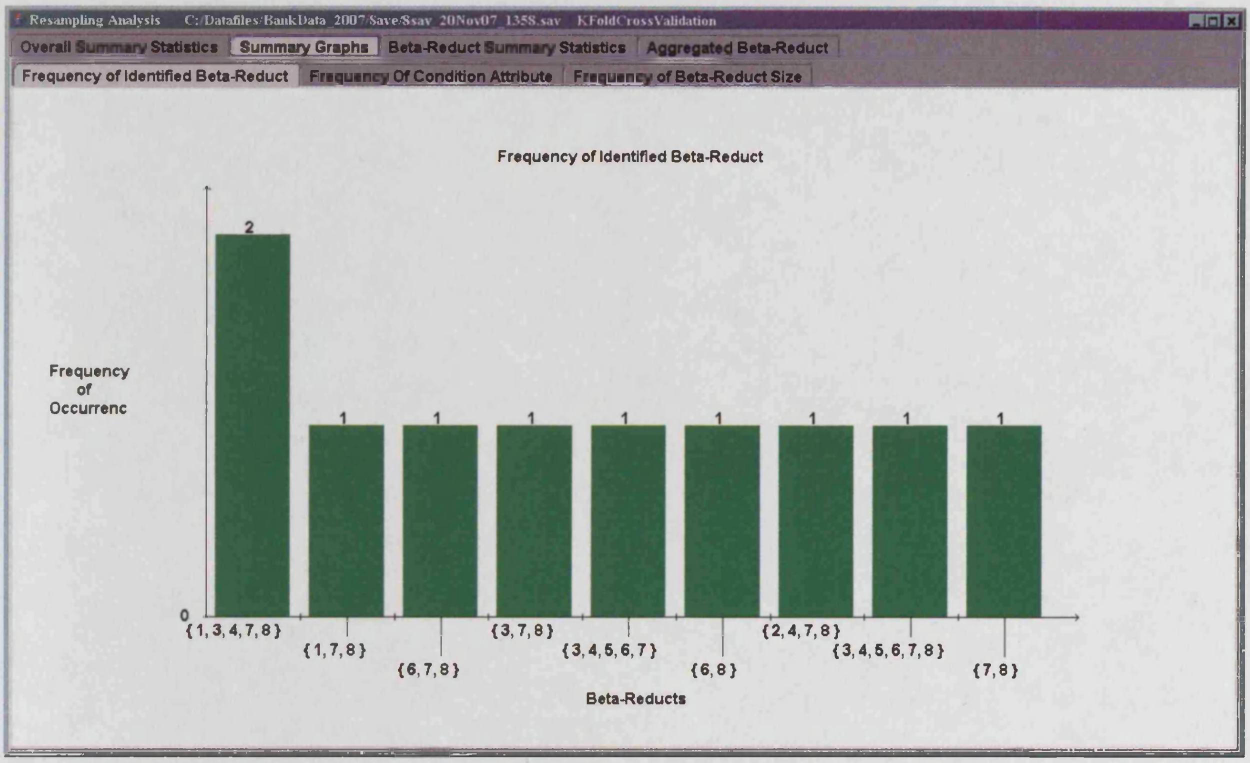

Resampling Analysis C:/Datafiles/BankData_2007/Save/Ssav_20Nov07_1358.sav KFoldCrossValidation
Overall Summary Statistics
Summary Graphs
Beta-Reduct Summary Statistics
Aggregated Beta-Reduct
Frequency of Identified Beta-Reduct
Frequency Of Condition Attribute
Frequency of Beta-Reduct Size
Frequency of Identified Beta-Reduct
Frequency of Occurrenc
0
2
1
1
1
1
1
1
1
1
{1,3,4,7,8}
{1,7,8}
{6,7,8}
{3,7,8}
{3,4,5,6,7}
{6,8}
{2,4,7,8}
{3,4,5,6,7,8}
{7,8}
Beta-Reducts

Figure A.3.1: Frequency of Occurrence of the Selected β-reducts, Associated with the Stratified 10-fold Cross-validation Analysis of the FIBR Training Set

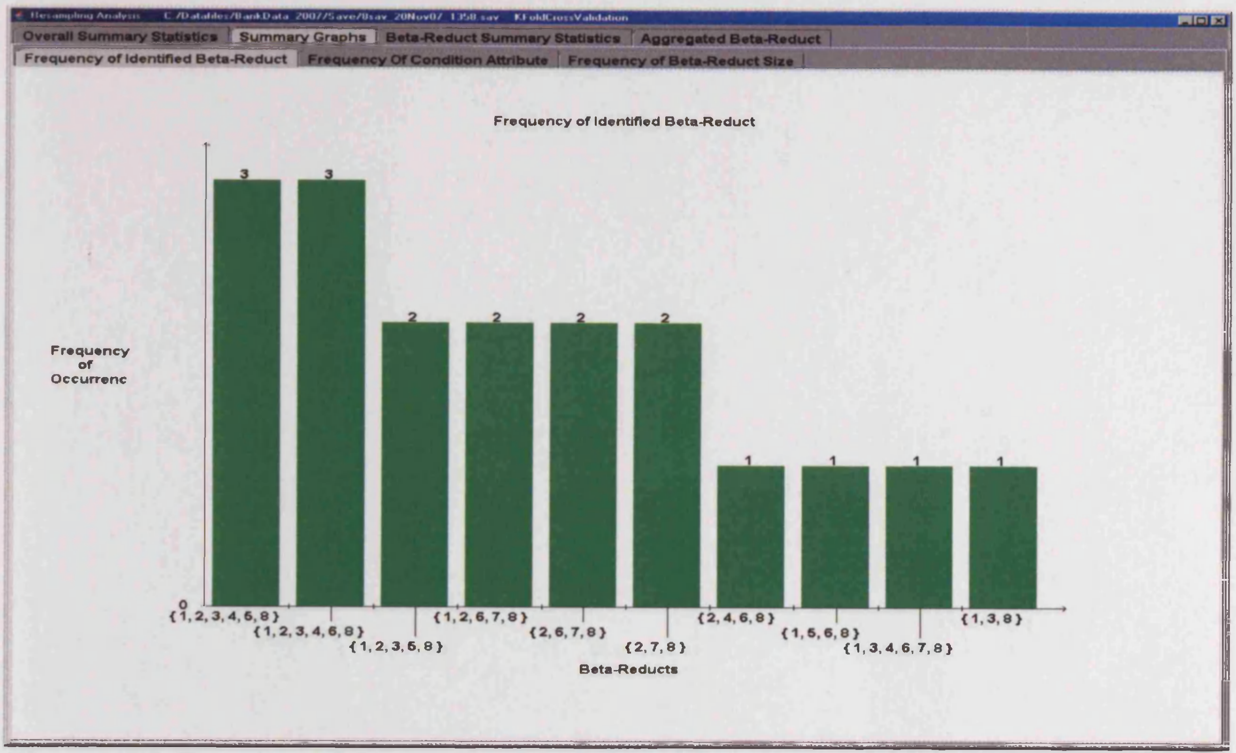


Figure A.3.2: Frequency of Occurrence of the Selected β-reducts, Associated with the Stratified 20-fold Cross-validation Analysis of the FIBR Training Set

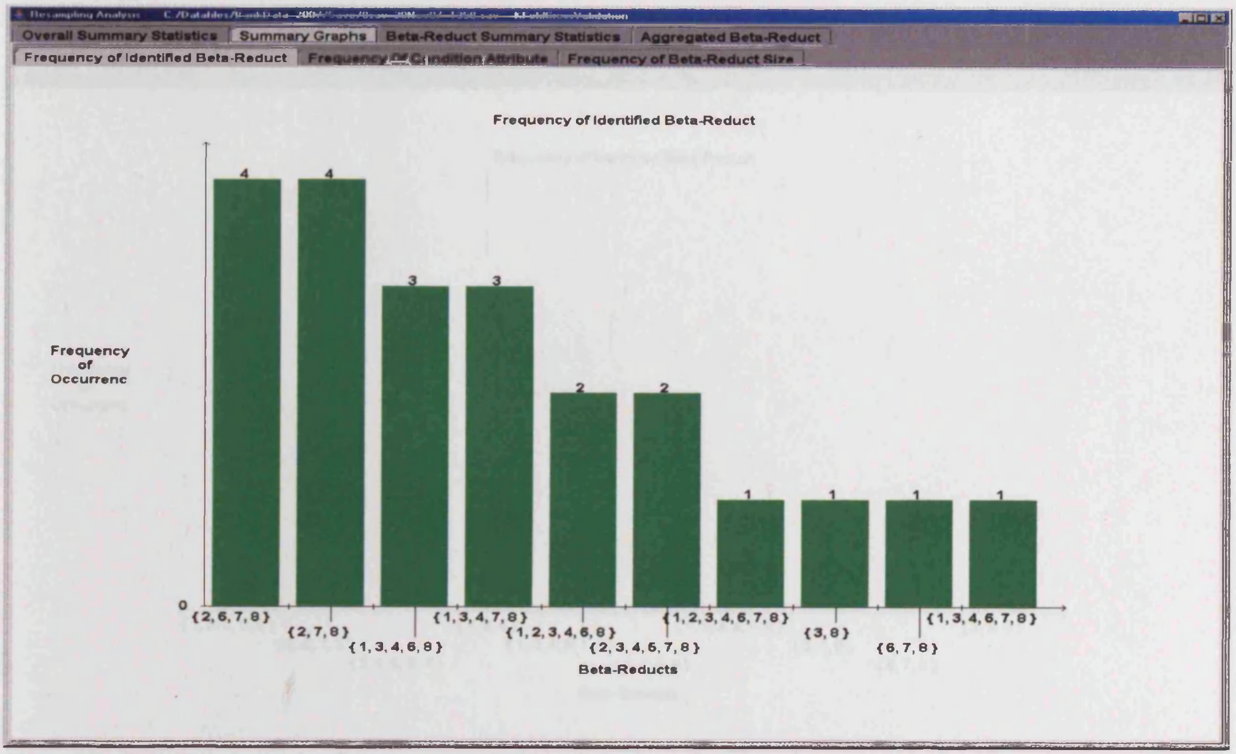


Figure A.3.3: Frequency of Occurrence of the Selected β-reducts, Associated with the Stratified 30-fold Cross-validation Analysis of the FIBR Training Set

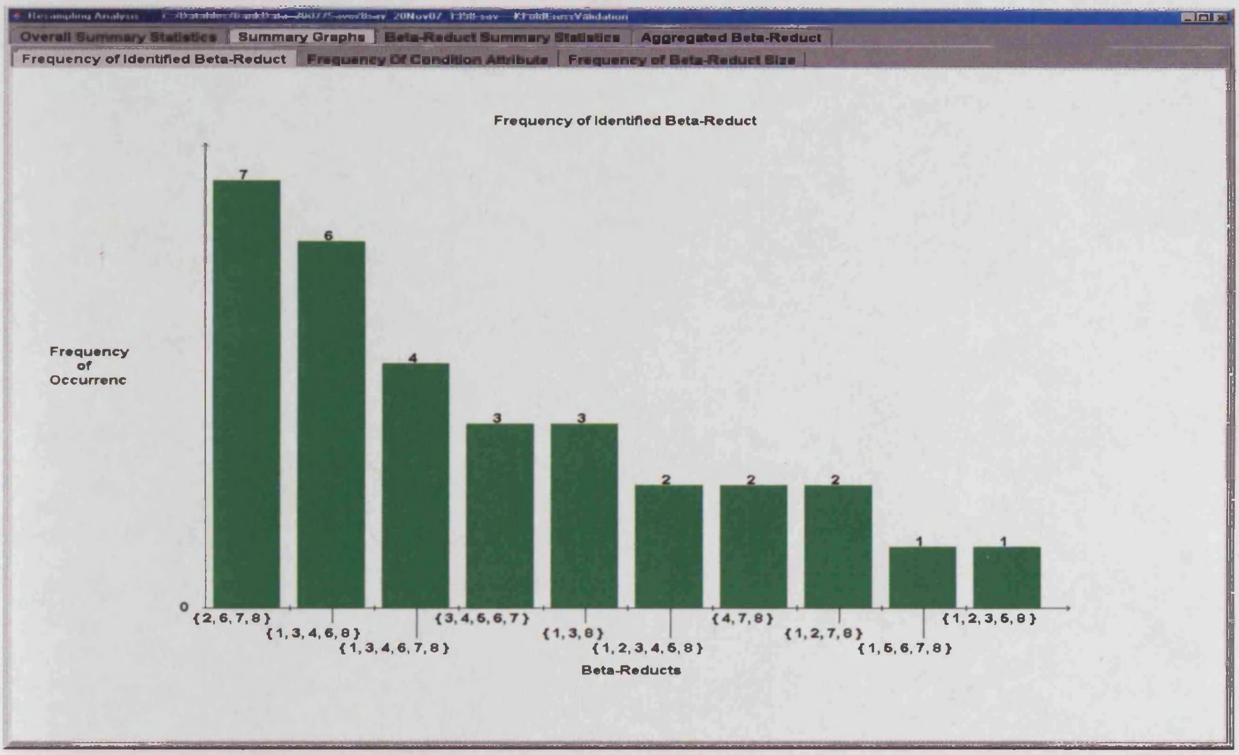


Figure A.3.4: Frequency of Occurrence of the Selected β-reducts, Associated with the Stratified 40-fold Cross-validation Analysis of the FIBR Training Set

