UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

# Dataset Selection for Aggregate Model Implementation in Predictive Data Mining

by

Patricia Elizabeth Nalwoga Lutu

Thesis submitted in partial fulfilment of the requirements for the degree of

Philosophiae Doctor

in the Faculty of Engineering Built Environment and Information Technology

The University of Pretoria

Pretoria

September 2010

Title: Dataset Selection for Aggregate Model Implementation in Predictive Data Mining

Author: Patricia Elizabeth Nalwoga Lutu

## Abstract

Data mining has become a commonly used method for the analysis of organisational data, for purposes of summarizing data in useful ways and identifying non-trivial patterns and relationships in the data. Given the large volumes of data that are collected by business, government, non-government and scientific research organizations, a major challenge for data mining researchers and practitioners is how to select relevant data for analysis in sufficient quantities, in order to meet the objectives of a data mining task. This thesis addresses the problem of dataset selection for predictive data mining. Dataset selection was studied in the context of aggregate modeling for classification.

The central argument of this thesis is that, for predictive data mining, it is possible to systematically select many dataset samples and employ different approaches (different from current practice) to feature selection, training dataset selection, and model construction. When a large amount of information in a large dataset is utilised in the modeling process, the resulting models will have a high level of predictive performance and should be more reliable. Aggregate classification models, also known as ensemble classifiers, have been shown to provide a high level of predictive accuracy on small datasets. Such models are known to achieve a reduction in the bias and variance components of the prediction error of a model. The research for this thesis was aimed at the design of aggregate models and the selection of training datasets from large amounts of available data. The objectives for the model design and dataset selection were to reduce the bias and variance components of the prediction error for the aggregate models.

Design science research was adopted as the paradigm for the research. Large datasets obtained from the UCI KDD Archive were used in the experiments. Two classification algorithms: See5 for classification tree modeling and K-Nearest

Neighbour, were used in the experiments. The two methods of aggregate modeling that were studied are One-Vs-All (OVA) and positive-Vs-negative (pVn) modeling. While OVA is an existing method that has been used for small datasets, pVn is a new method of aggregate modeling, proposed in this thesis. Methods for feature selection from large datasets, and methods for training dataset selection from large datasets, for OVA and pVn aggregate modeling, were studied.

The experiments of feature selection revealed that the use of many samples, robust measures of correlation, and validation procedures result in the reliable selection of relevant features for classification. A new algorithm for feature subset search, based on the decision rule-based approach to heuristic search, was designed and the performance of this algorithm was compared to two existing algorithms for feature subset search. The experimental results revealed that the new algorithm makes better decisions for feature subset search. The information provided by a confusion matrix was used as a basis for the design of OVA and pVn base models which are combined into one aggregate model. A new construct called a *confusion graph* was used in conjunction with new algorithms for the design of pVn base models. A new algorithm for combining base model predictions and resolving conflicting predictions was designed and implemented. Experiments to study the performance of the OVA and pVn aggregate models revealed the aggregate models provide a high level of predictive accuracy compared to single models. Finally, theoretical models to depict the relationships between the factors that influence feature selection and training dataset selection for aggregate models are proposed, based on the experimental results.

## Key words:

**data mining, predictive modeling, classification, model aggregation, ensemble classifiers, OVA classification, pVn classification, dataset selection, feature selection, variable selection, bias reduction, variance reduction, large datasets, dataset sampling, dataset partitioning.**

Thesis supervisor: Prof. A.P. Engelbrecht
Department: Department of Computer Science
Degree: Philosophiae Doctor
(Doctor of Philosophy)

# Dedication

*This thesis is dedicated in loving memory to my parents*

*Omwami Wilson Sebowa Lutu*

*and*

*Omumbejja Kasalina Zalwango Lutu.*

# Acknowledgements

I wish to express my sincere gratitude to my research supervisor Prof. Andries Engelbrecht. Thank you for all the time you have dedicated to advising me on my research activities, and especially for reading my thesis with what I call the magnifying glass! I have been privileged to benefit from your extensive research and authoring experience.

I also wish to express my sincere gratitude to the external examiners for taking the time to examine this thesis. Thank you for the encouraging feedback on the thesis contents, and the constructive advice you have given for the final thesis revisions.

I wish to thank the following people: Prof. Carina de Villiers, head, Department of Informatics. Thank you for all the support you have given me over the last five years, especially the research leave. My colleagues in the Statistics department: Dr. Raphael Kasonga, Mev. Judy Coetsee, and Mev. Dorothea Corbett. Thank you for the numerous advice you have given me on statistical inference. My colleagues in the library: Mnr. Danie Malan, Ma. Tebogo Mogakane, and Mev. Gerda Ehlers. Thank you for all the assistance you have given me in acquiring research articles and books.

I also wish to thank all the people and angels that have given me spiritual guidance over the years. Prof. Ojelanki Ngwenyama: Thank you for the spiritual teachings and for introducing me to Toulmin's work.

Last but not least, I wish to thank my son Subi for patiently supporting and encouraging me with my research. You often asked me: 'Mummy, when are you going to finish that thesis?' Well Subi, after five and half years, I have completed the thesis.

# Contents

# List of Figures

# List of Tables

*The Infinite Intelligence*

*is beyond human understanding.*

*The Infinite Intelligence*

*created the universe:*

*all that we perceive,*

*and all that we do not perceive.*

*The Infinite Intelligence*

*exists in silence,*

*gives in silence,*

*and*

*loves in silence.*

# Chapter 1
# Introduction

> 'Into thy presence we come, not by the works we have done, but by the grace and the grace alone, into thy presence we come.' (Benjamin Dube, 2007)

The rate of growth in data volumes stored by organisations continues to grow at a phenomenal rate. For many organisations, the amount of data stored in the data warehouses is in the region of many terabytes. At the extreme end, there are organizations whose data warehouse sizes are in the region of 50 terabytes or more. Data warehouses and business intelligence tools for data analysis have become a necessity in many organizations due to the ever increasing competitive nature of doing business in the information age.

Real-time data warehousing is not uncommon. Given the large volumes of data that are collected by business, government, non-government and scientific research organizations, a major challenge for data mining researchers and practitioners is how to select sufficient amounts of data for analysis, in order to meet the objectives of a data mining task. As second major challenge is design of fast methods of data analysis. The central argument of this thesis is that there is a need to employ methods of dataset selection that provide as much information as possible to the data mining algorithms. The dataset selection methods need to be coupled with fast and reliable methods of data analysis for the creation of reliable data mining models. The thesis concentrates on predictive data mining algorithms for classification tasks. Methods for feature selection, dataset selection, and model construction, are proposed and studied. It is argued and demonstrated that these methods result in the construction of reliable, high performance classification models for data mining from very large datasets.

## 1.1 Motivation for the research

Data mining is commonly defined as a collection of methods for the analysis of observational data (Hand et al, 2001; Smyth, 2001). The methods used in data

mining for purposes of data analysis originate mainly from the fields of Computer Science, Statistics and Operations Research. Several researchers (e.g. Giudici, 2003; Smyth, 2001; Hand, 1999) have observed that data mining lies at the interface between Computer Science and Statistics. More recently, Olafsson et al (2008) have discussed the contributions of Operations Research to data mining. Formally, Hand et al (2001) have defined data mining as follows.

> 'Data mining is the analysis of (often large) observational datasets, to find unsuspected relationships, and to summaries the data in novel ways that are both understandable and useful to the data owner.'

From the Computer Science perspective, the main contribution to the field of data mining has been algorithms from the area of machine learning. The algorithms that originate from machine learning are employed in the implementation of local and global models from observational data (Giudici, 2003; Smyth, 2001). From Statistics, the parent field for data analysis, the main contribution has been the large body of knowledge on the summarisation of data that is generated by stochastic processes, estimation of descriptive and predictive models for stochastic processes, and the evaluation of the estimated models (Giudici, 2003; Smyth, 2001). From Operations Research the most distinctive contribution has been optimisation methods that can be employed in various modeling activities and especially in the selection of the best model from a set of possible models (Olafsson et al, 2008; Osei-Bryson, 2004, 2007, 2008; Fu et al, 2003, 2006).

The research for this thesis was directed at the selection of training data from large datasets for purposes of aggregate modeling. Aggregate modeling is concerned with the creation of many base models which are then combined into one aggregate model. From a computational perspective, it can be argued that the processing time complexity of most machine learning algorithms employed in data mining is typically non-linear. This property of machine learning algorithms places a limit on the amount of data that can be processed in order to provide results within a reasonable and acceptable amount of time. The time complexity of machine learning algorithms for data mining is not the only issue to consider when faced with large data volumes. From a statistical perspective, it is not desirable to use a very large amount of data in the process of estimating one model. In the past there have been several negative comments, especially originating from the Statistics community, directed at various

research directions in data mining. In 1998, Hand (1998) made the following observation.

> '..the term data mining is … synonymous with data dredging.. and has been used to describe the process of trawling through data in the hope of identifying patterns. It has a derogatory connotation because a sufficiently exhaustive search will certainly throw up some patterns of some kind … the object of data analysis is not to model the fleeting random patterns of the moment, but to model the underlying structures which give rise to consistent and replicable patterns. ..the term data mining conveys the sense of naïve hope vainly struggling against the cold realities of chance.'

Both the computational perspective and the statistical perspective as discussed above, point to the need for data reduction. It is the author's opinion that research efforts should be directed towards the study of methods for the selection of relevant data that can be used to create models that provide a high level of predictive performance.

The problem that the work reported in this thesis aims to solve is the design of methods for training dataset selection, for purposes of creating many base models which can be combined into one aggregate model. Such an aggregate model should provide a higher level of predictive performance compared to a single model created from a single training dataset. This approach should lead to the usage of significantly large amounts of data while at the same time avoiding the computational and statistical problems highlighted above. The idea of using aggregate models is not new. As far back as 1996, Breiman (1996) proposed bootstrap aggregation as a method of improving predictive accuracy for models constructed from small datasets. At the present time, there are many research efforts directed at the design of aggregate predictive models.

## 1.2 Current debates and practices in data mining from large datasets

One approach that has been investigated by researchers in predictive data mining is the use of very large training datasets obtained from very large datasets. Training datasets of several millions records have been processed using very powerful machines (Chawla et al, 2001; Hall et al, 2000). The rationale behind this approach is

that when very large amounts of data are processed, then as much as possible of the information gathered about a subject area is incorporated in the model construction. An obvious disadvantage is that the model construction process takes a very long time. A second and more serious disadvantage may be explained through statistical theory. Smyth (2001), Hand et al (2001), and Hand (1998) have cautioned that when training datasets are very large it becomes very difficult to distinguish between noise and real structure in the data.

Another explanation of this disadvantage comes from the machine learning literature. Dietterich (1995) has observed that for classification problems, a predictive model which has a very high level of training accuracy is not necessarily reliable when put to practical use. The main purpose of predictive modeling is to process data in order to find relationships that can be generalized. If an inductive algorithm is used to create a predictive model from a very large amount of data it will minimize the training error. However, there is a very high risk that it will fit the predictive model to the noise in the training data by memorizing peculiarities of the training data rather than finding a general predictive rule. This phenomenon is called *overfitting* (Smyth, 2001; Dietterich, 1995). Prediction models based on very large amounts of data should therefore be treated with caution.