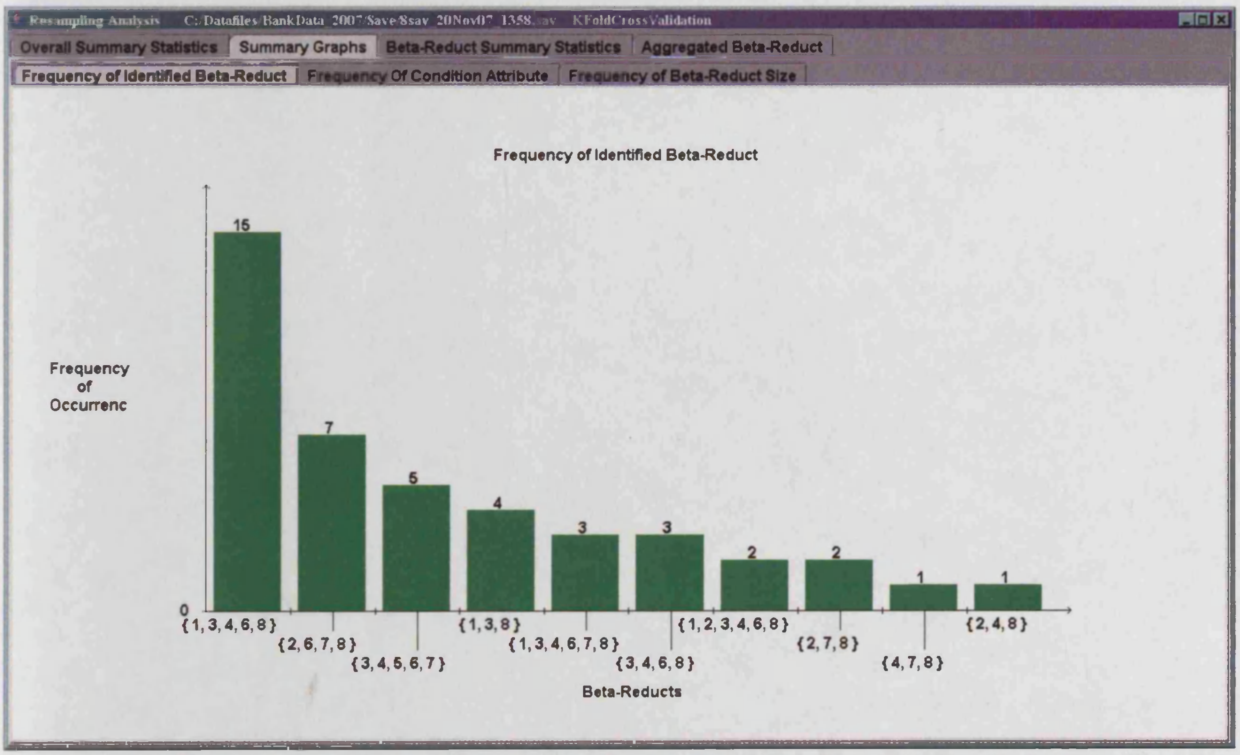


Figure A.3.5: Frequency of Occurrence of the Selected β-reducts, Associated with the Stratified 50-fold Cross-validation Analysis of the FIBR Training Set

A.4 Stratified *k*-fold Cross-validation Graphs Associated with the Frequency of Occurrence, of Condition Attributes, with Regards to the FIBR Data Set

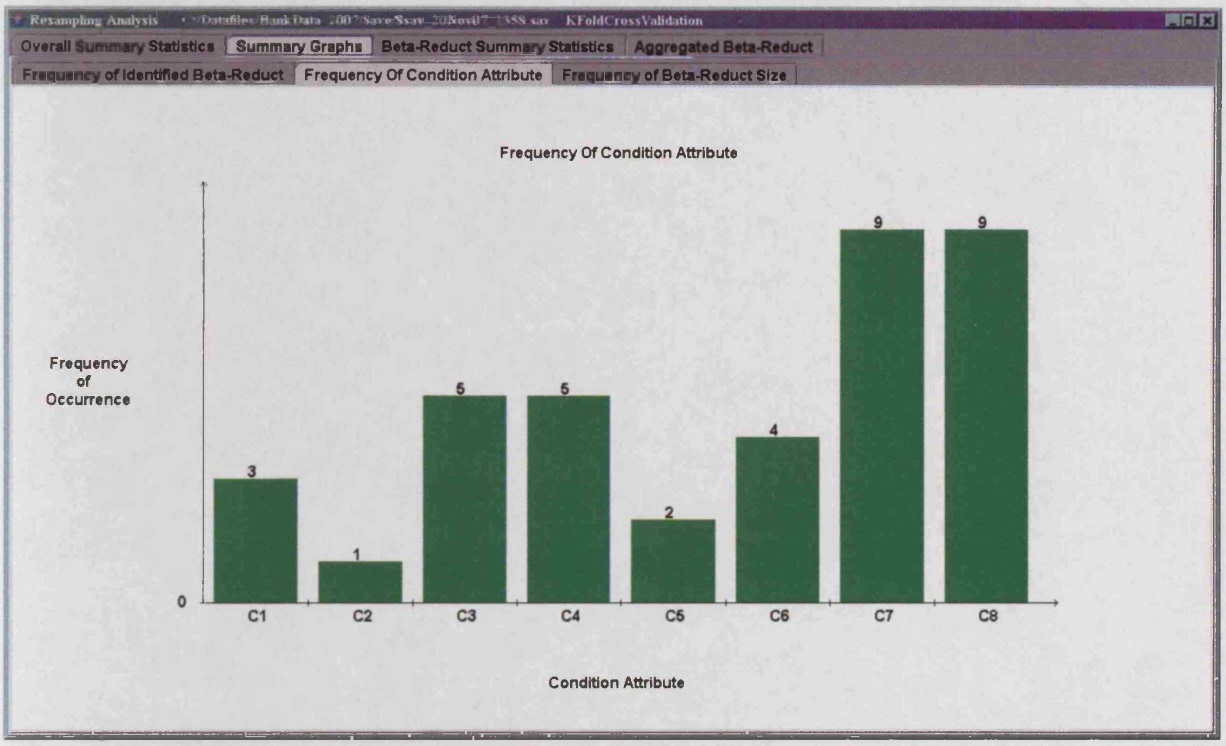

Resampling Analysis C:/Datafiles/BankData 2007/Save/Ssav_20Nov07_1358.sav KFoldCrossValidation
Overall Summary Statistics
Summary Graphs
Beta-Reduct Summary Statistics
Aggregated Beta-Reduct
Frequency of Identified Beta-Reduct
Frequency Of Condition Attribute
Frequency of Beta-Reduct Size
Frequency Of Condition Attribute
Frequency of Occurrence
0
3
1
5
5
2
4
9
9
C1
C2
C3
C4
C5
C6
C7
C8
Condition Attribute

Figure A.4.1: Frequency of Occurrence, of the Condition Attributes Associated with the selected β-reducts, with Regards to the 10-fold Cross-validation Analysis of the FIBR Training Set

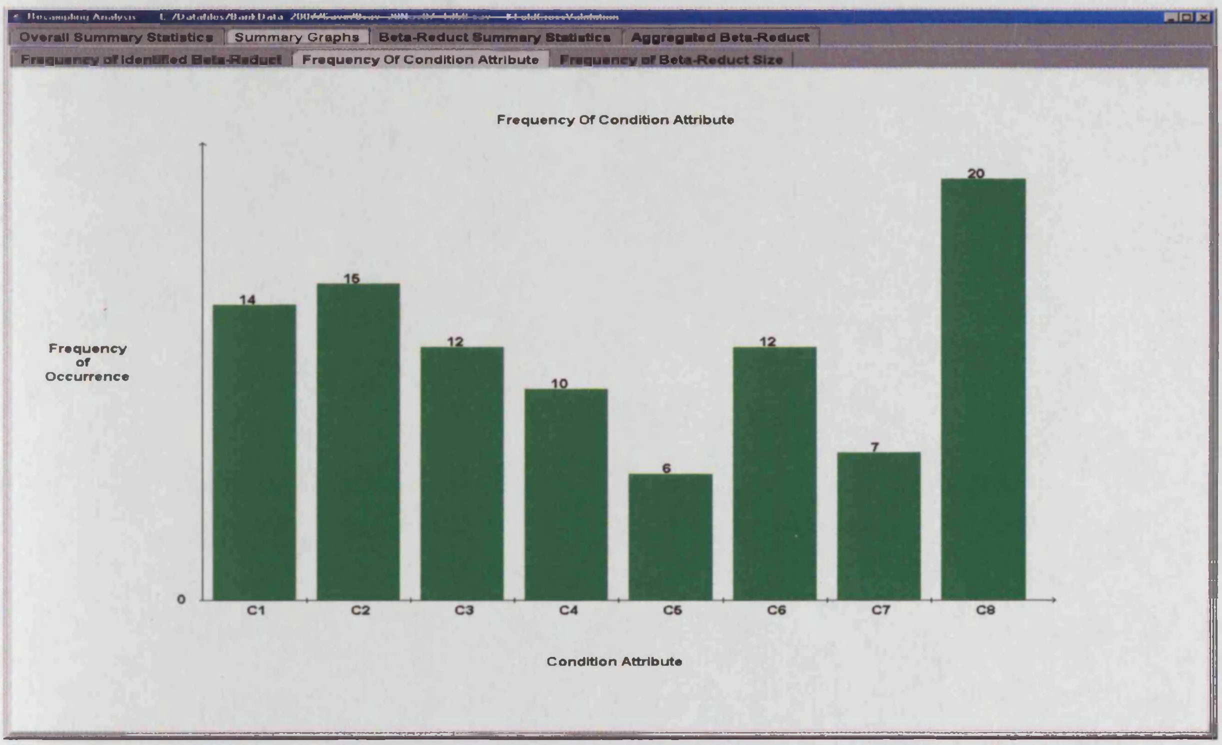

Resampling Analysis
Overall Summary Statistics
Summary Graphs
Beta-Reduct Summary Statistics
Aggregated Beta-Reduct
Frequency of Identified Beta-Reduct
Frequency Of Condition Attribute
Frequency of Beta-Reduct Size
Frequency Of Condition Attribute
Frequency of Occurrence
0
14
15
12
10
6
12
7
20
C1
C2
C3
C4
C5
C6
C7
C8
Condition Attribute

Figure A.4.2: Frequency of Occurrence, of the Condition Attributes Associated with the Selected β-reducts, with Regards to the 20-fold Cross-validation Analysis of the FIBR Training Set

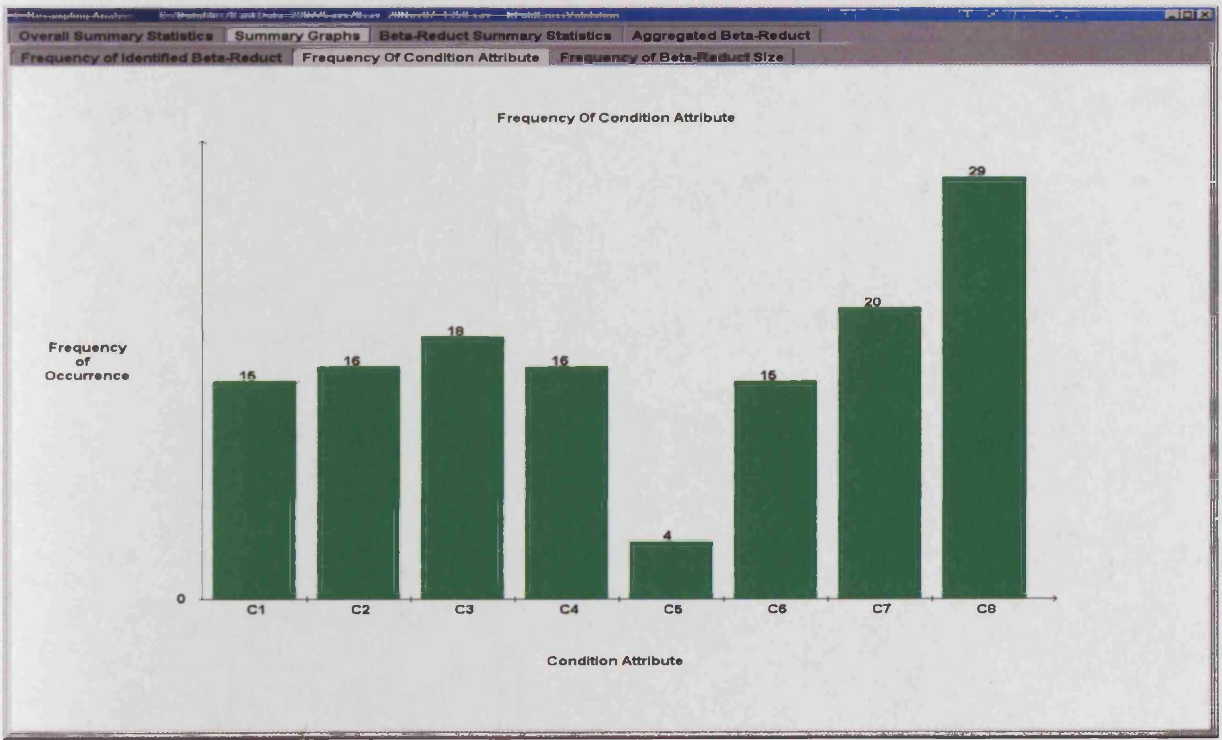

Resampling Analysis
Overall Summary Statistics
Summary Graphs
Beta-Reduct Summary Statistics
Aggregated Beta-Reduct
Frequency of Identified Beta-Reduct
Frequency Of Condition Attribute
Frequency of Beta-Reduct Size
Frequency Of Condition Attribute
Frequency of Occurrence
0
15
16
18
16
4
15
20
29
C1
C2
C3
C4
C5
C6
C7
C8
Condition Attribute