A second approach to predictive data mining from large datasets is to take a single sample from a very large dataset and use it for model construction. Additional samples are then taken for validation and testing (Domingos, 2001; Kohavi et al, 2004; Provost et al, 1999; John & Langley, 1996). This approach has also received much attention from theoretical research in statistical pattern recognition and machine learning, for example, Valiant (1984). The main advantage of this approach is that the training sample is typically much smaller than the large dataset, and so, is much faster to process. An obvious disadvantage is that the bulk of the data is discarded and only a small fraction of the data is used for making decisions about feature selection, model structure and model performance. A second disadvantage is that sampling results in stochasticity. If another random sample were to be taken, the selected features, model structure and measured performance may be significantly different.

A third approach to predictive data mining from large datasets is to partition a large dataset, construct a predictive model based on each partition and then combine the different models into one aggregate model (Chawla et al, 2001; Hall et al, 2000;

Chan & Stolfo, 1998). One obvious advantage of this approach is that partitioning attempts to use as much of the available data as possible. Several researchers who have studied aggregate modeling from large datasets (e.g. Chawla et al, 2001; Hall et al, 2000; Chan & Stolfo, 1998) have argued that the performance of an aggregate model normally exceeds that of a single model constructed from a single large training sample. On the other hand, other researchers (e.g. Hall et al, 2000; Ali & Pazzani, 1996) have argued that there are various domains where partitioning does not result in any performance gains and may in fact result in loss of accuracy.

The use of aggregate models has been studied by many researchers (e.g. Osei-Bryson et al, 2008; Sun & Li, 2008; Ooi et al, 2007; Neagu et al, 2006; Kim et al, 2002; Chan & Stolfo, 1998; Breiman, 1996; Krogh & Veldelsby, 1995; Kwok & Carter, 1990) even though these studies have not always been in the context of very large datasets. A large body of literature and evidence exists to support the claims that aggregate modeling often leads to improved predictive performance. Given the foregoing observations, it is the author's opinion that studies in dataset selection from large datasets should be directed towards improving the predictive accuracy of aggregate models.

## 1.3 Scope of the research

The title of this thesis makes reference to the term, *predictive data mining*. It is therefore important for the author to highlight the difference between predictive and non-predictive data mining.

Data mining tasks may be broadly divided into four categories, namely: exploratory data analysis (EDA), local methods for pattern detection and rule extraction, descriptive modeling, and predictive modeling (Hand et al, 2001). Exploratory data analysis is concerned with the exploration of data without any prior clearly articulated idea of what one is looking for, or any plan of what output needs to be generated. Pattern detection and rule discovery activities are concerned with the identification of regions of the instance space whose characteristics significantly differ from those of the other regions (e.g. association rule mining) or locating patterns of interest in data as is done in text mining (Hand et al, 2001). The objective of descriptive modeling is to create a model that describes the data or the process that generates the data (Hand et al, 2001). Examples of this include density estimation (estimation of the

overall probability distribution), cluster analysis (identification of naturally occurring groups in the data), segmentation (division of data into groups based on specified criteria) and, dependency modeling (description of the relationship between variables). For predictive modeling, the purpose is to create a model that may be used for the prediction of the value of the dependent variable, given the values of the independent (predictor) variables.

The term *predictive data mining* refers to data mining methods that create predictive models (Hand et al, 2001). Predictive models may be constructed to predict the values of a quantitative variable as in regression or to predict the values of a qualitative variable as in classification. The research reported in this thesis is primarily concerned with classification problems. As discussed in the last section, there is a large body of evidence to support the claim that aggregate modeling has the potential to improve classification performance. The scope of the research reported in this thesis is directed at classification methods that employ aggregate modeling.

In the data mining literature, Giudici (2003) has made a distinction between computational data mining and statistical data mining. The distinguishing characteristic between computational and statistical data mining is that while statistical data mining methods assume a specific probability distribution for the process that generates the data, computational data mining methods make no specific assumptions about the probability distribution for the data generating process. However for computational data mining and machine learning, there is the (not always stated) assumption that the data generating process is governed by a fixed but unknown probability distribution (Mitchell, 1997). The research reported in this thesis is aimed at computational data mining.

## 1.4 The claims of the thesis

The central argument of this thesis is that it is possible for predictive data mining to systematically select many dataset samples and employ different approaches (different from current practice) to feature selection, training dataset selection, and model construction. When a large amount of information in the large dataset is utilised in the modeling process, the resulting models should have a high level of predictive performance and should be reliable.

Ngwenyama (2007) has identified seven categories of scientific research claims. The first four claims identified by Ngwenyama (2007) are: (1) a scientific problem that has been solved (2) a general contribution to science (3) extension of a body of knowledge and (4) appropriateness of the research methodology. Ngwenyama (2007) has used the argumentation model by the philosopher Toulmin (Toulmin et al, 1979; Toulmin, 1958) to analyse the four categories of scientific research claims. In Toulmin's argumentation model (Toulmin et al, 1979; Toulmin, 1958) *claims* are supported by *data* (observations / evidence) and *warrants*. The *data* (observations / evidence) are the grounds on which the claim stands. *Warrants* consist of general rules of inference and existing theories that serve as bridges or connections between the *data* (observations / evidence) and the *claims*. *Warrants* are supported by *backings* which are the known authoritative sources from which the *warrants* are drawn. The claims of this thesis are presented in terms of Ngwenyama's (2007) categorisation and Toulmin's (1958) argumentation model. The scientific problem that has been solved and the general contributions to science are presented in this section. The extensions to the body of knowledge and the research paradigm are presented in the next two sections.

The first *claim* that is made in this thesis is that aggregate classification models based on One-versus-All (OVA) modeling (Ooi et al, 2007; Rifkin & Klautau, 2004) and positive-Versus-negative (pVn) modeling can be used to increase the amount of relevant data in the training datasets. Increasing training data through OVA and pVn modeling results in improved predictive performance compared to the use of a single model. OVA modeling involves the decomposition of a *k*-class prediction task into *k* 2-class prediction tasks. pVn modeling involves the decomposition of a *k*-class prediction task into *j (j<k)* prediction tasks. OVA and pVn aggregate models differ from the aggregate models commonly discussed in the literature (e.g. Osei-Bryson et al, 2008; Kim et al, 2002; Chan & Stolfo, 1998; Breiman, 1996; Krogh & Veldelsby, 1995; Kwok & Carter, 1990; Hansen & Salamon, 1990). Firstly, the aggregate models discussed in the literature cited above do not employ problem decomposition. Secondly, the training datasets used for the base models that constitute such aggregate models generally re-use the small amount of available data. The methods proposed in this thesis for the implementation of OVA and pVn aggregate models do not re-use training data, but rather, use a different training dataset for each base model. These methods result in high coverage of the instance space while at the same time avoiding the problems of data dredging and overfitting. Traditionally, data

dredging and overfitting are associated with the usage of large training datasets for single models. High coverage of the instance space provides more information for the prediction task which in turn results in high predictive performance.

The second *claim* of this thesis is that the performance of aggregate models can be improved when the training samples for the base models are purposefully designed to reduce the bias and variance components of the prediction error. The bias component of the prediction error reflects the level of error in the estimation process of the model. The variance component reflects the sensitivity of the model to the training sample used to estimate the model (Friedman, 1997; Geman et al, 1992).

The *warrants* and *backing* for the first and second *claim* are as follows: Based on statistical theory a random / stochastic process can be studied using many small samples of the data generated by the process in order to establish the underlying structure of that process. Secondly, theories have been formulated in machine learning and statistical pattern recognition to explain how prediction errors arise. Based on these theories, it is possible to select training datasets in such a way that the chances of error are significantly reduced. There have been various research efforts that use several samples in model construction and feature selection. Breiman (1996) has studied the use of many bootstrap samples from small datasets to implement classifier committees. Freund and Schapire (1997) have studied boosting through the sequential creation of many small training samples, where each successive training sample consists of a larger number of training instances that are difficult to predict correctly. Studies have been reported on dataset selection methods which are guided by information on the characteristics of the instance space (Chan & Stolfo, 1998; Kubat & Matwin, 1997). All the above studies have demonstrated that purposeful training dataset selection for base models can result in major improvements in the predictive performance of aggregate models.

The third *claim* of this thesis is that the use of many (relatively) small samples to measure correlations between the variables for the prediction task leads to a more reliable selection of the relevant features for the prediction task. The fourth *claim* of this thesis is that, when the domain-specific definitions of the strength of association between variables are incorporated into the feature selection decisions, good subsets of predictive features will be selected for the prediction task.

The *warrants* and *backing* for the third and fourth *claims* are as follows. Statistical theory tells that, when the correlation between two random variables is measured using one sample then if the sample is small, a small or large correlation coefficient could be purely due to chance (Smyth, 2001). On the other hand if the sample is large, a small correlation coefficient may appear to be statistically significant even though it has no practical significance (Cohen, 1988). For purposes of measuring the correlations between the predictive variables and the class variable, Bi et al (2003) have studied the use of many bootstrap samples for micro-array datasets, in order to achieve reliable feature subset selection. Even though the studies by Bi et al (2003) have been conducted on small datasets, the results of their studies indicate that there are benefits in using many small samples to establish feature relevance for prediction tasks. Research has been conducted on the incorporation of user preferences in algorithms for predictive modeling. Osei-Bryson (2004) has proposed the incorporation of user preferences in decision tree selection. Ooi et al (2007) and Yu and Liu (2004) have proposed the incorporation of user-specified preferences in feature selection methods. The foregoing observations provide motivation for the incorporation of domain-specific definitions of feature relevance into feature selection algorithms.

The fifth and final *claim* of this thesis is that research into aggregate model construction methods using different methods of sample composition and feature selection should lead to useful theories for the improvement of aggregate model performance. When the data available for model construction is small, as was typically the case in the past, statisticians invented effective methods of model construction, validation and testing (Mitchell, 1997; Cohen, 1995). Bootstrap sampling for example, is useful for purposes of creating several large samples which have the same statistical properties as the small sample from which they are generated (Cohen, 1995).

In this thesis the author further argues that, since at the present time very large amounts of data are available for data mining, it is productive to investigate (new) ways of predictive model construction coupled with new ways of dataset selection. It is the author's opinion that the following issues have not been sufficiently studied by researchers:

(1) The use of many samples drawn from very large datasets for purposes of feature selection.

(2) The use of sampling in conjunction with partitioning for purposes of dataset selection and aggregate model construction.
(3) The design of training dataset samples aimed at reducing bias and variance in the prediction error without the need to re-use training data.

The author further argues that when very large amounts of data are available, data mining researchers have at their disposal a great opportunity to conduct empirical studies of the factors, and the relationships between the factors that affect various aspects of predictive model design and construction. In the data mining literature, there seems to be a scarcity of clearly articulated theoretical models based on empirical studies that can help to explain the relationships between the factors that determine: (1) the quality of selected feature subsets, (2) the quality of selected dataset samples and, (3) the predictive performance of aggregate models. It should be pointed out however that for aggregate model construction, several researchers have conducted studies on various factors that affect aggregate model performance in the context of small datasets. Examples of these studies are Kwok and Carter (1990), Ali and Pazzani (1996), Breiman (1996), and Ho (1998).

The investigations of this thesis were directed at dataset selection methods from large datasets for purposes of aggregate model implementation. The main research question for the thesis was as follows:

*What methods of dataset selection can be used to obtain as much information as possible from large datasets while at the same time using training datasets of small sizes to create predictive models that have a high level of predictive performance?*

The investigation of the answers to the above question was conducted using the design science research paradigm which is described briefly in the following section and in detail in chapter 4. The design science research paradigm enabled the author to generate experimental *evidence* (*data*) to support the *claims* presented in this section.

## 1.5 Research paradigm

The research paradigm used for this research is design science research as described by March and Smith (1995), Hevner et al (2004), Vaishnavi and Kuechler

(2004/5), and Manson (2006). Design science research involves two distinct steps. In the first step, an artifact is created. In the second step, an analysis of the usage and performance of the artifact is conducted. The purpose of the analysis is to understand, explain, and possibly improve on one or more aspects of the artifact (Vaishnavi & Kuechler, 2004/5).

In the context of information systems, artifacts may be models (abstractions and representations), methods (algorithms and practices) and instantiations (implemented and prototype systems) (Hevner et al, 2004). Manson (2006) has summarised these views by observing that design science research is a process of using knowledge to design and create useful artifacts, and then using rigorous methods to analyse why, or why not, a particular artifact is effective. Scientific research is about generating knowledge. A design science research effort should therefore make a contribution to the knowledge base of the field. More specifically, the contributions of design science research could be:
(1) Constructs. These are the components of the conceptual vocabulary of the domain.
(2) Models. These are propositions expressing the relationships between the constructs / concepts of the research domain.
(3) Methods. This is the 'how-to' knowledge. It is specified in the form of steps used to perform a given task.
(4) Instantiations. This is the operationalisation of the constructs, models and methods to demonstrate that the models and methods can be implemented in a working system.
(5) Better theories.

Design science research was found to be appropriate for this thesis because the central argument is based on the development of methods for feature and training dataset selection as well as the design and creation of predictive models.

## 1.6 Research contributions

It was stated in the last section that design science research should make a contribution to the knowledge base of the field. The *claims* of the research contributions of this thesis to the knowledge base of predictive data mining are summarised in this section in terms of the expectations of design science research

outputs. Two additional components in Toulmin's (1958) argumentation model are *qualifiers* and *rebuttals*. *Qualifiers* are used to limit the strength of a *claim* and *rebuttals* provide an elaboration for the *qualifiers*. A detailed discussion of the *claims* of the research contributions and, the *qualifiers* and *rebuttals* identified by the author are presented in chapter 11 of this thesis.

### 1.6.1 Methods and instantiations

Methods for feature selection from large datasets were studied. The studies involved testing methods of reliable feature selection that involve the use of robust measures of correlation, the use of many samples to measure correlations, and the use of statistical tests, such as the t-test and fake variables, for the validation of selected features. Arising from these studies, recommendations are given in this thesis on how to conduct reliable ranking of predictive features when large datasets are available.

A new search algorithm for feature subset selection is proposed. This algorithm uses the domain-specific knowledge of the meanings of the terms *strong correlation* and *weak correlation* in order to select the best subset of features for a list of ranked features. It is claimed in this thesis that the proposed method makes better decisions compared to two feature subset selection algorithms proposed in the literature, namely: Correlation-based Feature Selection (Hall, 1999, 2000) and Differential Prioritisation (Ooi et al, 2007).

The implementation of One-versus-All (OVA) aggregate classification models in the presence of large datasets was studied. A new method of determining composition of the training dataset for each base model is proposed. A new method of aggregate model implementation, named positive-Vs-negative (pVn) classification is proposed. An algorithm is proposed for the determination of the classes to be included in each base model. A method of determining the sample composition for the training dataset of each base pVn model is proposed. An algorithm for combining base model predictions and resolving conflicting predictions is proposed.

### 1.6.2 Constructs, models and better theories

Theoretical models are propositions expressing the relationships between the constructs / concepts of the research domain. For feature selection, a model was created to combine the work of various researchers. This model was extended by the author to explain how the definition of feature relevance, the methods used to measure correlations, and the number of dataset samples used, all combine to affect the quality of selected feature subsets. For aggregate model construction, the work of Ho (1998), Freund and Schapire (1997), Ali and Pazzani (1996), Breiman (1996), Kwok and Carter (1990), and Hansen and Salamon (1990), was used as a basis to construct a theoretical model that explains the relationships between the factors that affect aggregate model performance. This model was extended by the author to explain how dataset partitioning methods, learning task complexity, overlap between learning tasks, overlap between training instances, and the quality of the selected features affect the performance of aggregate models. The experimental results were used to demonstrate the relationships between the various factors that affect predictive model performance.

## 1.7 Overview of the thesis

Chapters 2, 3 and 4 provide the background to the research. Chapter 2 provides a discussion of the dataset selection problem for predictive data mining. The chapter provides a background to this problem, giving examples of several application domains where very large datasets are to be found. A review of literature on current methods of selecting training set data from very large datasets for purposes of classifier construction is given. Theoretical methods as well as empirical methods are discussed. The discussion of this chapter also covers single model and aggregate model construction, since the problems of dataset selection and model construction are related. Chapter 3 provides an overview of the feature selection problem for classification tasks in predictive data mining. A review of the available methods for feature selection from small datasets is provided. The weaknesses of these methods are also highlighted. Robust measures of correlation are discussed briefly. In chapter 4, the research questions, the central argument of the thesis, and the research paradigm and research methods, are discussed in detail.

Chapters 5, 6, 7, 8 and 9 provide the details of the empirical studies that were conducted. Further details of the experimental results are provided in the appendices. In chapter 5, the experimental results on feature subset selection are presented. The experimental results demonstrate that the use of many samples results in more reliable feature selection. The results also demonstrate that the use of domain-specific knowledge will lead to better feature subset selection when heuristic subset feature selection is employed. Based on the experimental results of chapter 5 and the existing literature, a theoretical model for the factors that influence the quality of feature selection is proposed in chapter 10.

Chapter 6 provides a discussion of the methods that were used in the experiments for aggregate model design, training dataset design and selection, partitioning and sampling, and base model design and aggregation. The studies to evaluate the performance of the proposed methods are presented in chapters 7, 8 and 9.

Chapter 7 provides a discussion of the empirical study of the use of OVA modeling. It is demonstrated that the use of OVA base models where each base model uses a different training set of the same size as a single model can lead to significant improvements in predictive performance. It is further demonstrated that, by establishing the nature of the instance space and then determining which regions of the instance space to take samples from for each OVA base model, a level of predictive accuracy that is higher than that of a single *k*-class model can be obtained. Based on the experimental results of chapter 7 and the existing literature, a theoretical model for the factors that influence the performance of aggregate models is proposed in chapter 10.

Chapter 8 presents a discussion of the new method of aggregate model implementation called positive-Vs-negative (pVn) classification, as well as the proposed methods for determining the class and sample composition for each pVn base model. Experimental results of the studies to demonstrate the performance of pVn modeling are presented. The experimental results demonstrate that, for the datasets used in the experiments, pVn aggregate modeling provides a high level of predictive accuracy. The experimental results of chapter 8 are used in chapter 10 to enhance the theoretical model for the factors that influence the performance of aggregate models. Chapter 9 provides an in-depth analysis of the OVA and pVn aggregate models operating under different conditions.

Chapter 10 presents the recommendations for dataset selection based on the experimental results for the thesis. Chapters 11 and 12 provide discussions and conclusions for the thesis as well as suggestions for future work. Chapter 11 provides a discussion of the contributions of this thesis to the knowledge base of the field of predictive data mining using aggregate classification models. The discussion of the contributions is presented in terms of the outputs of design science research. Chapter 12 provides conclusions for the thesis as well as suggestions for future work.

# Chapter 2

# Dataset Selection and Modeling from Large Datasets

This chapter provides a discussion of the dataset selection problem for predictive data mining. The discussion provides a background to the dataset selection problem, giving examples of application domains where very large datasets are to be found. A review of the literature on current methods of selecting training set data from very large datasets for purposes of classification modeling is given. Theoretical methods as well as empirical methods are discussed. Since dataset selection and model construction are intimately linked, the discussion in this chapter also addresses single model and aggregate model construction. The strengths and shortcomings of the theoretical and empirical methods are highlighted. The chapter ends with a discussion of research directions that, in the author's opinion, are useful to pursue in order to effectively answer the research question which was presented in chapter 1.

This chapter is organised as follows: Section 2.1 provides motivation for the dataset selection problem with examples of four application domains for data mining. Sections 2.2 and 2.3 respectively introduce the classification modeling problem and dataset selection problem. Sections 2.4 and 2.5 respectively provide a review of theoretically based and empirically based methods for training dataset selection for single model construction. Section 2.6 gives a discussion of existing methods for training dataset selection for multiple model construction. Conceptual views of classification modeling and the sources of classification error are respectively discussed in sections 2.7 and 2.8. The limitations of current training dataset selection methods and the proposed methods of training dataset selection are respectively presented in sections 2.9 and 2.10. Section 2.11 concludes the chapter.

## 2.1 The need for dataset selection

Modern data warehouses store very large volumes of data. In many areas where data mining is applied, very large amounts of data are collected. There are many application areas where data mining from large datasets is applied. These areas

include scientific applications (Fayyad et al, 1996), forensic data mining for purposes of predicting telephone fraud (Hand, 1999), credit card fraud (Chan & Stolfo, 1998), computer network intrusion detection (Lee & Stolfo, 2000), web usage mining for analysing and predicting customer purchases behaviour (Theusinger & Huber, 2000; Kohavi et al, 2004), and customer relationship management (Rygielski et al, 2002; Kohavi et al, 2000; Berry & Linoff, 2000). This section provides examples of application areas where very large datasets for data mining are encountered. Customer Relationship Management (CRM) is discussed in section 2.1.1. Web usage mining and electronic commerce are discussed in section 2.1.2. Forensic data mining is discussed in section 2.1.3. Scientific applications of data mining are discussed in section 2.1.4.

### 2.1.1 Customer Relationship Management - CRM

Customer Relationship Management (CRM) (Giudici, 2003; Rygielski et al, 2002; Bose, 2002; Berry & Linoff, 2000) is a collection of business activities specifically aimed at maintaining good relationships with the business customers. CRM involves the formulation and implementation of strategies to encourage customer loyalty in order for a business to obtain as much value as possible from the customers. Statistically driven CRM (Giudici, 2003) involves the collection, storage and analysis of data about customer interactions with a business in order to obtain a better understanding of customer behaviour. A better understanding of customer behaviour enables businesses to provide better services and product offerings to the customers (Giudici, 2003; Rygielski et al, 2002; Bose, 2002).

Rygielski et al (2002) have argued that, in order for a business to succeed with CRM, the business needs to capture and analyse massive amounts of customer data, analyse the data and transform the analysis results into actionable information. Rygielski et al (2002) have also argued that the analysis of customer data using predictive data mining, especially to extract rules, is an essential component of CRM for the modern business. The use of electronic commerce has made it much easier to collect massive amounts of data about customer purchasing behaviour in data warehouses. The availability of large volumes of data on customer purchasing activities has given rise to research interest in the area of web usage mining for e-commerce. Typical usage of data mining for CRM includes the analysis of customer

attrition, churn, propensity to purchase and customer lifetime value (Giudici, 2003; Rygielski et al, 2002).

### 2.1.2 Web usage mining and electronic commerce

For electronic commerce, since data collection is an automated process, data volumes can grow very rapidly. One interesting application area which has emerged for e-commerce data is clickstream analysis (Kohavi et al, 2004; Theusinger & Huber, 2000). Clickstream analysis is used to study user navigation patterns at a website. The study of user navigation patterns at a website can expose structural or usability problems for a website, which in turn provide useful information for improving the website design. Such a study will also identify which click sequences lead to purchases (Theusinger & Huber, 2000). Kohavi et al (2004) have observed that websites that have 30 million page views per day will need to store in the region of 10 billion records of clickstream data each year. Linden et al (2003) have reported that Amazon.com™ conducts electronic trading with more than 29 million customers per month and stocks several million catalogue items at any given time. The collection of large amounts of web navigation and purchases data creates major challenges for clickstream analysis, for e-traders such as Amazon.com™. Web usage mining applications make explicit the fact that it may be practically impossible to process all of the available data for real-life e-commerce applications of data mining.