Figure A.4.3: Frequency of Occurrence, of the Condition Attributes Associated with the Selected β-reducts, with Regards to the 30-fold Cross-validation Analysis of the FIBR Training Set

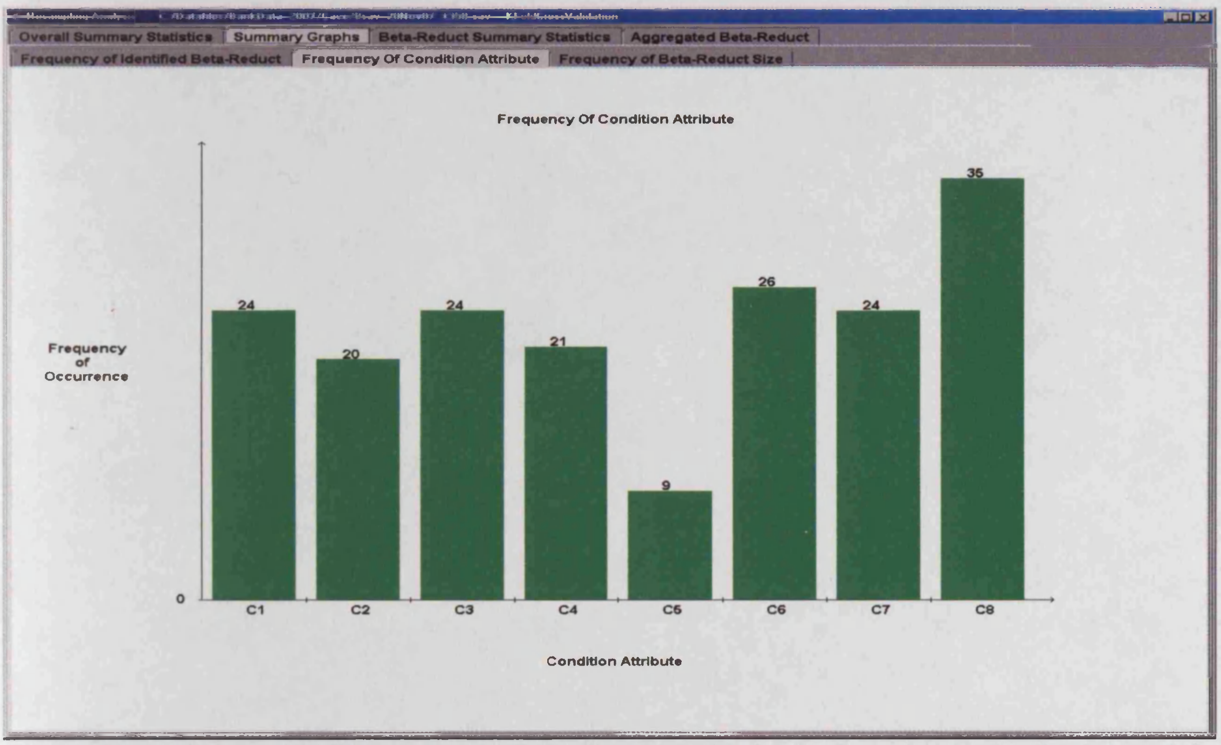


Figure A.4.4: Frequency of Occurrence, of the Condition Attributes Associated with the Selected β-reducts, with Regards to the 40-fold Cross-validation Analysis of the FIBR Training Set

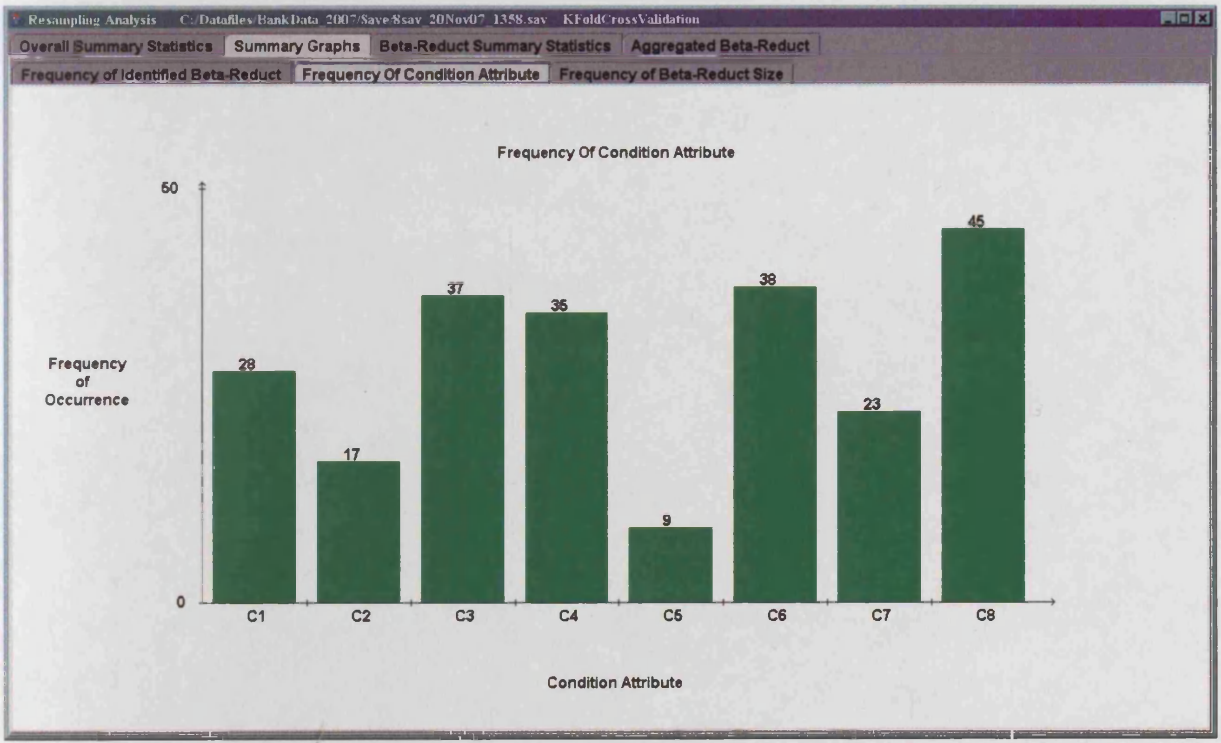


Figure A.4.5: Frequency of Occurrence, of the Condition Attributes Associated with the Selected β-reducts, with Regards to the 50-fold Cross-validation Analysis of the FIBR Training Set

A.5 Predictive Results for Banks Predicted by the Nearest Rule Method

This section presents results on the banks within the FIBR validation set, predicted by the nearest rule method, in the case of leave-one-out (Tables A5.1.1 and A5.1.2) and bootstrapping (Tables A5.2.1 and A5.2.2). These results are associated with the work presented in Chapter 8 section 8.4, and consider the top ten most frequently selected β-reducts with regards to the respective leave-one-out and bootstrapping analyses.

A.5.1 Leave-one-out Aggregation Results Based on the Banks within the FIBR Validation Set Only Predictable by Nearest Rules

β-reduct	Number of Aggregated Rules	Objects Predicted by Matching Rule	Objects Predicted Correctly	Objects Predicted Incorrectly	Predictive Accuracy (%)
$\{c_1, c_3, c_4, c_6, c_8\}$	104	15	8	7	53.33%
$\{c_2, c_6, c_7, c_8\}$	16	11	0	11	0.00%
$\{c_3, c_4, c_5, c_6, c_7\}$	66	13	8	5	61.53%
$\{c_2, c_4, c_8\}$	13	7	4	3	57.14%
$\{c_1, c_2, c_3, c_4, c_5, c_8\}$	101	9	5	4	55.55%
$\{c_1, c_2, c_3, c_4, c_7, c_8\}$	72	10	7	3	70.00%
$\{c_2, c_7, c_8\}$	14	7	0	7	0.00%
$\{c_1, c_3, c_4, c_6, c_7, c_8\}$	76	17	9	8	52.94%
$\{c_1, c_3, c_4, c_7, c_8\}$	65	20	7	13	35.00%
$\{c_2, c_3, c_4, c_5, c_7, c_8\}$	88	25	18	7	72.00%

Table A.5.1.1: Aggregated β-reduct Results Associated with the Leave-one-out Analysis of the FIBR Training Set, Applied to the FIBR Validation Set

	Individual Decision Class Predictive Accuracies				
β-reduct	0 (A)	1 (B)	2 (C)	3 (D)	4 (E)
$\{c_1, c_3, c_4, c_6, c_8\}$	0.00%	50.00%	50.00%	100.00%	0.00%
$\{c_2, c_6, c_7, c_8\}$	0.00%	0.00%	0.00%	0.00%	0.00%
$\{c_3, c_4, c_5, c_6, c_7\}$	0.00%	71.42%	60.00%	0.00%	0.00%
$\{c_2, c_4, c_8\}$	0.00%	0.00%	100.00%	0.00%	0.00%
$\{c_1, c_2, c_3, c_4, c_5, c_8\}$	0.00%	83.33%	0.00%	0.00%	0.00%
$\{c_1, c_2, c_3, c_4, c_7, c_8\}$	0.00%	75.00%	80.00%	0.00%	0.00%
$\{c_2, c_7, c_8\}$	0.00%	0.00%	0.00%	0.00%	0.00%
$\{c_1, c_3, c_4, c_6, c_7, c_8\}$	0.00%	45.45%	60.00%	100.00%	0.00%
$\{c_1, c_3, c_4, c_7, c_8\}$	0.00%	25.00%	42.85%	100.00%	0.00%
$\{c_2, c_3, c_4, c_5, c_7, c_8\}$	0.00%	88.23%	42.85%	0.00%	0.00%

Table A.5.1.2: Breakdown of the Validation Set Predictive Accuracies Across the Five Decision Classes, for the Ten Aggregated β-reducts Associated with the Leave-one-out Analysis of the FIBR Validation Set

A.5.2 Bootstrapping Aggregation Results Based on the Banks within the FIBR Validation Set Only Predictable by Nearest Rules

β-reduct	Number of Aggregated Rules	Objects Predicted by Matching Rule	Objects Predicted Correctly	Objects Predicted Incorrectly	Predictive Accuracy (%)
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	565	52	11	41	21.15%
$\{c_7, c_8\}$	19	15	6	9	40.00%
$\{c_1, c_2, c_7, c_8\}$	102	13	11	2	78.57%
$\{c_3, c_4, c_8\}$	86	11	9	2	81.81%
$\{c_2, c_7, c_8\}$	37	9	0	9	0.00%
$\{c_1, c_2, c_3, c_4, c_5, c_7, c_8\}$	366	31	16	15	51.61%
$\{c_4, c_6, c_8\}$	46	16	7	9	43.75%
$\{c_3, c_4, c_7, c_8\}$	133	14	7	7	50.00%
$\{c_1, c_3, c_4, c_8\}$	169	6	4	2	66.66%
$\{c_1, c_2, c_3, c_4, c_5, c_8\}$	329	10	6	4	60.00%

Table A.5.2.1: Aggregated β-reduct Results Associated with the Bootstrap Analysis of the FIBR Training Set, Applied to the FIBR Validation Set

	Individual Decision Class Predictive Accuracies				
β-reduct	0 (A)	1 (B)	2 (C)	3 (D)	4 (E)
$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8\}$	0.00%	13.88%	40.00%	50.00%	0.00%
$\{c_7, c_8\}$	0.00%	100.00%	18.18%	0.00%	0.00%
$\{c_1, c_2, c_7, c_8\}$	0.00%	100.00%	75.00%	0.00%	0.00%
$\{c_3, c_4, c_8\}$	0.00%	50.00%	83.33%	100.00%	0.00%
$\{c_2, c_7, c_8\}$	0.00%	0.00%	0.00%	0.00%	0.00%
$\{c_1, c_2, c_3, c_4, c_5, c_7, c_8\}$	0.00%	55.55%	62.50%	25.00%	0.00%
$\{c_4, c_6, c_8\}$	0.00%	0.00%	100.00%	25.00%	0.00%
$\{c_3, c_4, c_7, c_8\}$	0.00%	100.00%	44.44%	0.00%	0.00%
$\{c_1, c_3, c_4, c_8\}$	0.00%	50.00%	50.00%	100.00%	0.00%
$\{c_1, c_2, c_3, c_4, c_5, c_8\}$	0.00%	100.00%	33.33%	0.00%	0.00%

Table A.5.2.2: Breakdown of the Validation Set Predictive Accuracies Across the Five Decision Classes, for the Ten Aggregated β-reducts Associated with the Bootstrap Analysis of the FIBR Validation Set

A.6 Attributes Selected for VPRS Re-sampling Analyses Associated with the Benchmark Data Sets

The following tables display the condition attributes selected from the original benchmark data sets utilised in section 8.5, and passed into the VPRS re-sampling analyses associated with that section. The tables give the actual attribute names and the shorter index condition attribute names (e.g. c_1, c_2 etc.) used in section 8.5. Full listings and details of all attributes associated with the utilised benchmark data sets can be found at the the UCI Machine Learning Repository (see http://archive.ics.uci.edu/ml/index.html).