### 2.1.3 Forensic data mining

Forensic data mining involves processing large amounts of data in order to identify criminal activities such as credit card fraud (Chan & Stolfo, 1998; Hand, 1999) and computer network intrusion (Lee & Stolfo, 2000). Chan and Stolfo (1998) have reported studies conducted on data for credit card transactions. Chan and Stolfo (1998) have observed that, for the credit card fraud detection domain, there may typically be millions of transactions occurring every day. Hand (1999) has reported that 350 million transactions are recorded annually by UK's largest credit company. Hand (1999) has further discussed the need for real-time data analysis for fraud detection and has argued that, since banking transactions happen all the time, models created, say weeks or months after the fact are useless. There is a need to

constantly create new and up-to-date models. Hand (1999) has further reported that by 1999 AT&T™ was recording 200 million call detail records per day. Phua et al (2005) have reported that descriptive modeling (e.g. cluster analysis), predictive modeling (classification and regression), and pattern detection and rule extraction (e.g. association rules) are all data mining methods that are commonly employed in fraud detection. Scalability of these methods is therefore a serious issue for fraud detection, and dataset selection becomes a necessity.

For modern computer networks large volumes of data are collected and stored in server log files to record all user connections to each server in the network. The users who access the network servers may be authentic users, or may be malicious criminal entities. The data stored in the server log files may be used to create predictive models that are used as network intrusion detection systems (IDS) (Lee & Stolfo, 2000; Stolfo et al, 2000). Lee et al (2000) have observed that the volumes of data stored in server log files are typically huge, as computer networks can experience several million connections on some days due to denial-of-service attacks.

### 2.1.4 Scientific applications of data mining

Fayyad et al (1996) have presented various case studies of the application of data mining to scientific data. Fayyad et al (1996) have observed that the main challenge for the application of data mining to scientific data that is automatically collected by scientific instruments is that these instruments can easily generate terabytes of data at rates as high as several gigabytes per hour. One interesting example is the Palomar Observatory Sky Survey that was conducted over a period of six years (Fayyad et al, 1996). The data collected consisted of 3TB of image data containing 2 billion sky objects. The basic problem here was to create a survey catalogue recording the (predictive) features of each object with its class: star or galaxy. Fayyad et al (1996) have stated that the problem was solved using decision tree learning with multiple trees, and rule extraction with statistical optimisation.

A second interesting example of the application of data mining to scientific data is the analysis of geoscience data for purposes of earthquake detection. Stolorz and Dean (1996) have discussed the Quakefinder system which detects and measures tectonic activity in the earth's crust by examining satellite data. The Quakefinder system

processes massive datasets on a 256-node Cray™ T3D parallel supercomputer to ensure fast turnaround of results for scientists. It is generally not possible for a predictive data mining algorithm to process all of the data for scientific applications where data is automatically collected by measuring instruments. Supercomputers are however used in order to process as much of the data as possible.

## 2.2 Classification modeling from very large datasets

Classification modeling is the process of creating a model which predicts the values of a qualitative variable called the class variable. There are two approaches that have been proposed in the literature for the construction of predictive classification models from very large datasets. The first approach to modeling is concerned with constructing one model using a single sample whose performance is estimated to be as good as that of a model that would be obtained from the whole dataset. The second approach to modeling is concerned with the partitioning of a large dataset into many small subsets which can be efficiently processed, possibly in parallel, creating a base model from each subset of data, and then combining the base models into an aggregate model. The predictive performance of the aggregate model is expected to be at least as good and in several cases superior to that of a single model. Aggregate model construction methods are generally concerned with increasing accuracy compared with the use of a single predictive model. Several methods for aggregate model construction are directly concerned with the parallel processing of the dataset using massively parallel machines in order to ensure that all the data, or as much as of the data as possible, is used in model construction. This section provides a formal definition of the classification problem and the terminology for classification modeling. Methods for single model construction from large datasets as well as the methods for aggregate model construction from small datasets and from very large datasets are discussed. The terminology for classification modelling is presented in section 2.2.1. The classification modeling problem is discussed in section 2.2.2. Single model and aggregate construction are respectively discussed in sections 2.2.3 and 2.2.4. Serial and parallel aggregation, and model testing are respectively discussed in sections 2.2.5 and 2.2.6.

### 2.2.1 Terminology for classification modeling

A dataset for predictive modeling may be described as an *N x (d+1)* data matrix. In the data matrix each row represents *(d+1)* measurements on a real-life object so that the *N* rows in the data matrix represent N real-life objects (Hand et al, 2001). The rows of the data matrix are commonly called *patterns* (Liu & Motoda, 1998), *examples* (Mitchell, 1997), *instances* or *cases* (Hand et al, 2001). The columns of the data matrix are commonly called *variables*, *features* or *attributes* (Hand et al, 2001; Mitchell, 1997). For predictive modeling the first *d* columns are called the *predictor variables* or *features* and the $(d+1)^{st}$ column is called the *predicted variable*. Specific to classification modeling, the *predicted variable* is called the *class variable*. The *d*-dimensional space defined by the *variables* is commonly called the *measurement space* (Hand, 1997) or *instance space* (Mitchell, 1997). Within this *d*-dimensional space, each object (instance) corresponds to one point and the object has an associated class label specified by the $(d+1)^{st}$ column (*class variable*). In this thesis the term *instance* is used to refer to the objects, the term *feature* is used to refer to a predictor variable, the term *class variable* has the usual meaning and, the term *variable* is used to refer to a random variable in the generic sense. The term *instance space* is used to refer to the *d*-dimensional space defined by the predictor variables.

The variables for the data matrix may be quantitative or qualitative (Giudici, 2003; Hand et al, 2001). A *quantitative variable* has numeric values that are either discrete or continuous. The values of a *quantitative discrete variable* have a finite number of levels. The values of a *quantitative continuous variable* come from the domain of real numbers. A *qualitative variable* has values that are either nominal or ordinal. The values of a *qualitative nominal variable* have a finite number of categories which do not possess an ordering. The values of a *qualitative ordinal variable* have a finite number of categories which possess an ordering (Giudici, 2003; Hand et al, 2001). The term *categorical variable* is also used in the literature to refer to a *qualitative variable* (Giudici, 2003; Hand et al, 2001). The terms *quantitative variable*, *quantitative feature*, *qualitative variable,* and *qualitative feature* were adopted for this thesis.

In the literature on machine learning and data mining, various names are used to refer to predictive models for classification. A predictive classification model that is created from a single training sample using a single classification algorithm is called a classifier. When several classifiers are created from one or more training datasets

for purposes of combining them into one predictive model, these classifiers are called *base classifiers* or *base models*. A classifier that is created by combining several *base classifiers* is referred to using various terms. Breiman (1996) has used the term *aggregation* to refer to the process of combining classifier predictions, and the term *aggregate predictor* to refer to the model that results when several classifiers are combined into one model. The terms *ensemble* and *ensemble classifier* have been used by Hansen and Salamon (1990) to refer to combinations of artificial neural networks and are very commonly used in the current machine learning and data mining literature. The term *committee of classifiers,* originating from work on query by committee, has also been used to refer to *ensemble classifiers*. The term *multiple model* is also commonly used (Sun & Li, 2008; Ali & Pazzani, 1996; Kwok & Carter, 1990). In this thesis a decision was made to use the terms *single model, base model,* and *aggregate model.* The term *single model* is used to refer to a classifier created by one algorithm from a single training dataset. The term *aggregate model* is used to refer to a classification model that is created by combining several *base models*. The terms *single model* and *aggregate model* were chosen as it was felt that they capture more precisely, and clearly contrast the structures of the models to which they refer.

In the literature on ensemble classification the terms *complementary classifiers* and *complementarity* are used to refer to base classifiers which make uncorrelated errors (e.g. Martínez-Muñoz et al, 2009). Base model *diversity* is a property that is related to *complementarity*. The term *syntactic diversity* is also used in the literature to refer to base model *diversity* (e.g. Ho, 1998; Ali & Pazzani, 1996; Krogh & Vedelsby, 1995; Kwok & Carter, 1990; Hansen & Salamon, 1990). S*yntactic diversity* refers to the level of structural differences between the base models that constitute an aggregate model. Martínez-Muñoz et al (2009) have observed that base model *diversity* is a necessary but not sufficient condition for *complementarity*. The term *syntactic diversity* is used in this thesis to refer to base model *diversity*. The term *competence* is used in the literature (e.g. Ali & Pazzani, 1996) to refer to the high predictive performance or high predictive expertise of base models. The terms *competence* and *high expertise* are used synonymously in this thesis.

In machine learning literature, the terms *generalisation error* and *generalisation accuracy* are used to refer to the error and accuracy rates of a classifier on data that was not used in training the classifier (Mitchell, 1997). In statistics and data mining literature the terms *prediction error* and *prediction accuracy* are used to refer to the error and accuracy of a predictive model. In this thesis the terms *prediction error* and

*prediction accuracy* were adopted. The term *predictive performance* is used to generally refer to various measures of performance including *prediction error* and *prediction accuracy*. Performance measures for classification models are presented in chapter 4.

The term *bias* appears in machine learning and statistics literature with different meanings. In statistics literature the term *bias* refers to *estimation bias* which is the error in the estimation of a parameter or a model (Mitchell, 1997). In machine learning literature the term *bias* has been adopted with the same meaning as used in statistics (Mitchell, 1997; Geman et al, 1992). In machine learning the terms *inductive bias* and *preference bias* refer to the set of methods used by an inductive algorithm to select a hypothesis (model) from the set of all possible hypotheses (models) in the hypothesis space (model space) (Mitchell, 1997). In this thesis the term *bias* is used with the statistical meaning and the term *inductive bias* is used with the machine learning meaning. The term *search bias* is used to refer to the preferences of a heuristic search procedure.

### 2.2.2 The classification modeling problem

This research is specifically concerned with classification modeling. Classification modeling is the process of creating a model to be used for the prediction of the values of a qualitative variable, given the values of the predictive features. For applied data mining, classification modeling is part of a whole process which involves business understanding, data understanding and preparation, model creation, model assessment and deployment. The Cross-Industry Standard Process for Data Mining (CRISP-DM) is a process model that has been widely adopted for applied data mining (Shearer, 2000). CRISP-DM provides recommendations for the phases to be conducted for data mining projects. Within CRISP-DM the two phases that are directly related to predictive modeling are *data preparation* and *modeling*. For predictive classification modeling, these two phases involve (among others) the following activities: (1) data selection (2) data construction (e.g. creation of the class variable) (3) feature selection (4) model construction (5) estimation of model performance (Shearer, 2000).

It has been illustrated by the examples of the last section that for many application areas, data already exists in large quantities. Data selection is concerned with the

selection of instances and features that have some relevance to the prediction task. Feature selection is concerned with the selection of the most useful features for the prediction task. Classification modeling often requires the construction of a class variable using information derived from other variables based on the objectives of the classification task. Classification modeling involves the estimation of a mapping $m$ (or hypothesis $h$) from an instance $\boldsymbol{x} = (x_1,...,x_d)$ in the *d*-dimensional instance space to the values of the class variable which consists of classes $\{c_1,...,c_k\}$ (Hand et al, 2001). The two conceptual views of classification are discussed later in this chapter.

### 2.2.3 Single model construction

Methods for single model construction from large datasets are motivated by the learning curve. Several researchers have argued that the empirical estimation of training and predictive accuracy achievable from a given large dataset and a given learning algorithm may be done using learning curves (Provost et al, 1999; John & Langley 1996; Catlett 1991). A learning curve shows the relationship between sample size (x axis) and the accuracy of the model (y axis) produced by an inductive algorithm. Learning curves typically have three sections as shown in figure 2.1. The leftmost section has a steep slope, the middle section has a more gentle slope, while the rightmost section is a plateau (Provost et al, 1999; Catlett 1991). These three properties of the learning curve have been used as justification that a single model constructed from a large sample should provide a sufficient level of predictive accuracy (Provost at al, 1999; John & Langley, 1996; Catlett, 1991).

John and Langley (1996), Provost et al (1999) and others have conducted empirical studies and devised methods for establishing the sample size $n_{\min}$ needed to obtain maximum accuracy for a given dataset and algorithm. Extrapolation of learning curves (ELC) is one method that has been used to fit learning curves (Frey & Fisher 1999). For ELC, training sets of increasing size are used to fit a parametric learning curve, which is an estimate of the algorithm's accuracy as a function of training set size.

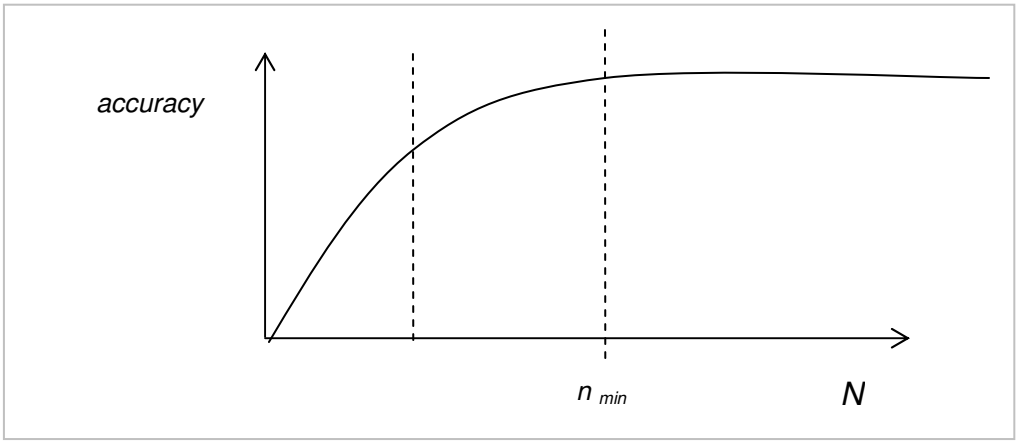


*Figure 2.1 A typical learning curve*

### 2.2.4 Aggregate model construction

The idea of using an aggregate model originates from the work of Breiman (1996) on bagging predictors. Breiman (1996) has demonstrated that, by creating classifiers from many bootstrap samples of a small dataset, prediction performance may be greatly improved. Bootstrap samples are created by using sampling with replacement in order to create many training datasets each with the same size as the original dataset. Hand et al (2001) have observed that model aggregation has conceptual similarities with Bayesian model-averaging. For Bayesian model-averaging all models in the model space are used in order to maximise predictive accuracy. The vote of each model is weighted by the posterior probability of that model, given the training data (Domingos, 2000b; Ali & Pazzani, 1996). Since the generation of all models is intractable, all implementations of aggregate modeling have to be approximations, and bagging predictors are an example of such an approximation (Ali & Pazzani, 1996).

Chawla et al (2001) have proposed a method of improving classifier accuracy by partitioning a large dataset, constructing a base model with all the data from each partition, and combining the base models into an aggregate model. Chawla et al (2001) have concluded that such a strategy leads to a higher level of predictive performance compared to the use of a single model constructed from the whole dataset. Chawla et al (2001) have argued that bagging is not suitable for very large datasets. In their experiments with various ways of partitioning a dataset, Chawla et al (2001) have concluded that disjoint partitioning results in the best performance. It should be highlighted that Chawla et al (2001) used a supercomputer with a

massively parallel architecture and it took ten hours to create an aggregate model for a 3.6 million record dataset with 304 features.

Hall et al (2000) have conducted experiments that are fairly similar to those of Chawla et al (2001), using the same architecture as that used by Chawla et al (2001). The main difference in the studies is that Hall et al (2000) have used four very large datasets (1.6, 3.2, 6.4 and 51 million instances) in their experiments compared to Chawla et al (2001) who have used one very large dataset (3.6 million instances). Hall et al (2000) have observed that for different datasets, different amounts of partitioning provide different levels of accuracy. Hall et al (2000) have reported that accuracy will actually decrease when partitioning is applied to very large datasets where very small classes are present in the data. Partitioning of such datasets causes the very small classes to appear as noise. The main conclusion made by Hall et al (2000) is that the use of disjoint partitions of a very large dataset may result in a model with the same accuracy as that obtained without any partitioning. Hall et al (2000) have further concluded that the use of overlapping subsets, in a manner similar to bagging, may provide an increased level of accuracy.

Ali and Pazzani (1996) have studied the use of aggregate models on data originating from many different domains. The objective of Ali and Pazzani's (1996) study has been to explain why there is a significant variation in prediction error reduction from domain to domain when aggregate models are used. Ali and Pazzani (1996) have tested twenty nine (29) datasets and found that aggregate models provide significant prediction error reduction on only half of these datasets. Ali and Pazzani (1996) have made four main conclusions from their study. The first conclusion is that aggregate models are better at reducing prediction error on domains for which the prediction error is already very low, than on domains that have noisy data. The second conclusion is that aggregate models improve prediction performance in those domains with many irrelevant features. The third conclusion is that as the number of irrelevant features increases, the performance of aggregate models decreases. The fourth conclusion is that when the prediction errors made by the base models are strongly correlated, the aggregate model does not provide any prediction performance improvements.

Several authors (e.g. Ho, 1998; Krogh & Vedelsby, 1995; Kwok & Carter, 1990; Hansen & Salamon, 1990) have argued that when aggregate models exhibit syntactic diversity, then major improvements in prediction performance should be

realised. On the other hand, Ali and Pazzani (1996) have argued that the accurate models that can be learned for several domains are syntactically similar, so that increasing syntactic diversity does not result in improvements. Ali and Pazzani (1996) have further argued that in order to minimize aggregate model prediction error, it is necessary to balance increased diversity with competence, that is, ensure that the base models are all competent, and have a very high level of training accuracy.

Ho (1998) has discussed the use of decision forests for the improvement of decision tree accuracy. For a decision forest, an aggregate model is constructed through random sampling of the feature space. Each classification tree that is constructed is capable of (an expert in) classification of instances that reside in the instance space defined by that subset of features which has been randomly selected. The combined performance of the decision forest is then higher than that of a single decision tree that is created to predict in the instance space defined by all the features of the dataset. The experiments conducted by Ho (1998) on feature space partitioning have been based on small datasets. Ho's (1998) method however shows promise for a divide-and-conquer approach for very large datasets of high dimensionality. The method demonstrates that syntactic diversity can be achieved through variation of the feature space for each base model.

Chan and Stolfo (1998) have proposed a method of aggregate model construction that addresses the problem of handling large two-class datasets with skewed class distributions. Chan and Stolfo (1998) have compared their method to that of using a single model and have concluded that their method provides superior performance. A more detailed discussion of Chan and Stolfo's (1998) method is given in section 2.6 where the methods that combine dataset sampling and partitioning are discussed.

Boosting (Freund & Schapire, 1997) is a method of aggregate model construction which combines training set selection with aggregate model creation. For boosting, a sequence of base models is created, with each base model in the sequence having a higher level of competence at the classification of 'difficult' instances. In this context a 'difficult' training instance is one that cannot be classified correctly by all preceding base models in the sequence. A more detailed discussion of boosting is provided in section 2.6 of this chapter.

### 2.2.5 Serial and parallel model aggregation

In general all aggregate models consist of two components. The first component is the set of base models. The second component is the combination algorithm. A combination algorithm may perform *parallel combination* or *serial combination* of the predictions of the base models. The methods for aggregate model construction which were discussed in the last section employ a *parallel combination* algorithm. The method of *parallel combination* consists of two steps. In the first step, all the base models make their individual predictions. In the second step, the combination algorithm selects that prediction with the strongest supporting evidence. Kittler (1998) has observed that base model combination methods for parallel aggregation fall into two categories. The first category involves discrete classification (Fawcett, 2004, 2006) where only the class labels for the classes predicted by the base models are available. For this category, a voting scheme based on the majority rule (Breiman, 1996; Hansen & Salamon, 1990) is appropriate for the combination of base model predictions. The majority rule is implemented by selecting that class which is predicted by the majority of base models.

The second category involves probabilistic classification (Fawcett, 2004, 2006) where probabilistic scores for each class are provided by the base models. Given the base models $M_1,..,M_A$, and the classes $c_1,..,c_k$, let the probabilistic scores assigned to a query instance $\mathbf{x}_q$ by models $M_1,..,M_A$ for classes $c_1,..,c_k$, be denoted by the values $CONF(c_i,M_j)$, $i=1,..,k, j=1,..,A$. Kittler (1998) has discussed four different rules that can be used to combine the $CONF(c_i,M_j)$ scores in order to select the winning class. The *product rule* involves the multiplication of the scores for each class to obtain the combined class score $conf_i$ for each class, where $conf_i$ is defined as (Kittler, 1998)

$$conf_i = \prod_{j=1}^{A} CONF(c_i, M_j) \tag{2.1}$$

and selecting the class with the largest value of $conf_i$ defined as (Kittler, 1998)

$$conf_i^* = max\{conf_1,..,conf_k\} \tag{2.2}$$

The *sum rule* involves the summation of the scores for each class to obtain the combined class score $conf_i$ defined as (Kittler, 1998)

$$conf_i = \sum_{j=1}^{A} CONF\ (c_i, M_j) \tag{2.3}$$

and selecting the class with the largest value of $conf_i$ as defined by equation (2.2). The *max rule* involves the selection of the class with the largest score defined as (Kittler, 1998)

$$conf_i = \max_{j=1}^{A} CONF\ (c_i, M_j) \tag{2.4}$$

and selecting the class with the largest value of $conf_i$ as defined by equation (2.2). The *min rule* involves the selection of the class with the smallest score defined as (Kittler, 1998)

$$conf_i = \min_{j=1}^{A} CONF\ (c_i, M_j) \tag{2.5}$$

and selecting the class with the largest value of $conf_i$ as defined by equation (2.2). Ho (1998), and Kwok and Carter (1990) have implemented the sum rule for decision tree base models by computing the arithmetic mean of the scores for each class and selecting that class with the largest arithmetic mean score. Berry and Linoff (2000: pg 217) have provided an illustrative example of how the product rule may be implemented.

More recently, a second method of base model combination called *serial combination* has been proposed (Sun & Li, 2008; Neagu, 2006; Kim et al, 2002). *Serial combination* is a multi-step process. In the first step the base models are arranged in a series. In order to classify a new instance, the instance is passed to the first base model in the series. If the base model makes a *'credible prediction',* then the process stops otherwise the instance is passed to the next base model in the series. In general, if a base model makes a 'credible prediction' the process stops otherwise the instance is passed to the next base model in the series (Sun & Li, 2008).The meaning of a *'credible prediction'* may be defined and implemented in a variety of ways. Sun & Li (2008) have used the following definition and implementation. For each class, the base model that has the highest predictive accuracy on that class is

identified. When a base model predicts a class that it is best at predicting, the base model has made a '*credible prediction'*, otherwise the prediction is considered to be '*not credible'*. Sun and Li (2008) have demonstrated that their method of serial combination produces an aggregate model whose performance on each class is as good as the performance of the best base model on the class.