Note that attributes associated with the the original SPECT data set had no meaningful attribute names, and are recorded as F1 to F22.

A.6.1 Breast-cancer Data Set Selected Attributes

Condition Attribute	Indexed as
Clump Thickness	c_1
Uniformity of Cell Size	c_2
Uniformity of Cell Shape	c_3
Single Epithelial Cell Size	c_4
Bare Nuclei	c_5
Bland Chromatin	c_6
Normal Nucleoli	c_7

Table A.6.1.1: Attributes Selected with regards to the Breast-cancer Data Set

A.6.2 Iris Data Set Selected Attributes

Condition Attribute	Indexed as
Sepal length (cm)	c_1
Sepal width (cm)	c_2
Petal length (cm)	c_3
Petal width (cm)	c_4

Table A.6.2.1: Attributes Selected with regards to the Iris Data Set

A.6.3 SPECT Data Set Selected Attributes

Condition Attribute	Indexed as
F4	c_1
F13	c_2
F14	c_3
F16	c_4
F17	c_5
F22	c_6

Table A.6.3.1: Attributes Selected with regards to the SPECT Data Set

A.6.4 Wine Data Set Selected Attributes

Condition Attribute	Indexed as
Alcohol	c_1
Malic acid	c_2
Ash	c_3
Alcalinity of ash	c_4
Magnesium	c_5
Total phenols	c_6
Flavanoids	c_7
Nonflavanoid phenols	c_8

Table A.6.4.1: Attributes Selected with regards to the Wine Data Set

A.6.5 Zoo Data Set Attributes Selected Attributes

Condition Attribute	Indexed as
feathers	c_1
eggs	c_2
milk	c_3
toothed	c_4
backbone	c_5
breathes	c_6

Table A.6.5.1: Attributes Selected with regards to the Zoo Data Set

Bibliography

Aburdene, P. (2007). *Megatrends 2010: The Rise of Conscious Capitalism.* Hampton Roads Publishing Company.

Adibi, J., Ghoreishi, A., Fahimi, M. and Maleki, Z. (1993). Fuzzy Logic-Information Theory Hybrid Model for Medical Diagnostic Expert Systems. In: *Proceedings of the Twelfth Southern Biomedical Engineering Conference*, 211-213.

Allen, B., P. (1994). Case-Based Reasoning: Business Applications. *Communications of the ACM*, 37(3), 40-42.

Allenby, R., B., J., T. (1997). *Numbers and Proofs*. Arnold, London.

Almuallim, H. and Dietterich, T., G. (1991). Learning With Many Irrelevant Features. In: *Proceedings of the Ninth National Conference on Artificial Intelligence*, CA, AAAI Press, 547-552.

Altman, E. (1968). Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy. *The Journal of Finance*, 23(4), 589-609.

Altman, E., Avery, R., B., Eisenbeis, R., A. and Sinkey, J., F., Jr. (1981). Application of Classification Techniques in Business, Banking and Finance. *Journal of Money, Credit and Banking*, 15(4), 532-535.

Ambroise, C. and McLachlan, G., J. (2002). Selection Bias in Gene Extraction on the Basis of Microarray Gene-Expression Data. *Proceedings of the National Academy of Sciences*, USA, 99, 6562-6566.

An, A., Cercone, N. and Huang, X. (2001). A Case Study for Learning from Imbalanced Data Sets. In: Stroulia, E. and Matwin, S. (eds.), *Proceedings of the 14th Biennial Conference of the Canadian Society on Computational Studies of intelligence: Advances in Artificial intelligence*. Springer-Verlag, London, 1-15.

An, A., Shan, N., Chan, C., Cercone, N. and Ziarko, W. (1996). Discovering Rules for Water Demand Prediction: An Enhanced Rough-Set Approach. *Engineering Applications and Artificial Intelligence*, 9(6), 645-653.

Andrews D., W., K. (2004). The Block-Block Bootstrap: Improved Asymptotic Refinements.

Econometrica, 72(3), 673-700.

Andrews, D., K. and Buchinsky, M. (2000). A Three-Step Method for Choosing the Number of Bootstrap Repetitions. *Econometrica*, 68, 1, 23-51.

Arabie, P., Hubert, L., J. and De Soete, G. (1996). *Clustering and Classification.* World Scientific.

Bachmann, R., Malsch, T. and Ziegler, S. (1993). Success and Failure of Expert Systems in Different Fields of Industrial Application. In: Ohlbach, H. J. (ed.). *Proceedings of the 16th German Conference on Artificial intelligence: Advances in Artificial intelligence*, 671, 77-86.

Bankscope. (2007). Bureau van Dijk. https://bankscope.bvdep.com.

Basel. (2007). http://www.bis.org/bcbs/index.htm.

Baumann, K. (2003). Cross-validation as the Objective Function for Variable-Selection Techniques. *Trends in Analytical Chemistry*, 22, 395-406.

Bauer, E. and Kohavi, R. (1999). An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. *Machine Learning*, 36, 105-139.

Bazan, J., Skowron, A. and Synak, P. (1994). Dynamic Reducts as a Tool for Extracting Laws from Decision Tables. In: Ras, Z., W. and Zemankova, M. (eds.), *International Symposium on Methodologies for Intelligent Systems*, Springer-Verlag, 869, 346-355.

Bazan, J. (1998). A Comparison of Dynamic and Non-Dynamic Rough Set Methods for Extracting Laws from Decision Tables. In: Polkowski, L. and Skowron, A. (eds.), *Rough Sets in Knowledge Discovery*, Physica-Verlag, Heidelberg, 321-365.

Belkaoui, A. (1980). Industrial Bond Ratings: A New Look. *Financial Management*, 9(3), 44-51.

Bell, M., Z. (1985). Why Expert Systems Fail. *The Journal of the Operational Research Society*, 36(7), 613-619.

Beynon, M. (2001). Reducts Within the Variable Precision Rough Sets Model: A Further Investigation. *European Journal of Operational Research*, 134, 592-605.

Beynon, M., J. and Peel, M., J. (2001). Variable Precision Rough Set Theory and Data Discretisation:an Application to Corporate Failure Prediction. *Omega*, 29(6), 561-576.

Beynon, M., J. (2003). The Introduction and Utilization of (*l*, *u*)-graphs in the Extended Variable Precision Rough Sets Model. *International Journal of Intelligence Systems*, 18, 1035-1055.

Beynon, M., J. and Buchanan, K., L. (2003). An Illustration of Variable Precision Rough Set Theory: The Gender Classification of the European Barn Swallow (*Hirundo rustica*) *Bulletin of Mathematical Biology*, 65, 835-858.

Beynon, M., J. (2004). The Elucidation of an Iterative Procedure to *β*-Reduct Selection in the Variable Precision Rough Sets Model. In: Tsumoto, S., Slowinski, R., Komorowski, J. and Grzymala-Busse, J., W. (eds.), Rough Sets and Current Trends in Computing. *Fourth International Conference on Rough Sets and Current Trends in Computing*, Uppsala, Sweden, Springer-Verlag, Berlin Heidelberg, 412-417.

Beynon, M., J., Clatworthy, M., A. and Jones, M., J. (2004). A Prediction of Profitability Using Accounting Narratives: A Variable Precision Rough Set Approach. *International Journal of Intelligent Systems in Accounting, Finance & Management*, 12, 227-242.

Bishop, E. (1967). *Introduction to Foundations of Constructive Analysis.* New York: Academic Press.

Bloomberg (2008). http://www.bloomberg.com.

Borra, S. and Di Ciaccio, A. (2002). Improving Nonparametric Regression Methods by Bagging and Boosting. *Computational Statistics & Data Analysis,* 38, 407-420.

Boullé, M. (2004). Khiops: A Statistical Discretization Method of Continuous Attributes. *Machine Learning*, 55(1), 53-69.

Brachman, R. and Anand, T. (1996). The Process of Knowledge Discovery in Databases: A Human-Centered Approach. In: Fayyad, U., Piatetsky-Shapiro, G., Smyth, P. and Uthurusamy, R (eds.), *Advances in Knowledge Discovery and Data Mining*, AAAI Press, California, USA, 37-58.

Brachman, R., Khabaza, T., Kloesgen, W., Piatesky-Shapiro, G. and Simoudis, E. (1996). Mining Business Databases. *Communication of ACM*, 39(11), 42-48.

Breiman, L., Friedman, J., H., Olshen, R., A. and Stone, C., J. (1984). *Classification and Regression Trees*. Wadsworth, Belmont, California.

Breiman, L. (1996a). Bagging Predictors. *Machine Learning*, 24, 123-140.

Breiman, L. (1996b). Out-of-bag Estimation. *Technical Report,* Statistics Department, University of California.

Breiman, L. (2001). Using Iterated Bagging to Debias Regressions. *Machine Learning,* 45, 261-277.

Brooks, S., P., Giudici P. and Roberts G., O. (2003). Efficient Construction of Reversible Jump MCMC Proposal Distributions. *Journal of the Royal Statistical Society, Series B,* 1, 1-37.

Brownstone, D. and Valletta, R. (2001). The Bootstrap and Multiple Imputations: Harnessing Increased Computing Power for Improved Statistical Tests. *The Journal of Economic Perspectives,* 15(4), 129-141.

Bryll, R., Gutierrez-Osuna, R. and Quek, F. (2003). Attribute Bagging: Improving Accuracy of Classifier Ensembles by Using Random Feature Subsets. *Pattern Recognition,* 36, 1291-1302.

Bureau van Dijk. (2008). http://www.bvdep.com/en/companyInformationHome.html.

Business Intelligence (BI) Channel. (2008). DM Review.

http://www.dmreview.com/channels/business_intelligence.html.

Cantor, G. (1883). *Grundlagen Einer Allgemienen Mannigfaltigkeitslehre.* Leipzig, Germany.

Cantor, R. and Packer, F. (1994). The Credit Rating Industry. *Federal Reserve Bank of New York - Quarterly Review,* Summer-Fall.

Cantor, R. and Packer, F. (1995). The Credit Rating Industry. *Journal of Fixed Income,* 5, 10-34.

Carr, E., H. (2001). *What is History?* Palgrave Macmillan.

Chan, C., Batur, C. and Srinivasan, A. (1991). Determination of Quantization Intervals in Rule Based Model for Dynamic Systems. In, *Proceedings of the IEEE Conference on Systems, Man, and Cybernetics.* Charlottesvile, Virginia, 1719-1723.

Chawla, N., V., Bowyer, K., W., Hall, L., O. and Kegelmeyer, W., P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Jounal of Artificial Intelligence Research,* 16, 321-357.

Chmielewski, M., R. and Grzymala-Busse, J., W. (1995). Global Discretization of Continuous Attributes as Preprocessing for Machine Learning. In: Lin, T., Y. and Wilderberger, A., M. (eds.), *Soft Computing: Rough Sets, Fuzzy Logic, Neural Networks, Uncertainty Management, Knowledge Discovery,* San Jose, CA: Simulation Councils, 294-297.

Chmielewski, M., R. and Grzymala-Busse, J., W. (1996). Global Discretization of Continuous Attributes as Preprocessing for Machine Learning. *International Journal of Approximate Reasoning,* 15, 319-331.

Chouchoulas, A. and Shen, Q. (2001). Rough Set-Aided Keyword Reduction for Text Categorisation. *Applied Artificial Intelligence,* 15(9), 645-664.

Chouchoulas, A., Halliwell, J. and Shen, Q. (2002). On Implementation of Rough Set Attribute Reduction. *Proceedings of the 2002 U.K. Workshop on Computational Intelligence,* 18-23.

Cios, K., J., Wedding, D., K. and Liu, N. (1997). CLIP3: Cover Learning Using Integer Programming. *Kybernetes,* 26(4-5), 513-536.

Cios, K., J. and Kurgan, L. (2001). Hybrid Inductive Machine Learning: An Overview of CLIP Algorithms. In: Jain, L,.C. and Kacprzyk, J. (eds). *New Learning Paradigms in Soft Computing,* Physica-Verlag (Springer), 276-321.

Clark, P. and Niblett, T. (1989). The CN2 Induction Algorithm. *Machine Learning,* 3, 261-283.

Cochran, W., G. (1963). *Sampling Techniques, Second Addition.* Wiley, New York.

Cohen, W., W. (1995). Fast Effective Rule Induction. In: Prieditis, A. and Russell, S. (eds.), *Proceedings of the 12th International Conference on Machine Learning,* Tahoe City, California, Morgan Kaufmann, 115-123.

Coppin, B. (2004). *Artificial Intelligence Illuminated.* Jones and Bartlett publishers, Illuminated Series.

Craven, M., W. and Shavlik, J., W. (1997). Using Neural Networks for Data Mining. *Future Generation Computer Systems, Special Double Issue on Data Mining,* 13(1-3), 211-229.

Cureton, E., E. (1951). Symposium: The Need and Means of Cross-validation, 2. Approximate Linear Restraints and Best Predictor Weights. *Educational and Psychological Measurement,* 11, 12-15.

Davison, A., C. and Hinkley, D., V. (1997). *Bootstrap Methods and their Application.* Cambridge University Press, Cambridge, UK.

Daughen, J. and Binzen, P. (1999). *The Wreck of the Penn Central, 2nd edition.* Beard Books, U.S.

Dembczyński, K., Greco, S., Kotłowski, W. and Słowiński, R. (2007). Statistical Model for Rough Set Approach to Multicriteria Classification. *Knowledge Discovery in Databases*, 164-175.

Derviz, A. and Podpiera., J. (2004). Predicting Bank CAMELS and S&P Ratings: The Case of the Czech Republic. *Working Papers Series of the Czech National Bank.*

DeYoung, R., Flannery, M., J., Lang, W., W. and Sorescu, S., M. (2001). The Information Content of Bank Exam Ratings and Subordinated Debt Prices. *Journal of Money, Credit and Banking*, 33(4), 900-925.

Dietterich, T., G. (1997). Machine Learning Research: Four Current Directions. *AI Magazine*, 18(4), 97-136.

Dietterich, T., G. (2000a). An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization. *Machine Learning*, 40, 139-157.

Dietterich, T., G. (2000b). Ensemble Methods in Machine Learning. *Lecture Notes in Computer Science*, 1857, 1-15.

Dixon, P., M., Weiner, J., Mitchell-Olds, T. and Woodley, R. (1987). Bootstrapping the Gini Coefficient of Inequality. *Ecology*, 68(5), 1548-1551.

Dobson, A., J. (2001). *Introduction to Generalized Linear Models, Second Edition.* Chapman and Hall/CRC, London.

Domingos, P. (1999). The Role of Occam's Razor in Knowledge Discovery. *Data Mining and Knowledge Discovery*, 3, 409-425.

Dougherty, J., Kohavi, R. and Sahami, M. (1995). Supervised and Unsupervised Discretization of Continuous Features. In: *Proceedings of the 12th International Conference on Machine Learning*, Morgan Kaufmann, Tahoe City, CA, 194-202.

Doumpos, M. and Pasiouras, F. (2005). Developing Models for Replicating Credit Ratings: A

Multicriteria Approach. *Computational Economics*, 25, 327-341

Dow Jones Newswires. (2008). http://www.djnewswires.com.

Drummond, C. and Holte, R., C. (2000). Exploiting the Cost (In)sensitivity of Decision Tree Splitting Criteria. In: Langley, P. (ed.), *Proceedings of the Seventeenth international Conference on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 239-246.

Duda, Ŕ. and Hart, P. (1973). *Pattern Classification and Scene Analysis*. Wiley, New York U.S.A.

Duda, R., Hart, P. and Stork, D. (2000). *Pattern Classification*. Wiley-Interscience.

Düntsch, I. and Gediga, G. (1997). Statistical Evaluation of Rough Set Dependency Analysis. *International Journal of Human-Computer Studies*, 46, 589-604.

Efron, B. (1979). Bootstrap Methods: Another Look at the Jackknife. *Annals of Statistics*, 7, 1-26.

Efron, B. (1982). *The Jackknife, the Bootstrap and Other Re-sampling Plans*. SIAM, Piladelphia U.S.A.

Efron, B. (1983). Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation. *Journal of the American Statistical Association*, 78(382), 316-331.

Efron, B. (2003). Second Thoughts on the Bootstrap. *Statistical Science: Silver Anniversary of the Bootstrap*, 18(2) 135-140.

Efron, B. and Tibshirani, R. (1993). *An Introduction to the Bootstrap*. Chapman and Hall.

Efron, B. and Tibshirani, R. (1997). Improvements on Cross-Validation: the .632+ Bootstrap Method. *Journal of the American Statistical Association*, 92, 548-560.

EIU. (2007). The European Intelligence Unit. http://www.eiu.com.

Estabrooks, A., Jo, T. and Japkowicz, N. (2004). A Multiple Resampling Method for Learning from Imbalanced Data Sets. *Computational Intelligence*, 20(1), 18-36.

Falkenstein, E., Boral, A. and Carty L., V. (2000). *RiskCalc for Private Companies: Modelling Methodology*. Moody's KMV Company.

Fayyad M., U. and Irani B., K. (1992). Technical Note: On the Handling of Continuous-Valued Attributes in Decision Tree Generation. *Machine Learning*, 8(1), 87-102.

Fayyad, M., U., Piatetsky-Shapiro, G. and Smyth, P. (1996). From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, 17(3), 37-54.

FDIC (2008). http://www.fdic.gov/bank/individual/failed/IndyMac.html.

Feldman, R., Jagtiani, J. and Schmidt, J. (2003). The Impact of Supervisory Disclosure: Will Bank Supervisors Be Less Likely to Downgrade Banks? *Federal Reserve Banks of Minneapolis, Kansas City, Minneapolis; Banking and Policy Studies.*

Fisher, L. (1959). Determinants of Risk Premiums on Corporate Bonds. *The Journal of Political Economy*, 67(3), 217-237 .

Fitch (2007). www.fitchratings.com.

Foster, G. (1978). *Financial Statement Analysis*, Prentice-Hall.

Frawley, W., J., Piatetsky-Shapiro, G. and Matheus, C. (1992). Knowledge Discovery In Databases: An Overview. *AI Magazine*, 13(3), 57-70.

Frege, G. (1903). *Gtundgesetzen der Arithmetik*, 2.

Freund, Y. and Schapire, R. (1997). A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1), 119-139.

Friedman, J., H. (1977). A Recursive Partitioning Decision Rule for Nonparametric Classification. *IEEE Transactions on Computers*, 26(4), 404-408.

Friedman, J., H. (1999). Greedy Function Approximation: A Gradient Boosting Method. *Technical Report*, Department of Statistics, Stanford University.

Friedman, J,. H. (2002). Stochastic Gradient Boosting. *Nonlinear Methods and Data Mining*, 38(4), 367-378.

Friedman, T., L. (1996). The News Hour, Interview with Jim Lehrer, PBS Television Broadcast. Feb. 13th.

Fu, W., J., Carroll, R., J. and Wang, S. (2005). Estimating Misclassification Error with Small Samples via Bootstrap Cross-Validation. *Bioinformatics*, 21(9), 1979-1986.

Geman, S., Bienenstock, E. and Dousat, R. (1992). Neural Networks and the Bias/Variance Dilemma. *Neural Computation*, 5, 1-58.

Gilbert, R., Meyer, A. and Vaughan, M. (2000). The Role of a CAMEL Downgrade Model in Bank Surveillance. *Federal Reserve Bank of St. Louis, Working Papers*.

Giudici, P. (2003). *Applied Data Mining: Statistical Methods for Business and Industry*. Wiley, England.

Goldberg P., W. (2001). When Can Two Unsupervised Learners Achieve PAC Separation? In: Helmbold, D. and Williamson, B. (eds.), *Proceedings of Computational Learning Theory: 14th Annual Conference on Computational Learning Theory, and 5th European Conference on Computational Learning Theory*, Amsterdam, The Netherlands, 303-319.

Goldstein, M. and Wooff, D. (2007). *Bayes Linear Statistics: Theory and Methods*. Wiley.

Gonzalez, F., Haas, F., Johannes, R., Persson, M., Toledo, L., Violi, R., Wieland, M. and Zins, C. (2004). Market Dynamics Associated with Credit Ratings, A Literature Review. *European Central Bank, Occasional Paper Series*, 16.

Greco, S., Matarazzo, B. and Słowiński, R. (1999). Handling Missing Values in Rough Set Analysis of Multi-Attribute and Multi-Criteria Decision Problems. In: Zhong, N., Skowron, A. and Ohsuga, S. (eds.), *Proceedings of the 7th international Workshop on New Directions in Rough Sets, Data Mining, and Granular-Soft Computing*, Springer-Verlag, London, 1711, 146-157.

Greco, S., Matarazzo, B. and Słowiński, R. (2001). Rough Sets Theory for Multicriteria Decision Analysis. *European Journal of Operational Research*, 129, 1-47.

Greco, S., Matarazzo, B. and Słowiński, R. (2005). Generalizing Rough Set Theory Through Dominance-Based Rough Set Approach. Rough Sets, Fuzzy Sets, *Data Mining, and Granular Computing*, 1-11.

Green, P., J., Hjort, N. and Richardson, S. (2003). *Highly Structured Stochastic Systems*. Oxford University Press, Oxford.

Griffiths, B. and Beynon, M., J. (2005). Expositing Stages of VPRS Analysis in an Expert System: Application with Bank Credit Ratings, *Expert Systems with Applications*, 29, 879-888.

Griffiths, B. and Beynon, M., J. (2007). Re-sampling Based Data Mining Using Rough Set Theory. In: Zhu, X. and Davidson, I. (eds.), *Knowledge Discovery and Data Mining: Challenges and Realities with Real World Data*. Idea Group, U.S.A, 244-264.

Griffiths, B. and Beynon, M., J. (2008). Information Veins and Re-sampling with Rough Set Theory In: Wang, J. (ed.), *Encyclopaedia of Data Warehousing and Mining, 2nd Edition,* Idea Group, U.S.A. (*In Press*).

Grzymala-Busse, J., Stefanowski, J. and Wilk, S. (2005). Comparison of Two Approaches to Data Mining from Imbalanced Data. *Journal of Intelligent Manufacturing*, 16(6), 565-573.

Guo, H. and Viktor, H., L. (2004). Learning from Imbalanced Data Sets with Boosting and Data Generation: the DataBoost-IM Approach. *SIGKDD Explorations New Letter*, 6(1), 30-39.

Harington, H. (1997). Not Moody–Just Angry. *The Banker*, February, 22-23.

Hastie, T., Tibshirani, R. and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer, New York.

Hall, M. and Smith, L. (1999). Feature Selection for Machine Learning : Comparing a Correlation-Based Filter Approach to the Wrapper. *In Proceedings of the Florida Artificial Intelligence Symposium*.

Hall, M. (2000). Correlation-based Feature Selection for Discrete and Numeric Class Machine Learning. *Proceedings of the Seventeenth International Conference on Machine Learning*, 359-366.

Han, J. and Kamber, M. (2006). *Data Mining: Concepts and Techniques*. Morgan Kaufmann, U.S.A.

Hand, D., Mannila, H. and Smyth, P. (2001). *Principles of Data Mining*. MIT Press.

Harnett, P. and Young, P. (2004). Hedge fund managers: Initial interviews and discussions with.

Harnett, P. and Young, P. (2007). Hedge fund managers: Follow up interviews, software appraisal and feed back.

Haussler, D. (1988). Quantifying Inductive Bias: AI Learning Algorithms and Valiant's Learning Framework. *Artificial Intelligence*, 36, 177-221.

Hills, M. (1966). Allocation Rules and Their Error Rates (with Discussion). *Journal of the Royal Statistic Society, Series B*, 28, 1-31.

Ho, K., M. and Scott, P., D. (1997). Zeta: A global Method for Discretization of Continuous Variables. In, *KDD97: 3rd International Conference of Knowledge Discovery and Data Mining*. Newport Beach, California, 191-194.

Ho, K., M. and Scott, P., D. (1998). An Efficient Global Discretization Method. *Research and Development in Knowledge Discovery and Data Mining.* Springer Berlin/Heidelberg, 383-384

Holte, R., C. (1993). Very Simple Classification Rules Perform Well on Most Commonly Used Datasets. *Machine Learning*, 11, 63-91.

Hong, S., J. (1997). Use of Contextual Information for Feature Ranking and Discretization. *IEEE Transactions on Knowledge and Data Engineering*, 9(5), 718-730.

Hong, S., J. and Weiss S., M. (2001). Advances in Predictive Models for Data Mining. *Pattern Recognition Letters*, 22, 55-61.

Horrigan, J., O. (1966). The Determination of Long-Term Credit Standing with Financial Ratios. *Journal of Accounting Research*, 4, 44-62.

Horrigan, J., O. (1968). A Short History of Financial Ratio Analysis. *The Accounting Review*, 43(2), 284-294.

Horst, P. (1941). Prediction of Personal Adjustment. In: *New York: Social Science Research Council (Bulletin 48).*

Hothorn, T. and Lausen, B. (2003a). Bagging Tree Classifiers for Laser Scanning Images: A Data and Simulation-Based Strategy. *Artificial Intelligence in Medicine*, 27, 65-79.