For the research reported in this thesis, the method of *parallel combination* was studied. In chapter 10, a comparison is made between the advantages of serial combination and the advantages of the methods proposed in this thesis.

### 2.2.6 Model testing

Traditionally, the three methods of model testing in machine learning and statistical pattern recognition are, the hold-out method, *K*-fold cross validation, and the bootstrap method (Mitchell, 1997; Moore & Lee, 1994). These methods of model testing were designed for model construction from small datasets, and primarily address the problem of data shortage. For the hold-out method the available data is split into a training set and a test set (hold-out set). The test set is used to estimate the predictive accuracy. The test set may be ½, ⅓, or ¼ of the available data. For, *K-fold* cross validation, the available dataset consisting of *n* instances is divided into *K* subsets of equal size. For each of the *K* subsets, the remaining *K-1* subsets are combined into the training set, and the remaining subset is used to estimate the error. For $K << n$, the entire process is typically iterated many times (e.g. 100) and the results are averaged. When $K = n$, the leave-one-out (LOO) method is obtained. For the bootstrapping method, a training set of size *n* is chosen randomly with replacement, which means that each item may appear more than once in the training set. Only those items that do not appear in the training set are used for the test set and only once each. This process is iterated many times (e.g. 200) and the error rates are averaged (Moore & Lee, 1994).

Testing models in the presence of large volumes of data continues to be done using either K-fold cross validation or the hold-out method. *K*-fold cross validation is used to establish the accuracy on the training data. When only one model is being considered, the hold-out method is used to create two datasets, one for training and one for measuring the predictive accuracy of the final model. When several models are constructed with the objective of selecting the best one, the hold-out method is

used to create three datasets, one for training, one for validation, and one for measuring the predictive accuracy of the final model that is selected. The validation dataset is used to determine which of the many models has the best predictive performance. Given the stochastic behaviour of predictive models, several samples taken from the validation and test datasets are used for both the validation and testing steps and the results are averaged. Several researchers have argued that predictive accuracy should not be the only measure of model performance (Osei-Bryson, 2004, 2007; Giudici, 2003; Hand, 1997). Various measures of classification model performance are discussed in chapter 4.

## 2.3 The dataset selection problem

It was stated in section 2.2.2 that data preparation is one of the phases of the CRISP-DM model for applied data mining (Shearer, 2000). Within CRISP-DM, data preparation involves three steps namely data selection, data construction, and feature selection, among others. Data selection is concerned with the identification and selection of sufficient quantities of good quality data that is relevant to the data mining goals (Shearer, 2000). Data records (instances) as well as relevant attributes (features) are identified and selected during this step. Data construction is concerned with the creation of any necessary new features, for example, the class variable for classification (Shearer, 2000).

The data selected during the data preparation phase as prescribed in the CRISP-DM model is commonly pre-processed further when the modeling task is to create predictive models. Firstly, it is important to select training data so that overfitting of predictive models is avoided (Smyth, 2001; Dietterich, 1995). This is accomplished through data reduction. Hand et al (2001) have advised that one approach to reducing the amount of training data when the objective of data mining is to create models, is through sampling from the very large dataset. A second approach that is suggested by Hand et al (2001) is the use of sufficient statistics. Hand et al (2001) have provided least squares regression as an example of modeling where the use of sufficient statistics is enough to estimate the regression coefficients. For least squares regression, the sufficient statistics are the sum for each variable, sum of squared values for each variable, and sum of products for the values of the regression variables. Note that regression models are predictive. For classification, there are algorithms for which the usage of sufficient statistics seems feasible. The

Naïve Bayes classifier (Mitchell, 1997) is characterised by two types of probabilities: the probability of the class and the probability of a variable value given the class. For the creation of a Naïve Bayes classifier, the data records could be replaced by the probability values.

Secondly, pre-processing may be done to make the data suitable for a classification algorithm. For example, artificial neural networks (Engelbrecht, 2002; Bishop, 1995) require normalised data, and K-nearest neighbour algorithms (Cover & Hart, 1995) perform best with normalised data. Thirdly, pre-processing may also be done to increase the likelihood that the classification algorithm will produce a classification model with high predictive performance. This third type of pre-processing involves selecting the most relevant training data for the classification task (e.g. Blum & Langley, 1997), or altering the probability distribution of the training data when data has a skewed class distribution (e.g. Chan & Stolfo, 1998; Kubat & Matwin, 1997). Fourthly, pre-processing is done to further select the most relevant features for the prediction task.

The dataset selection problem addressed in this thesis was concerned with the selection of relevant features and relevant training data for the construction of many base models that make up an aggregate model. The use of aggregate models was studied for purposes of increasing the amount of (relevant) training data while at the same time avoiding the problem of overfitting. Training dataset selection was directed at classification algorithms for which data appears in raw form (at the instance level) to the algorithm. The next two sections provide a discussion of dataset selection methods that have been found appropriate for the modeling methods discussed in the last section, and for which training data must be presented to the algorithm at the instance level as opposed to a summarised (aggregated) level. Feature selection methods are discussed in chapter 3.

## 2.4 Theoretical methods for single sample selection

Predictive data mining has its roots in the fields of machine learning and statistical pattern recognition. The purpose of this section is to discuss the theories of machine learning and statistical pattern recognition which have been proposed for purposes of characterising the behaviour of algorithms that create predictive classification models through a process of induction from supplied example data. These theories may be

used to estimate a sufficient sample size, or sample complexity, for achieving a given level of accuracy for a single predictive classification model. The important lessons to be learned from the theories on sample complexity, as well as the weaknesses of these theories, are highlighted in this section. The probably approximately correct learning theory is presented in section 2.4.1. The theory on the Hoeffding-Chernoff bounds is discussed in section 2.4.2.

### 2.4.1 Probably Approximately Correct (PAC) learning

The probably approximately correct (PAC) theoretical model of learning proposed by Valiant (1984) and discussed by Mitchell (1997) has been designed for purposes of characterising algorithms that learn target concepts by generating a hypothesis *h* from a set *H* of all possible hypotheses that belong to some concept class. The learning algorithms use training instances drawn at random according to some unknown, but fixed, probability distribution. PAC is concerned with the identification of classes of hypotheses that can and cannot be learned from a polynomial number of instances. Within the PAC theory various measures of hypothesis space complexity have been proposed for purposes of establishing bounds for the number of training instances required for achieving a given level of accuracy for inductive learning algorithms. Within the PAC framework, a learning algorithm that finds the hypothesis $h \in H$ with the minimum training error is called an *agnostic* (or robust) learner. For a hypothesis space *H*, it is guaranteed with probability $(1-\delta)$, that an agnostic learner will output a hypothesis $h \in H$, which has a prediction error rate of at most $\varepsilon$. This guarantee will hold provided that *n*, the size of the training sample used to generate *h*, conforms to (Mitchell, 1997)

$$n \geq \frac{1}{2\varepsilon^2}\left(\ln\frac{1}{\delta} + \ln \mid H \mid\right) \tag{2.6}$$

Equation (2.6) is applicable to classes of hypotheses for which /*H*/, the size of the hypothesis space, is finite. One major problem with the sample complexity estimates based on equation (2.6) is that the size of the hypothesis space is not always easy to estimate. As an example, for decision trees the hypothesis space is the set of all possible decision trees that can be created from the given dataset. A second problem is that the instances in the training sample are assumed to be independent and

identically distributed, a requirement that is extremely difficult to satisfy. A third problem is that the hypothesis space may be infinite in size. For infinite hypothesis spaces, a useful measure of the complexity of *H* is its Vapnik-Chervonenkis dimension, *VC(H)* (Vapnik & Chervonenkis, 1971). *VC(H)* is the size of the largest subset of instances that can be shattered (split in all possible ways) by *H.* An alternative upper bound for the sample complexity *n,* under the PAC model is given by (Mitchell, 1997)

$$n \geq \frac{1}{\varepsilon}\left(4\log_2(2/\delta) + 8VC(H)\log_2(13/\varepsilon)\right) \tag{2.7}$$

One major problem with the sample complexity estimates based on equation (2.7) is that it is not always easy to estimate the *VC* dimension for a given classification algorithm. Additionally, the *VC* dimension might be infinite, as is the case for a fully grown decision tree. In artificial neural network learning, however, the application of the *VC* dimension has been used successfully. A general criticism of the use of equations (2.6) and (2.7) is that they provide a training sample size estimation which is usually excessively large.

### 2.4.2 The Hoeffding-Chernoff bounds

The Hoeffding-Chernoff theorems (Hoeffding, 1963) have been proposed by several researchers (e.g. Watanabe, 2005; Domingo et al, 2002; Kiniven & Manila, 1993) as an alternative method for training sample size estimation. Kiniven and Manila (1993) have discussed the use of concentration bounds (Hoeffding-Chernoff bounds) for determining sufficient sample sizes for a specified level of accuracy, when determining the truth of universal sentences expressed as first order logic formulae. Toivonen (1996) has discussed the use of these bounds for sample size estimation in association rule mining. The major criticism of the usage of the Hoeffding-Chernoff bounds is similar to that of PAC estimates. The sample sizes they estimate are usually excessively large.

Watanabe (2005) and Domingo et al (2002) have proposed an adaptive sampling scheme, which incorporates the use of the sample size bounds stated by the Hoeffding-Chernoff theorems. Watanabe (2005) and Domingo et al (2002) have argued that the methods they have proposed preserve the theoretical guarantees

(level of accuracy and confidence in the level of accuracy) of the theorems while at the same time providing good and practical estimates of sample sizes.

## 2.5 Empirical methods for single sample selection

For the empirical estimation of a sufficient training sample size, three approaches have been reported in the literature. A sufficient training sample size is one which provides a level of predictive accuracy that is comparable to processing the whole dataset. The first approach to empirical sample size estimation involves taking progressively larger samples from a large dataset until the sufficient sample size has been reached (Provost et al 1999; John & Langley 1996). The second approach is based on the assumption that a sample that has statistical similarity to the whole dataset is a sufficient sample. Statistical similarity is measured in terms of the descriptive statistics for the dataset variables (Lutu & Engelbrecht 2006; John & Langley 1996). The third approach to the empirical estimation of sufficient sample sizes is to select samples based on the characteristics of the instance space (Palmer & Faloutsos, 2000; Kubat & Matwin, 1997). The three approaches are discussed in this section. Dynamic sampling and progressive sampling methods are discussed in section 2.5.1 and 2.5.2 respectively. Static sample size estimation is presented in section 2.5.3. Density-biased sampling and one-sided sampling are respectively discussed in sections 2.5.4 and 2.5.5.

### 2.5.1 The Dynamic Sampling method

John and Langley (1996) have proposed a method they call dynamic sampling, which combines database sampling with the estimation of classifier accuracy. The method is most efficiently applied to classification algorithms which are incremental, for example, Naïve Bayes and artificial neural network algorithms such as backpropagation. John and Langley (1996) have defined the concept of *'probably close enough' (PCE),* which they use for determining when a training sample size provides an accuracy that is probably good enough. *'Good enough'* in this context means that there is a small probability $\delta$ that the classification algorithm could do better by using the entire dataset. The smallest sample size *n*, is chosen from a dataset of size *N*, so that $P_r(accuracy(N) - accuracy(n) > \varepsilon) \leq \delta$, where

*accuracy(n)* is the accuracy after processing a sample of size *n*, and ε is a parameter that describes what '*close enough'* means.

Dynamic sampling works by gradually increasing the sample size *n* until the PCE condition is satisfied. *accuracy(n)* is estimated by taking a new sample from the database, classifying all instances in the sample and measuring the accuracy. *accuracy(N)* is estimated using the method of Extrapolation of Learning Curves (ELC). In their study, John and Langley (1996) have compared the accuracy of static and dynamic sampling for the Naïve Bayes classifier, and have concluded that the use of dynamic sampling results in the selection of a single sample which provides a level of accuracy that is very close to that obtained when the whole large dataset is used for classifier construction.

## 2.5.2 The progressive sampling method

Provost et al (1999) have proposed progressive sampling as an alternative method for the empirical estimation of sufficient training sample sizes. Provost et al (1999) have addressed the issue of convergence, where convergence means that a learning algorithm has reached its plateau of accuracy. In order to detect convergence, Provost et al (1999) have defined the notion of a sampling schedule as a sequence $\{n_0, n_{1,}..n_i\}$ of sample sizes to be provided to an inductive algorithm. Provost et al (1999) have argued that schedules where the sample size $n_i$ increases geometrically as $\{n_0, a.n_0, a^2 n_0, ..., a^i n_0\}$ are asymptotically optimal. Progressive sampling is similar to the adaptive sampling method of John and Langley (1996), except that a non-linear increment for the sample size is used. Provost et al (1999) have handled the problem of convergence detection by using a method called *Linear Regression with Local Sampling (LRLS).* *LRLS* fits a linear regression line in the neighbourhood of $n_i$, the size of the last training sample obtained. If the slope of the line is sufficiently close to zero, then convergence is detected. Provost et al (1999) have reported experimental results which show that geometric progressive sampling far outperforms dynamic sampling.

### 2.5.3 Static sample size estimation

John and Langley (1996) and Provost et al (1999) have made a distinction between static and dynamic sampling for data mining. For static sampling, $n_{min}$, the smallest sample size needed to achieve maximum accuracy, is determined on the basis of a sample's statistical similarity to the whole large dataset. Statistical similarity is measured in terms of the descriptive statistics for the dataset variables. Lutu and Engelbrecht (2006) have studied the selection of samples based on statistical validity and concluded that statistical validity is not a sufficient test for dataset selection. Lutu and Engelbrecht (2006) have concluded that there is a statistically significant performance difference between small statistically valid samples and large statistically valid samples. One important difference they have identified is information content as measured using the entropy function.

### 2.5.4 Density-biased sampling

Palmer and Faloutsos (2000) have proposed density biased sampling as a suitable method for sampling from large datasets in which clusters of differing sizes occur. Palmer and Faloutsos (2000) have argued that for such datasets, uniform sampling fails to represent small clusters (small groups) of interesting instances in the instance space. For density biased sampling, the aim is to sample so that within each cluster, instances are selected uniformly to obtain a training sample that is density preserving and biased by cluster size. Density preserving in this context means that the expected sum of weights of the sampled instances for each cluster is proportional to the cluster's size. The method of density-biased sampling is used to select instances to be included in the dataset based on the density of the various regions of the instance space. The purpose is to ensure that all regions of the instance space are equally represented in the selected dataset.

### 2.5.5 One-sided sampling

One-sided sampling is a training sample selection method that has been proposed by Kubat and Matwin (1997) for the selection of training instances based on the class distributions in the different regions of the instance space. Kubat and Matwin (1997) have argued that one-sided sampling is suitable for datasets with skewed class

distributions. For datasets with skewed class distribution, Kubat and Matwin (1997) have argued that the training datasets should be selected based on where the decision boundaries of the classes lie in the instance space. For 2-class problems with positive and negative instances for the concept to be learned, Kubat and Matwin (1997) have identified four types of negative instances as follows:

(1) Noisy instances. These are instances that incorrectly have the negative class label.

(2) Borderline instances. These are instances that are located very close to the decision boundary between the positive and negative class.

(3) Redundant instances. These are instances that lie far away from any decision boundary.

(4) Safe instances. All the instances that do not fall into any of the above categories are safe instances.

For the one-sided sampling method, instances that fall in categories (1), (2) and (3) above are removed from the training dataset. The rationale for one-sided sampling is that when one-sided samples are used for training, then the regions of class confusion are removed from the training data. Therefore classifiers based on discriminative classification should not experience any class confusion. Kubat and Matwin (1997) have demonstrated that this scheme produces good training datasets for the k-Nearest Neighbour and decision tree classifiers. One obvious problem with one-sided sampling is that when borderline negative instances (category 2) are removed from the training dataset, the resulting predictive model has limited information to predict instances that are located in the borderline regions. However, the results of the studies conducted by Kubat and Matwin (1997) may be used to argue that purposeful dataset selection, based on the characteristics of the instance space, may lead to the selection of training datasets that result in a higher level of predictive accuracy compared to training datasets obtained through pure random sampling.

## 2.6 Methods for selecting multiple training datasets

The construction of aggregate models requires the use of several training datasets. Each training dataset is used to construct one base model, and the base models are then combined into one aggregate model. For small datasets, methods such as bootstrapping and boosting have been devised for purposes of increasing the

number of training instances available for base model creation. Breiman (1996) has investigated the use of bootstrap sampling of small datasets in order to create the training datasets for the base models. Traditionally, boosting has been used in statistical modeling to improve model performance. Boosting involves the use of several variations of one training dataset to create several base models (Giudici, 2003; Freund & Schapire, 1997). For large datasets, partitioning and sampling have been used to create training datasets for base models. Chawla et al (2001) have investigated the partitioning of a large dataset in order to create several training datasets for the base models. Chan and Stolfo (1998) have investigated combining dataset partitioning with sampling in order to create the base models. In this section, the methods proposed in the literature, for obtaining multiple training datasets (samples) for aggregate model construction are discussed. The methods for bootstrap sampling and boosting of small datasets are presented in section 2.6.1. Partitioning of large datasets and the methods for combining partitioning and sampling from large datasets are respectively discussed in sections 2.6.2 and 2.6.3.

### 2.6.1 Bootstrap sampling and boosting of small datasets

For small datasets, Breiman (1996) has proposed the use of bootstrap sampling (Cohen, 1995) in order to create the required number of training datasets. Bootstrap samples are created by using sampling with replacement in order to create many training datasets each with the same size as the original dataset. Breiman (1996) has recommended that at least 30 training datasets should be generated and used to create the base models of an aggregate model when bootstrap sampling is applied to a small dataset.

Boosting is a statistical approach to model construction which aims to direct the largest effort of model construction towards the more difficult aspects of the process to be modeled. Giudici (2003) has observed that early versions of boosting fitted models on several versions of the training dataset, where the observations with the poorest fit received the largest weight. For classification modeling, Adaboost (Schapire, 2003; Freund & Schapire, 1997) is a boosting algorithm which creates many base classifiers that are finally combined into one prediction model. At each iteration of Adaboost, the training instances that are misclassified by the most recently created base classifier are assigned larger weights in the training set for the next base classifier. For classification, this means that the instances are replicated in

the next training set in proportion to the assigned weights. The rationale behind training dataset selection by Adaboost is to increase the representation of the instances that come from those regions of the instance space that are very difficult to model and predict.

The method of bootstrap sampling is commonly used in statistics to create larger datasets that have the same statistical properties as the small dataset from which the bootstrap sample is obtained (Cohen, 1995). In the context of aggregate model creation, bootstrap sampling provides a large amount of data for purposes of creating the base models. When large amounts of data are available, bootstrap sampling obviously becomes unnecessary. Boosting, as implemented in Adaboost, aims to increase coverage of the difficult regions of the instance space when there is a shortage of data, as is the case for small datasets. The studies reported in this thesis demonstrate that, first of all, the use of aggregate models as is done in bootstrap aggregation also provides performance improvements over single models when large amounts of data are available. Secondly, when large amounts of data are available, it is possible to increase coverage of the difficult regions of the instance space without using the methods of Adaboost, and without using all of the available data.

### 2.6.2 Partitioning of large datasets

For very large datasets, the training datasets are typically obtained by dividing the large dataset into several partitions. The most common approach to dataset partitioning for data mining is to use horizontal partitioning. For horizontal partitioning, a criterion is applied to assign each instance of the dataset to one of $P$ partitions. The partitioning criteria that have been studied include disjoint partitioning and overlapped partitioning. For disjoint partitioning every instance in the dataset (of size $N$) appears in exactly one partition (Chawla et al, 2001; Hall et al, 2000). The original dataset is divided into $PT$ partitions each of size $(N/PT)$ so that each instance appears in exactly one partition (Chawla et al, 2001).

For overlapped partitioning an instance may appear in more than one partition (Chawla et al, 2001; Hall et al, 2000; Breiman, 1996). Each partition is created independently of the others using either random sampling with replacement or random sampling without replacement. Randomly selected instances are added to the partition until the partition is of size $(N/PT)$. If sampling is done with replacement

some replication of instances within each partition and across the *PT* partitions will occur. If sampling is done without replacement, replication of instances within partitions does not occur (Chawla et al, 2001).

### 2.6.3 Combining dataset sampling and partitioning

Chan and Stolfo (1998) have reported experiments conducted on data for credit card transactions for purposes of identifying fraudulent transactions. Data for credit card transactions typically has a skewed class distribution with the fraudulent transactions having a representation in the range of 1% to 5% of the whole dataset (Chan & Stolfo 1998). In their studies, Chan and Stolfo (1998) have addressed the problem of creating training datasets with balanced class distributions and then creating base models from each training dataset. In order to create the training datasets, they have proceeded as follows. First, the whole dataset is partitioned according to the two classes *{normal, fraudulent}* to create two partitions *NORMAL* and *FRAUDULENT*. Since *fraudulent* is the minority class and the objective of partitioning is to balance the class distributions, the *NORMAL* partition is further divided into smaller partitions $NORMAL_1,...,NORMAL_J$. The training datasets for the base classifiers are then constructed by combining each of the small partitions $NORMAL_1...,NORMAL_J$ with the partition *FRAUDULENT.* In other words, each of the training datasets has all the minority class instances and *(1/J)*$^{th}$ of the majority class instances. Chan & Stolfo (1998) have concluded that compared to simple random sampling, this method of constructing training datasets results in better predictive performance for datasets with skewed class distributions.

## 2.7 Conceptual views of classification modeling

There are two well accepted (conceptual) views of classification, namely: discriminative classification and probabilistic classification (Hand et al, 2001). It is important to briefly discuss these views of classification modeling in order to establish the extent to which methods of data selection from large datasets attempt, or should attempt to satisfy the objectives of these views. Sections 2.7.1 and 2.7.2 respectively provide a discussion of discriminative and probabilistic classification modeling. A

concise definition of decision boundaries for classification that was adopted for the experiments of this thesis is given in section 2.7.3. Section 2.7.4 provides a discussion of training dataset selection methods aimed at supporting the objectives of classification modeling.

## 2.7.1 Discriminative classification

For discriminative classification (Hand et al, 2001), a classification model provides a mapping, $m$, from an instance $\boldsymbol{x} = (x_1,...,x_d)$ in the *d*-dimensional instance space to a set of classes $\{c_1,...,c_k\}$. The *d*-dimensional instance space is viewed as consisting of regions with labels for each of the *k* classes. The mapping, $m$, defines the various regions of the instance space. For each class $c_i$, the union of all the regions with that class label is called the *decision region* for the class. The mapping may also be interpreted as a definition of the *decision boundaries* between the *decision regions.* For real life classification problems, the classes are usually not perfectly separable in the *d*-dimensional instance space so that there are regions of class confusion for the mapping, $m$. Discriminative models handle the problem of class confusion by assigning a probability for each class to each decision region in the instance space. In the process of classification, a new instance ***x*** is assigned to the most probable class for the region in which it falls. The classification modeling problem may therefore be defined as a process of estimating the *decision boundaries* as closely as possible, with the objective of minimizing class confusion in each decision region. Examples of classifiers that follow this approach are decision trees for classification (Quinlan, 1993; Quinlan, 1986; Breiman et al, 1984), artificial neural networks (Engelbrecht, 2002; Bishop 1995), and K Nearest Neighbour (Cover & Hart 1967).