Hothorn, T. and Lausen, B. (2003b). Double-Bagging: Combining Classifiers by Bootstrap Aggregation. *Pattern Recognition*, 36, 1303-1309.

Hothorn, T. and Lausen, B. (2005). Bundling Classifiers by Bagging Trees. *Computational Statistics and Data Analysis*, 49, 1068-1078.

Huang Y., Cai, J., Ji, L. and Li, Y. (2004a). Classifying G-protein Coupled Receptors with Bagging Classification Tree. *Computational Biology and Chemistry*, 28, 275-280.

Huang, Z., Chen, H., Hsu, C., Chen, W. and Wu, S. (2004b). Credit Rating Analysis with Support Vector Machines and Neural Networks: A Market Comparative Study. *Decision Support Systems*, 37(4), 543-558.

Huang, Z. and Shen, Q. (2006). Fuzzy Interpolative Reasoning Via Scale and Move Transformation. *IEEE Transactions on Fuzzy Systems*, 14(2), 340-359.

Huang, Z. and Shen, Q. (2008). Fuzzy Interpolative and Extrapolative Reasoning: A Practical Approach. *IEEE Transactions on Fuzzy Systems*, 16(1),13-28.

Hull, J. (2006). *Options, Futures and Other Derivatives*. Pearson Prentice Hall, New Jersey, 6th Edition.

Ilczuk, G. and Wakulicz-Deja, A. (2007). Selection of Important Attributes for Medical Diagnosis Systems. In: Peters, F., J. and Skowron, A. (eds.), *Transactions on Rough Sets*, 7, Springer-Verlag, Berlin Heidelberg, 70-84.

Israel, G., D. (1992). *Sampling The Evidence Of Extension Program Impact*. Program Evaluation and Organizational Development. Institute of Food and Agricultural Sciences, University of Florida.

Israel, G., D. (2007). *Determining Sample Size. Institute of Food and Agricultural Sciences.* University of Florida.

Jaganathan, P., Thangavel, A., K. and Pethalakshmi, A. (2006). Effective Classification with Hybrid Evolutionary Techniques. In: *International Conference on Advanced Computing and Communications*, 335-338.

Jagtiani. J., Kolari, J., Lemiex, C. and Shin, H. (2003). Early Warning Models for Bank Supervision: Simpler Could be Better. *Economic Perspectives*. Federal Reserve Bank of Chicargo.

Japkowicz, N. (2000). Learning from Imbalanced Data Sets: a Comparison of Various Strategies. Technical Report, In: *Papers from the AAAI Workshop on Learning from Imbalanced Data Sets*. WS-00-05, Menlo Park, CA.

Jensen, R. (2004). *Combining Rough and Fuzzy Sets for Feature Selection*. Ph.D. Thesis, School of

Informatics, University of Edinburgh.

Jensen, R. and Shen, Q. (2004). Semantics-Preserving Dimensionality Reduction: Rough and Fuzzy- Rough-Based Approaches. In: *Transactions on Knowledge and Data Engineering*, 16(12), 1457-1471.

Jensen, R. and Shen, Q. (2008). *Computational Intelligence and Feature Selection: Rough and Fuzzy Approaches*. IEEE Press and Wiley & Sons.

Jiang, Q. and Abidi, S., S., R. (2005). A Hybrid of Conceptual Clusters, Rough Sets and Attribute Oriented Induction for Inducing Symbolic Rules. In: *Proceedings of 2005 International Conference on Machine Learning and Cybernetics*, 9, 5573-5578.

Jingbo, Z., Na, Y., Xinzhi, C., Wenliang, C. and Tsou, B., K. (2005). Using Multiple Discriminant Analysis Approach for Linear Text Segmentation. *Natural Language Processing*, 3651, 292-301.

Jin, R., Breitbart, Y. and Muoh, C. (2007). Data Discretization Unification. *Knowledge and Information Systems*, 183-192.

Johnson, R. and Wichern, D. (2007). *Applied Multivariate Statistical Analysis*. Pearson, U.S.

Jönsson, P. and Wohlin, C. (2006). Benchmarking *k*-nearest Neighbour Imputation with Homogeneous Likert Data. *Empirical Software Engineering*. 11(3), 463-489.

Kantardzic, M. (2003). *Data Mining: Concepts, Models, Methods and Algorithms*. Wiley-Interscience, U.S.A.

Kattan, M., W. and Cooper, R., B. (1998). The Predictive Accuracy of Computer-Based Classification Decision Techniques, A review and Research Directions. *OMEGA*, 26, 467-482.

Katzberg, J., D. and Ziarko, W. (1994). Variable Precision Rough Sets with Asymmetric Bounds. In: Ziarko, W. (ed.), *Rough Sets, Fuzzy Sets and Knowledge Discovery*, Springer-Verlag, New York, 167-177.

Katzberg, J., D. and Ziarko, W. (1996). Variable Precision Extension of Rough Sets. *Fundamenta Informaticae*, 27, 155-168.

Katzell, R., A. (1951). Symposium: The Need and Means of Cross-Validation, 3. Cross Validation of Item Analysis. *Educational and Psychological Measurement*, 11, 16-22.

Keen, P., G., W. (1980). Decision Support Systems: A Research Perspective. In: Fick, G. and Sprague, R., H. (eds.), *Decision Support Systems: Issues and Challenges*. Oxford/New York, Pergamon Press.

Kennedy, J. and Eberhart, R. (1995). Particle Swarm Optimization. In: *Proceedings of IEEE International Conference On Neural Networks*. Perth, 1942-1948.

Kennedy, S. (2003). China's Credit Rating Agencies Struggle for Relevance. *The China Business Review*, November-December, 36-40.

Kerber, R. (1992). ChiMerge: Discretization of Numeric Attributes. In *Proceedings of AAAI-92, 10th Conference of the American Association for Artificial Intelligence*, San Jose, US, 123-128.

Kim, Y. and Sohn, S., Y. (2008). Random Effects Model for Credit Rating Transitions. *European Journal of Operational Research*, 127, 2, 561-573.

Kira, K. and Rendell, L., A. (1992a). A Practical Approach to Feature Selection. 9*th International Workshop on Machine Intelligence*, Aberdeen, Scotland, Morgan-Kaufman.

Kira, K. and Rendell, L., A. (1992b). The Feature Selection Problem: Traditional Methods and a New Algorithm. In: *10th National Conference on Artificial Intelligence,* MIT Press (1992) 129-134.

Kohavi, R. and John, G. H. (1997). Wrappers for Feature Subset Selection. *Artificicial Intelligence (Special issue on relevance)*, 97(1-2), 273-324.

Kohavi, R., Sommerfield, D. and Dougherty, J. (1997). Data Mining using MLC++: A Machine Learning Library in C++. *International Journal on Artificial Intelligence Tools*, 6(4), 537-566.

Komorowski, J., Pawlak, Z., Polkowski, L. and Skowron, A. (1999). Rough Sets: A Tutorial. In: Pal, S., K. and Skowron A. (eds.), *Rough Fuzzy Hybridization: A New trend in decision Making*, Springer-Verlag, Singapore, 1-98.

Kononenko, I. (1994). Estimating Attributes: Analysis and Extensions of RELIEF. In P*roceedings of the 1994 European Conference on Machine Learning,* 171-182.

Kosmidou, K., Pasiouras, F., Zopounidis, C. and Doumpos, M. (2006). A Multivariate Analysis of the Financial Characteristics of Foreign and Domestic Banks in the UK. *Omega*, 34(2), 189-195.

Kotsianti, S., B. and Kanellopoulos, D. (2007). Combining Bagging, Boosting and Dagging for Classification Problems. *Knowledge-Based Intelligent Information and Engineering Systems*, Springer-Verlag Berlin Heidelberg, 493-500.

Krainer, J. and Lopez, J. (2003). Forecasting Supervisory Ratings Using Securities Market Information. *Proceedings, Federal Reserve Bank of Chicago*, May, 278-289.

Kubat, M. and Matwin, S. (1997). Addressing the Curse of Imbalanced Data Sets: One-Sided Sampling. In: *Proceedings of the Fourteenth International Conference on Machine Learning*, 179-186.

Kuo, T. and Yajima, Y. (2003). Approximate Reducts of an Information System. In: Wang, G. et al. (eds.), *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing: 9th International Conference, RSFDGrC'03,* Springer-Verlag, Berlin Heidelberg, 137-145.

Kuncheva, L., I., Bezdek, J., C. and Duin R., P., W. (2001). Decision Templates for Multiple Classifier Fusion: An Experimental Comparison. *Pattern Recognition*, 34(2), 299-314.

Kurgan, L., A., Cios, K., J., Tadeusiewicz, R., Ogiela, M. and Goodenday, L., S. (2001). Knowledge Discovery Approach to Automated Cardiac SPECT Diagnosis. *Artificial Intelligence in Medicine*, 23(2), 149-169.

Lachenbruch, P. and Mickey, M. (1968). Estimation of Error Rates in Discriminant Analysis. *Technometrics,* 10, 1-11.

Laita, L., M., González-Páez, G., Roanes-Lozano, E., Maojol, V., Ledesmal, L. and Laita, L. (2001). A Methodology for Constructing Expert Systems for Medical Diagnosis. In: Crespo, J., Maojo, V. and Martin, F. (eds.), *Medical Data Analysis*, Springer-Verlag, Berlin Heidelberg, 146-152.

Larose, D., T. (2005). *Discovering Knowledge in Data: An introduction to Data Mining*. Wiley, U.S.A.

Larson, S., C. (1931). The Shrinkage of the Coefficient of Multiple Correlation. *Journal of Educational Psychology*, 22, 45-55.

Lawrence, S., Giles, C., L. and Tsoi, A., C. (1998), Symbolic Conversion, Grammatical Inference and Rule Extraction for Foreign Exchange Rate Prediction. In: Abu-Mostafa, A. and Weigend, A. (eds.), *Neural Networks in the Capital Markets NNCM96*, World Scientific Press, Singapore, 333-345.

Le Bras, A. and Andrews, D. (2004). Bank Rating Methodology: *Criteria Report*. Fitch (available at www.fitchratings.com).

Leifler., O. (2002). *Comparison of LEM2 and a Dynamic Reduct Classification Algorithm, Masters Thesis*. Faculty of Mathematics, Informatics and Mechanics, Warsaw University.

Lendasse, A., Wertz, V. and Verleysen, M. (2003). Model Selection with Cross-Validations and Bootstraps - Application to Time Series Prediction with RBFN Models. In: Kaynak, O. et al. (eds.), *Artificial Neural Networks and Neural Information Processing*, Springer-Verlag Berlin Heidelberg, 573-580.

Levich, M., Majnoni, G. and Reinhart, C. (2002). Introduction: Ratings, Rating Agencies and the Global Financial System: Summary and Policy Implications. *Ratings, Rating Agencies and the Global Financial System*. Kluwer Academic Publishers, Boston, 1-1 5.

Li, G., Liu, T. and Cheng, V., S. (2006). Classification of Brain Glioma by Using SVMs Bagging with Feature Selection. In: J., Li et al. (eds.), *Data Mining for Biomedical Applications*, Springer-Verlag, Berlin Heidelberg, 124-130.

Li, T., Qing, K., Yang, N. and Xu, Y. (2004). Study on Reduct and Core Computation in Incompatible Information Systems. In: Tsumoto, S., Slowinski, R., Komorowski, J. and Grzymala-Busse, J., W. (eds.), *Rough Sets and Current Trends in Computing. Fourth International Conference on Rough Sets and Current Trends in Computing*, Uppsala, Sweden, Springer-Verlag, Berlin Heidelberg, 471-476.

Li, Y., Shiu, S., C., Pal, S., K. and Liu, J., N. (2007). Using Approximate Reduct and LVQ in Case Generation for CBR Classifiers. In: Peters, F., J. and Skowron, A. (eds.), *Transactions on Rough Sets*, 7, Springer-Verlag, Berlin Heidelberg, 85-102.

Li, R. and Wang, Z. (2004). Mining Classification Rules Using Rough Sets and Neural Networks. *European Journal of Operational Research*, 157, 439-448.

Lin, T., Y. and Yin, P. (2004). Heuristically Fast Finding of the Shortest Reducts. In: Tsumoto, S., Slowinski, R., Komorowski, J. and Grzymala-Busse, J., W. (eds.), *Rough Sets and Current Trends in Computing. Fourth International Conference on Rough Sets and Current Trends in Computing*, Uppsala, Sweden, Springer-Verlag, Berlin Heidelberg, 465-470.

Ling, C., X. and Li., C. (1998). Data Mining for Direct Marketing: Problems and Solutions. In: *Proceedings 4th International Conference on Knowledge Discovery in Databases (KDD-98)*, New York.

Linoff, G. (1998). Which Way to the Mine? *Systems Management*, 42-44.

Liu, A. Y. (2004). *The Effect of Oversampling and Undersampling on Classifying Imbalanced Text Datasets, Maters Thesis.* University of Texas at Austin, U.S.

Liu, H. (2008). Homepage, *www.public.asu.edu/~huanliu/.*

Liu, H., Hussain, F., Tan, C., L. and Dash, M. (2002a). Discretization: An Enabling Technique. *Data Mining and Knowledge Discovery.* 6(4), 393-423.

Liu, H. and Motoda, H. (2000). *Feature Selection for Knowledge Discovery and Data Mining,* Kluwer Academic Publishers.

Liu, H. and Motoda, H. (2008). *Computational Methods of Feature Selection,* Chapman and Hall/CRC.

Liu, H., Motoda, H. and Yu, L. (2002b). Feature Selection with Selective Sampling. In: *Proceedings of the Nineteenth International Conference on Machine Learning*, 395-402.

Liu, H. and Setiono, R. (1996a). A Probabilistic Approach to Feature Selection - A Filter Solution. *13th International Conference on Machine Learning (ICML'96)*, Bari, Italy, 319-327.

Liu, H. and Setiono, R. (1996b). Feature Selection and Classification - A Probabilistic Wrapper Approach. *The 9th International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems* (IEA-AIE'96), Fukuoka, Japan, 419-424.

Liu, H. and Setiono, R. (1997). Feature Selection and Discretization. *IEEE Transactions on Knowledge and Data Engineering*, 9, 1-4.

Liu, H. and Yu, L. (2002). Feature Selection For Data Mining. *www.public.asu.edu/~huanliu/sur-fs02.ps.*

Lu, H., Setiono, R. and Liu, H. (1995). NeuroRule: A Connectionist Approach to Data Mining. In: Dayal, U. et al. (eds.). *Proceedings of the 21th international Conference on Very Large Data Bases*. Morgan Kaufmann Publishers, San Francisco, CA, 478-489.

Maloof, M. (2003). Learning When Data Sets are Imbalanced and When Costs are Unequal and Unknown. In: *ICML Workshop on Learning from Imbalanced Data Sets II.*

McNeil, A., Frey, R. and Embrechts, P. (2005). *Quantitative Risk Management: Concepts Techniques and Tools*, Princeton University Press, Princeton.

Mi, J., Wu, W. and Zhang, W. (2004). Approaches to Knowledge Reduction Based on Variable Precision Rough Set Model. *Information Sciences*, 159, 255-272.

Mitchell, M. (1998). *An Introduction to Genetic Algorithms*. The MIT Press.

Mosier, E., E. (1951). Symposium: The need and means of cross-validation, 1. Problem and Designs of Cross-Validation. *Educational and Psychological Measurement*, 11, 5-11.

Mosteller, F. and Tukey, J., W. (1968), Data Analysis, Including Statistics. In: Lindzey, G. and Aronson, E. (eds.), *Handbook of Social Psychology*, Addison-Wesley, Massachusetts, 148-156.

Murthy, S., K. (1998). Automatic Construction of Decision Trees From Data: A Multi-Disciplinary Survey. *Data Mining and Knowledge Discovery*, 2, 345-389.

Moodys. (2007). Http://www.moodys.com.

Nolan, J. (1997). Estimating the True Performance of Classification-Based {NLP} Technology. In: Burstein, J. and Leacock, C. (eds.), *From Research to Commercial Applications: Making {NLP} Work in Practice*, Association for Computational Linguistics, Somerset, New Jersey, 23-28.

Naisbitt, J. (1984). *Megatrends: Ten New Directions Transforming Our Lives.* Macdonald & Co, London and Sydney.

Naisbitt, J. (1988). *Megatrends: Ten New Directions Transforming Our Lives.* Grand Central Publishing.

Naisbitt, J. (1996). *Megatrends 2000. Ten New Directions for the 1990s*. Smithmark Publishers.

Oelericha, A. and Poddig, T. (2006). Evaluation of Rating Systems. *Expert Systems with Applications*, 30(3), 437-447.

Olecka, A. (2007). Beyond Classification: Challenges of Data Mining for Credit Scoring. In: Zhu,

X. and Davidson, I. (eds.), *Knowledge Discovery and Data Mining: Challenges and Realities with Real World Data.* Idea Group, U.S.A, 244-264.

Quinlan, J., R. (1986). Induction of Decision Trees. *Machine Learning,* 1, 81-106.

Quinlan, J. (1989). Unknown Attribute Values in Induction. In: Segre, A. (ed.), *Proceedings of the Sixth International Machine Learning Workshop*, Cornell, New York. Morgan Kaufmann.

Quinlan, J., R. (1993). *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Mateo, California.

Quinlan, J., R. (2007). Rulequest Research, http://www.rulequest.com/see5-info.html.

Pal, S., K. and Skowron, A. (1999). *Rough Fuzzy Hybridization: A New Trend in Decision-Making.* Springer-Verlag Telos .

Parsons, L., Haque, E. and Liu, H. (2004). Subspace Clustering for High Dimensional Data: A Review. *ACM SIGKDD Explorations Newsletter*, 6(1), 90-105.

Partnoy, F. (1999) The Siskel and Ebert of Financial Markets: Two Thumbs Down for the Credit Rating Agencies. *Washington University Law Quarterly*, 77, 619-712.

Partnoy, F. (2002). The Paradox of Credit Ratings. *Ratings, Rating Agencies and the Global Financial System.* Kluwer Acedemic Publishers, Boston, 65-85.

Partnoy, F. (2006). How and Why Credit Rating Agencies are Not Like Other Gatekeepers. Financial Gatekeepers: Can They Protect Investors? In: Fuchita, Y. and Litan, R., E. (eds.), *Brookings Institution Press and the Nomura Institute of Capital Markets Research.*

Pasiouras, F., Gaganis, C. and Zopoundis, C. (2006). The Impact of Bank Regulations, Supervision, Market Structure, and Bank Characteristics on Individual Bank Ratings: A Cross-Country Analysis. *Review of Quantitative Finance and Accounting*, 27, 403-438.

Pawlak, Z. (1982). Rough Sets. *International Journal of Computer and System Sciences*, 11, 341-356.

Pawlak, Z. (1991). *Rough Sets: Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers.

Pawlak, Z. (2004). Some Issues on Rough Sets. In: Peters, F., J. and Skowron, A. (eds.), *Transactions on Rough Sets*, 1, Springer-Verlag, Berlin Heidelberg, 1-58.

Pawlak, Z. (2005). A Treatise on Rough Sets. In: Peters, F., J. and Skowron, A. (eds.), *Transactions on Rough Sets*, 4, Springer-Verlag, Berlin Heidelberg, 1-17.

Peavy, J. (1984). Forecasting Industrial Bond Rating Changes: A Multivariate Model. *Review of Business and Economic Research*, 19, 2.

Peters J., F. and Skowron, A. (2007). *Transactiona on Rough Sets 5*, Springer-verlag Berlin Heidelberg.

Piatetsky-Shapiro, G. (2000). Knowledge Discovery in Databases: 10 Years After. S*IGKDD Explorations, Newsletter*, 1(2), 59-61.

Pinches, G. and Mingo, K. (1973). A Multivariate Analysis of Industrial Bond Ratings. *The Journal of Finance*, 28(1), 1-1 8.

Pinches, G. and Singleton, J. (1978). The Adjustment of Stock Prices to Bond Rating Changes. *The Journal of Finance*, 33(1), 29-44.

Poon, W., P., H., Firth, M. and Fung, H. (1999). A Multivariate Analysis of the Determinants of Moody's Bank Financial Strength Ratings. *Journal of International Financial Markets, Institutions and Money*, 9(3), 267-283.

Poon, W., P., H. (2003). Are Unsolicited Credit Ratings Biased Downward? *Journal of Banking and Finance*, 27(4), 593-614.

Poon, W., P., H. and Firth, M. (2005). Are Unsolicited Credit Ratings Lower? International Evidence From Bank Ratings. *Journal of Business Finance & Accounting*, 32, (9-10), 1741-1771.

Poon, W., P., H. and Chan, K., C. (2008). An Empirical Examination of the Informational Content of Credit Ratings in China. *Journal of Business Research*, 61(7), 790-797.

Potter, M. (2004). *Set Theory and Its Philosophy: A Critical Introduction*. Oxford University Press, USA.

Power, D., J. (2002). *Decision Support Systems: Concepts and Resources for Managers*. Quorum Books.

Power, D., J. (2008). *A Brief History of Decision Support Systems.* http://dssresources.com/history/dsshistory.html.

Provost, F. (2000). Machine Learning from Imbalanced Data Sets. Invited paper for the *AAAI'2000 Workshop on Imbalanced Data Sets.*

Puuronen, S., Tsymbal, A. and Skrypnyk, I. (2001). Correlation-Based and Contextual Merit-Based Ensemble Feature Selection. *Lecture Notes in Computer Science*, 2189, 135-144.

Rao, P., S., R., S. (2000). *Sampling Methodologies in Statistical Science.* Chapman and Hall/CRC, London.

Raudys, S., J. and Jain, A., K. (1991). Small Sample Size Effects in Statistical Pattern Recognition: Recommendations for Practitioners. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13, 252-262.

Raveh, A. (2000). The Greek Banking System: Reanalysis of Performance. *European Journal of Operational Research*, 120(3), 525-534.

Raymond, R., C. (1966). Use of the Time-sharing Computer in Business Planning and Budgeting, *Management Science*, 12(8), B363-B381

Reilly, F. and Joehnk, M. (1976). The Association Between Market-Determined Risk Measures for Bonds and Bond Ratings. *The Journal of Finance*, 31(5), 1387-1403.

Reuters. (2008a). http://www.reuters.com.

Reuters. (2008b). http://www.reuters.com/article/pressRelease/idUS108706+08-Apr-2008+BW20080408.

Ridgeway, G. (2002). Looking for lumps: Boosting and Bagging for Density Estimation. *Computational Statistics and Data Analysis,* 38, 379-392.

Robnik-Šikonja, M. and Kononenko, I. (1997). An Adaptation of Relief for Attribute Estimation in Regression. In: Fisher, D. (ed.), *Machine Learning: Proceedings of the Fourteenth International Conference (ICML'97)*, Morgan Kaufmann Publishers, 296-304.

Robnik-Šikonja, M. and Kononenko, I. (2003). Theoretical and Empirical Analysis of ReliefF and RReliefF. *Machine Learning*, 53(1-2), 23-69.

Rooney, D., Hearn, G. and Ninan, A. (2005). *Handbook on the Knowledge Economy.* Cheltenham/Edward Elgar.

Rough Sets Data Base System. (2008). http://rsds.univ.rzeszow.pl/.

Roy, A. (2000). A Theory of the Brain: There are Parts of the Brain that Control Other Parts. In: *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, 2, 81-86.

Rudnicki, W., R. and Komorowski, J. (2004). Feature Synthesis and Extraction for the Construction of Generalised Properties of Amino Acids. In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymala-Busse, J., W. (eds.), *Fourth International Conference on Rough Sets and Current Trends in Computing*, Uppsala, Sweden, Springer-Verlag, Berlin Heidelberg, 786-791.

Sahajwala, R. and Van den Bergh, P. (2000). Supervisory Risk Assessment and Early Warning Systems. *BASEL Committee on Banking Supervision Working Papers Working papers 4,* Bank for International Settlements.

Sealy, T., S. (1997). *International Country Risk Guide*. The PRS Group, 43, 10.

SEC. (2003). U.S. Securities and Exchange Commission. *Report on the Role and Function of Credit Rating Agencies in the Operation of the Securities Markets.*

Sewell, M. (2007). Ensemble Methods. http://www.machinelearning.net/ensembles/

Shannon, C., E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal*, 27, 379-423 and 623-656.

Shao., J. (1988). On Re-sampling Methods for Variance and Bias Estimation in Linear Model. *Annals of statistics*, 16, 986-1008.

Shao, J. (1993). Linear Model Selection by Cross Validation. *Journal of the American Statistical Association*, 89, 486-494.

Shao, J. and Tu, D. (1995). *The Jackknife and Booststrap*. Springer-Verlag, New York.

Shen, Q. and Jensen, R. (2007). Rough Sets, Their Extensions and Applications. *International Journal of Automation and Computing (IJAC)*, 4(3), 217-218.

Shin, K. and Han, I. (2001). A Case-Based Approach Using Inductive Indexing for Corporate Bond Rating. *Decision Support Systems*, 32(1), 41-52.

Sima, C. and Dougherty, E., R. (2006). Optimal Convex Error Estimators for Classification. *Pattern Recognition*, 39, 1763-1780.

Simon, F., H. (1971). *Prediction Methods in Criminology*. Home Office Research Study, 7. Her Majesty's Stationery Office, London.

Skowron, A. and Rauszer, C. (1992). The Discernibility Matrices and Function in Information Systems. In: Słowiński, R. (ed.), *Intelligent Decision Support - Handbook of Applications and Advances of the Rough Sets Theory*, Kluwer Academic Publishers, Dordrecht, 331-362.

Skurichina, M. and Duin R., P., W. (1998). Bagging for Liner Classifiers. *Pattern Recognition*, 31, 7, 909-930.

Ślęzak, D. and Wróblewski, J. (2003). Order Based Genetic Algorithms for the Search of Approximate Entropy Reducts. In: Wang, G. et al. (eds.), *Proceedings of Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing: 9th International Conference, RSFDGrC'03*, Chongqing, China, 308-311.

Ślęzak, D. and Ziarko, W. (2003) Variable Precision Bayesian Rough Set Model. In: Wang, G., Liu, Q., Yao, Y. and Skowron, A. (eds.), *Proceedings of the 9-th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing (RSFDGrC'03)*, Chongqing, China, Springer-Verlag, Heidelberg, 312-315.

Ślęzak, D. (2004). The Rough Bayesian Model for Distributed Decision Systems. In: Tsumoto, S., Slowinski, R., Komorowski, J. and Grzymala-Busse, J., W. (eds.), *Rough Sets and Current Trends in Computing. Fourth International Conference on Rough Sets and Current Trends in Computing*, Uppsala, Sweden, Springer-Verlag, Berlin Heidelberg, 384-393.

Ślęzak, D., Wang, G., Szczuka, M., S., Düntsch, I. and Yao, Y. (2005a). *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, 10th International Conference, RSFDGrC 2005*, Regina, Canada.

Ślęzak, D., Yao, J., Peters, J., F., Ziarko, W. and Hu, X. (2005b). *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, 10th International Conference, RSFDGrC 2005*, Regina, Canada.

Słowiński, R. (1992). *Intelligent Decision Support: Handbook of Applications and Advances in Rough Set Theory*, Kluwer Academic Publishers, Dordrecht.

Słowiński, R. 1993. Rough Set Learning of Preferential Attitude in Multi-Criteria Decision Making. In: Komorowski H. J. and Ras Z. W. (eds.), *Proceedings of the 7th international Symposiu on Methodologies For intelligent Systems*, Springer-Verlag, London, 689, 642-651.

Smith, M., F. (1983). Sampling Considerations In Evaluating Cooperative Extension Programs. *Florida Cooperative Extension Service Bulletin.* Institute of Food and Agricultural Sciences, University of Florida.

Smith, R., C. and Walter, I. (2002). Rating Agencies: Is there an Agency Issue? *Ratings, Rating Agencies and the Global Financial System.* Kluwer Acedemic Publishers, Boston, 289-318.

Song, L., Smola, A., Gretton, A., Borgwardt, K., M. and Bedo, J. (2007). Supervised Feature Selection Via Dependence Estimation. In: Ghahramani, Z. (ed.), *Proceedings of the 24th international Conference on Machine Learning (ICML '07)*, Corvalis, Oregon, 227, 823-830.

Standard and Poors. (2008). http://www2.standardandpoors.com.

Stefanowski, J. (1998). The Rough Set Based Rule Induction Technique for Classification Problems. In: *Proceeding of the 6th European Conference on Intelligent Techniques and Soft Computing EUFIT 98*, Aachen, 109-113.

Stefanowski, J. (2004). The Bagging and n^2-classifiers Based on Rules Induced by MODLEM. In: Tsumoto et al. (eds.), *Proceedings of the 4th International Conference on, Rough Sets and Current Trends in Computing*, Uppsala, Sweden, 488-497.

Stefanowski, J. (2007). Combined Classifiers, Rule Induction and Rough Sets, in: Peters J., F. and Skowron, A. (eds.), *Transactiona on Rough Sets 5*, Springer-verlag Berlin Heidelberg, 329-350.

Stempel, J. (2007). Countrywide Plunges on Downgrade, Bankruptcy Fear. Reuters, Aug 15th.

Stone, M. (1974). Cross-Validatory Choice and Assessment of Statistical Predictions. *Journal of the Royal Statistical Society, Series B,* 36, 2, 111-147.

Susmaga, R. (2004). Tree-Like Parallelization of Reduct and Construct Computation. In: Tsumoto, S., Slowinski, R., Komorowski, J. and Grzymala-Busse, J., W. (eds.), *Rough Sets and Current Trends in Computing. Fourth International Conference on Rough Sets and Current Trends in Computing*, Uppsala, Sweden, Springer-Verlag, Berlin Heidelberg, 455-464

Sylla, R. (2002). An Historical Primer on the Business of Credit Ratings. *Ratings, Rating Agencies and the Global Financial System.* Kluwer Academic Publishers, Boston, 19-41.

Tay, F., Shen, L. and Cao, L. (2003). *Ordinary Shares, Exotic Methods: Financial Forecasting Using Data Mining Techniques*. World Scientific, New Jersey, USA.

Thomassey, S. and Fiordaliso, A. (2006). A Hybrid Sales Forecasting System Based on Clustering and Decision Trees. *Decision Support Systems*, 42, 1, 408-421.

Tibshirani, R. (1996). Bias, Variance, and Prediction Error for Classification Rules. *Technical Report*, Statistics Department, University of Toronto.

Thomson Reuters. (2008). http://thomsonreuters.com.

Toussaint, G. and Sharpe, P. (1975). An Efficient Method for Estimating the Probability of Misclassification Applied to a Problem in Medical Diagnosis. *Computers in Biology and Medicine*, 4, 269-278.

Trumbower, R., D., Karan S., R. and Faghri, P., D. (2006). Identifying Offline Muscle Strength Profiles Sufficient for Short-Duration Fes-Lce Exercise: A Pac Learning Model Approach. *Journal of Clinical Monitoring and Computing*, 20(3), 209-220.

Tsumoto, S., Słowiński, R., Komorowski, J. and Grzymala-Busse, J., W. (2004). *Fourth International Conference on Rough Sets and Current Trends in Computing*, Uppsala, Sweden, Springer-Verlag, Berlin Heidelberg.

Turban, E. (1967). The Use of Mathematical Models in Plant Maintenance Decision Making. *Management Science*, 13(6), B342-B359.

Valiant, L., G. (1985). Learning Disjunctions of Conjunctions. In: Joshi, A., K. (ed.). *Proceedings of the International Joint Conference on Artificial Intelligence*, Los Angeles, Morgan Kaufmann, 560-566.

Van Roy, P. (2006). Is There a Difference Between Solicited and Unsolicited Bank Ratings and if so, Why? *National Bank of Belgium, Research Series*, 200603-1.

Vapnik, V., N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag.

Wang, K. and Liu, B. (1998). Concurrent Discretization of Multiple Attributes. In, *Pacific-Rim International Conference on AI*, 250-259.

Wang, X., Yang, J., Peng, N. and Teng, X. (2005). Finding Minimal Rough Set Reducts with

Particle Swarm Optimization. In: Ślezak, D., et al. (eds.), *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*, Springer-Verlag, Berlin Heidelberg, 451-460.

Webb, G., I. (2000). MultiBoosting: A Technique for Combining Boosting and Wagging. *Machine Learning*, 40, 2, 159-196.

Weiss, S., M. (1991). Small Sample Error Rate Estimation for k-NN classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3), 285-289.

Weiss, S., M. and Kulikowski, C., A. (1991). *Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neaural Nets, Machine Learning, and Expert Systems*. Morgan Kaufmann, California U.S.A.

Weiss, S., M. and Indurkhya, N. (1998). *Predictive Data Mining: A practical Guide*. Morgan Kaufmann, California U.S.A.

West, R., R. (1970). An Alternative Approach to Predicting Corporate Bond Ratings. *Journal of Accounting Research*, 8, 1, 118-125.

Wherry, R., J. (1951). Symposium: The need and means of cross-validation, 4. Comparison of Cross Validation with Statistical Inferences of Betas and Multiple *R* from a Single Sample. *Educational and Psychological Measurement*, 11, 23-28.

White, L. (2002). The Credit Rating Industry: An Industrial Organization Analysis. *Ratings, Rating Agencies and the Global Financial System*, Kluwer Academic Publishers, Boston, 41-65.

Whitehorn M. and Whitehorn, M. (1999). *Business Intelligence: The IBM Solution: Data Warehousing and OLAP.* Springer.

Widz, S., Kenneth, R. and Ślezak, D. (2004). An automated Multi-spectral MRI Segmentation Algorithm Using Approximate Reducts. In: Tsumoto, S., Slowiński, R., Komorowski, J., Grzymala-Busse, J., W. (eds.), *Fourth International Conference on Rough Sets and Current Trends in Computing*, Uppsala, Sweden. Springer-Verlag, Berlin Heidelberg, 815-824.

Wisnowski, J., W., Simpson J., R., Douglas, C., Montgomery D., C. and Runger, G., C. (2003). Resampling Methods for Variable Selection in Robust Regression. *Computational Statistics and Data Analysis*, 43, 341-355.

Wittgenstein, L. (2001). *Remarks on the Foundations of Mathematics*, Oxford.

Wolpert, D., H. and Macready, W., G. (1999). An Efficient Method To Estimate Bagging's Generalization Error. *Machince Learning*, 35(1), 41-55.

Wong, S. and Ziarko, W. (1986). On Learning and Evaluation of Decision Rules in the Context of Rough Sets. In: Ras, Z. W. and Zemankova, M. (eds.), *Proceedings of the ACM SIGART First International Symposium on Methodologies for Intelligent Systems Knoxville (ISMIS'86)*, Tennessee, USA, 308-324.

Wroblewski, J. (1995). Finding Minimal Reducts using Genetic Algorithms. In: Wang, P., P. (ed.), *Proceedings of the International Workshop on Rough Sets Soft Computing at Second Annual Joint Conference on Information Sciences (JCIS'95)*, Wrightsville Beach, North Carolina, 186-189.

Wroblewski, J. (1998). Genetic Algorithm in Decomposition and Classification Problems. In: Polkowski, L. and Skowron, A. (eds.), *Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems*, Physica-Verlag, Heidelberg, 472-492.

Wu, J. (2002). *The Value in Mining Data.* Business Intelligence, DM Review, http://www.dmreview.com/news/4618-1.html

Yamane, T. (1967). *Statistics, An Introductory Analysis*. Harper and Row, New York, 2nd edition.

Yang, X., Yang, J., Wu, C. and Yu, D. (20080. Dominance-Based Rough Set Approach and Knowledge Reductions in Incomplete Ordered Information System. *Information Sciences,* 178(4), 1219-1234.

Zadeh, L., A. (1965). Fuzzy Sets. *Information and Control*, 8, 338-353.

Zhang, G., Hu, M., Y., Patuwo, B., E. and Indro, D., C. (1999). Artificial Neural Networks in Bankruptcy Prediction: General Framework and Cross-Validation Analysis. *European Journal of Operational Research*, 116(1), 16-32.

Zhang, X., Wu, J., Lu, T. and Jiang, Y. (2007). A Discretization Algorithm Based on Gini Criterion. International Conference on Machine Learning and Cybernetics, 5, 19-22, 2557-2561

Zhou, J., Foster, D., P., Stine, R., A. and Ungar, L. H. (2006). Streamwise Feature Selection. *Journal of Machine Learning Research*, 7, 1861-1885.

Ziarko, W. (1993a). Variable Precision Rough Set Model. *Journal of Computer and System Sciences*, 46, 39-59.

Ziarko, W. (1993b). Analysis of Uncertain Information in the Framework of Variable Precision Rough Set. *Foundations of Computing and Decision Sciences*, 18, 381-396.

Ziarko, W. (1998). Approximation Region Based Decision Tables. In: Polkowski, L. and Skowron, A. (eds.), *Proceedings of the First International Conference on Rough Sets and Current Trends in Computing, RSCTC'98*, Warsaw, Poland, Springer-Verlag, Berlin Heidelberg, 178-185.

Ziarko, W. (1999). Decision Making with Probabilistic Decision Tables, in: Zhong, N., Skowron, A., Ohsuga, S. (eds.), *New Directions in Rough Sets, Data-Mining, and Granular-Soft Computing. Proceedings 7th International Workshop, RSFDGrC'99*. Yamaguchi, Japan. Springer-Verlag, Berlin Heidelberg, 463-471.

Ziarko, W. (2001). Probabilistic Decision Tables in the Variable Precision Rough Set Model. *Computational Intelligence*, 17(3), 593-603.

Ziarko, W. (2003). Evaluation of Probabilistic Decision Tables. In: Wang, G. et al. (eds.), *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, 9th International Conference, RSFDGrC'03*, Chongqing, China. Springer-Verlag, Berlin Heidelberg, 189- 196.

Ziarko, W. and Xiao X. (2004). Computing Minimal Probabilistic Rules from Probabilistic Decision Tables: Decision Matrix Approach. In: Favela et al. (eds.), *Advances in Web Intelligence*, Springer-Verlag, Berlin Heidelberg, 84-94.

Zighed, A., D., Auray, J., P. and Duru, G. (1992). Sipina: Méthode et logiciel. *Lacassagne.*

Zighed, D., A., Rabaséda, S. and Rakotomalala, R. (1998). FUSINTER: A method for Discretization of Continuous Attributes. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6, 3, 307-326.

Zopoundis, C. and Doumpos, M. (2002). Multi-group discrimination Using Multi-criteria analysis: Illustrations from the Field of Finance. European Journal of Operational Research, 139, 371-389.

Zorn, T., E. and Taylor, J., R. (2003). Knowledge Management and/as Organizational Communication. In: Tourish, D. and Hargie, O. (eds.). *Key Issues in Organizational Communication*. London and New York: Routledge.