## 2.7.2 Probabilistic classification

*Probabilistic models* for classification are based on the assumption that, for all instances $\boldsymbol{x} = (x_1,...,x_d)$ belonging to class $c_k$, there is a probability distribution or density function governing the characteristics of the class $c_k$. For example, the probability distribution functions for a multivariate dataset with quantitative features might be multivariate normal with estimated means and variances for the features

(Hand et al, 2001). If the means associated with the different classes are far enough apart and the variances are small, then the classes will be well separated. In practice, the appropriate functional forms for describing the probability distributions for the classes are not known. However, it is possible to estimate from the data the prior probabilities $p_r(c_i)$ for each class, and the posterior probabilities $P_r(c_i \mid (x_1,...,x_d))$ of instance ***x*** **=** $(x_1,...,x_d)$ belonging to class $c_i$. The posterior probabilities $P_r(c_i \mid (x_1,...,x_d))$ can be viewed as carving the instance space into at least *k* decision regions and at the same time defining the decision boundaries for the classes. An examples of a modeling method based on probabilistic classification is the Naïve Bayes classifier.

One distinguishing characteristic between *discriminative classification modeling* and *probabilistic classification modeling* is that *probabilistic models* are created by computing the prior and posterior probabilities that determine whether an instance belongs to a given class. On the other hand, for *discriminative modeling* probabilities are used when the most likely class must be assigned to an instance **x**. For *probabilistic classification*, the training datasets should have the same probability distributions as the parent dataset, but for *discriminative classification* this limitation does not hold.

### 2.7.3 Definition of decision boundaries and class confusion regions

One of the training dataset selection methods proposed in this thesis is based on the identification of decision boundaries for classification and those regions where predictive models confuse one class for another class (confusion regions). From a probabilistic view of classification modeling, Hand et al (2001) have defined a decision boundary between two classes $c_i$ and $c_j$ as a 'contour' or 'surface' in the instance space which has

$$p_r(c_i, \mathbf{x}) = P_r(c_j, \mathbf{x}) = 0.5 \qquad (2.8)$$

where $P_r(c_i, \mathbf{x})$ is the prior probability that instance ***x*** has the class label $c_i$ and $P_r(c_j, \mathbf{x})$ is the prior probability that instance ***x*** has the class label $c_j$. The 'contour' defined by equation (2.8) is depicted in figure 2.2 as the bold curve.

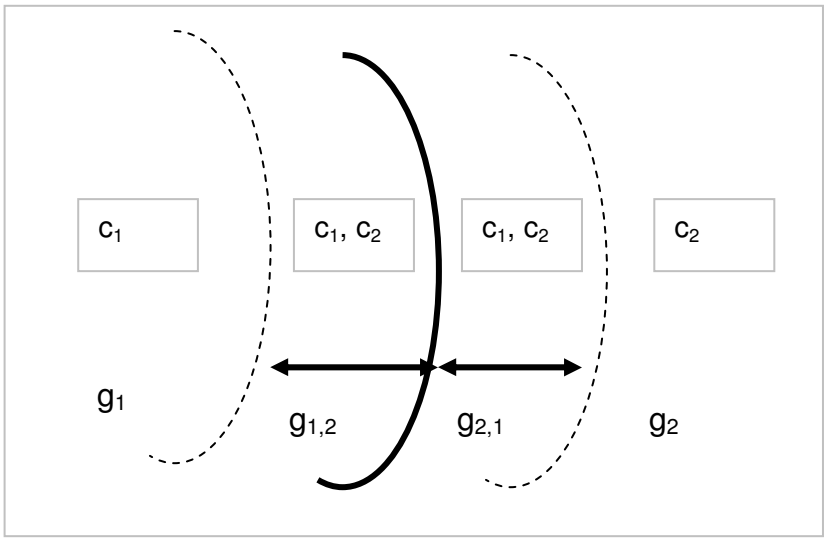


*Figure 2.2: Confusion region for two classes*

Based on Hand et al's (2001) definition of a decision boundary, the confusion region for classes $c_i$ and $c_j$ was formulated by the author as follows: On either side of the decision boundary there are two regions $g_{1,2}$ and $g_{2,1}$ where the two classes $c_i$ and $c_j$ occur together as depicted in figure 2.2. The region $g_{1,2}$ is characterised by three inequalities: $P_r(c_i, \mathbf{x}) > \mathrm{P_r}(c_j, \mathbf{x})$, $0 < \mathrm{P_r}(c_j, \mathbf{x}) < 0.5$ and $0.5 < P_r(c_i, \mathbf{x}) < 1.0$. The region $g_{2,1}$ is characterised by the three inequalities: $P_r(c_j, \mathbf{x}) > \mathrm{P_r}(c_i, \mathbf{x})$, $0 < \mathrm{P_r}(c_i, \mathbf{x}) < 0.5$ and $0.5 < P_r(c_j, \mathbf{x}) < 1.0$. The confusion region for classes $c_i$ and $c_j$ is composed of the regions $g_{1,2}$ and $g_{2,1}$. Regions $g_1$ and $g_2$ in figure 2.2 represent the instance space regions where there is no class confusion between the two classes.

## 2.7.4 Selection of training data to support the objectives of classification

The dataset selection methods based on theoretical bounds such as the PAC theorems (Valiant, 1984) and Hoeffding-Chernoff theorems (Hoeffding, 1963) directly

support the objectives of probabilistic classification. The dataset selection methods that employ the learning curve to empirically estimate the sufficient sample size (Lutu & Engelbrecht, 2006; Provost et al 1999; John & Langley, 1996) also support the objectives of probabilistic classification. These dataset selection methods attempt to obtain the minimum amount of data selected randomly across the instance space. The selected data enables a classification algorithm to create a predictive model based on data that reflects the natural probability distributions of the classes and variable values.

Density biased sampling (Palmer & Faloutsos, 2000) and one-sided sampling (Kubat & Matwin, 1997) directly support the objectives of discriminative classification for single model construction. These methods have the primary objective of ensuring that those regions in the instance space where prediction is difficult are sufficiently represented in the training datasets. Breiman's (1996) method of bootstrapping a dataset to create many training datasets, Freund and Schapire's (1997) method of boosting with many training datasets, and Chan and Stolfo's (1998) method of partitioning and sampling also support discriminative classification. All these methods attempt to establish the decision boundaries for the classes by using as many training datasets as possible. Additionally, Freund and Schapire's (1997) boosting method attempts to create the highest possible coverage of the decision boundary regions. Partitioning methods that process the whole dataset (Chawla et al, 2001; Hall et al, 2000) do not appear to be directed at any specific view of classification. There is, perhaps, the un-stated assumption that the large dataset is still a very large sample of the real-world data that could be collected for the application domain.

## 2.8 Sources of classification error

It is important to briefly examine the sources of error in predictive classification modeling. Surely if the reasons why errors arise are known, it becomes possible to design data selection methods that have the potential to produce training datasets which minimize the prediction errors. This section provides a discussion of the components of prediction error and factors that influence these components. A discussion of how training datasets can be selected to reduce prediction errors is also provided. The components of prediction error and factors that influence these errors are respectively discussed in sections 2.8.1 and 2.8.2. Methods for selecting

training data for purposes of reducing predictive classification errors are discussed in section 2.8.3.

### 2.8.1 Bias, variance and intrinsic errors in classification

For statistical regression modeling and artificial neural network modeling where the objective function to be minimized is the *mean squared error*, the prediction error has been decomposed into three components, namely *bias, variance* and *intrinsic error* (Giudici, 2003; Geman et al, 1992). For classification modeling in machine learning where the objective function to be minimized is the *0-1 loss* function, the prediction error has been decomposed into the same three components (van der Putten & van Someren, 2004; James, 2003; Domingos, 2000a; Friedman, 1997; Breiman, 1996; Kohavi & Wolpert, 1996; Dietterich & Kong, 1995). A prediction error has a cost (penalty) of 1 and a correct prediction has a cost of 0 for the *0-1 loss* function.

The *bia*s of a predictive model reflects how closely, on average, the (estimated) predictive model is able to approximate the target function. *Bias* reflects the error in the estimation process for the model and is due to the algorithm or inference method as well as the domain for the modeling task (van der Putten & van Someren, 2004; Giudici, 2003; Hand et al, 2001; Friedman, 1997). The *variance* reflects the sensitivity of the (estimated) predictive model to the training sample that is used to create the model. Low variance means that the (estimated) model is more stable to the variations introduced by sampling to obtain the training data (Giudici, 2003; Hand et al, 2001; Friedman, 1997). The phenomenon of *overfitting* which is discussed in the next section is also responsible for the *variance* error (van der Putten & van Someren, 2004). A simple model will have small variance, but large *bias*. A very complex model will have small bias, but large variance (Giudici, 2003).

The third component of the prediction error is called *intrinsic error* (van der Putten & van Someren, 2004; Friedman, 1997; Kohavi & Wolpert, 1996). For a given training dataset and classification algorithm, there exists a hypothetical least-error rate classifier known as the *Bayes optimal classifier* with an error rate known as *Bayes optimal error rate* (Mitchell, 1997; Breiman et al, 1984). The Bayes optimal classifier combines predictions of all possible models (hypotheses) weighted by their posterior probabilities in order to calculate the most probable prediction for a new instance (Mitchell, 1997). *Bayes optimal error rate* is the *intrinsic error* component of the

prediction error and is an irreducible component of the prediction error (van der Putten & van Someren, 2004; Friedman, 1997; Kohavi & Wolpert, 1996).

### 2.8.2 Factors that influence the components of prediction error

Figure 2.3 shows the components of prediction error, the factors that cause these prediction errors and the relationships between the components and the factors, as discussed in section 2.8.1. *Variance error* is caused by sampling variation in the training datasets as well as *overfitting* of models to training data. For purposes of dataset selection from large datasets it is useful to establish how *variance errors* can be reduced through the avoidance of *overfitting.* A predictive model which has a high level of predictive accuracy on the training data and a low predictive accuracy on the test data is called an *overfitted* model (Mitchell, 1997; Hand, 1997; Dietterich, 1995). The causes of *overfitting* and their relationship to *variance error* are depicted in figure 2.3. *Overfitting* arises due to one or a combination of the following reasons. Firstly, when a large number of model parameters is used in the model, the functional form (or structure) of the model becomes very complex.

For classification, examples of model parameters are the nodes of a classification tree and the nodes and connections of an artificial neural network (Engelbrecht, 2002; Hand, 1997). Secondly, when the size of the training dataset is too small and/or does not provide a representative sample for the estimation of the target function then model parameters cannot be accurately estimated (Mitchell, 1997). Thirdly, when the size of the training dataset is very large, it becomes very difficult to distinguish between noise and real structure in the data (Hand et al, 2001; Smyth, 2001; Cohen, 1995). The model is then fitted to the noise and phantom structure in the data. The first two causes of overfitting as discussed above occur most commonly when small datasets are used for training, and it could be argued that these causes of overfitting could be removed by using sufficiently large training datasets. However several researchers have cautioned against the use of very large training datasets (Hand et al, 2001; Smyth, 2001; Hand, 1998).

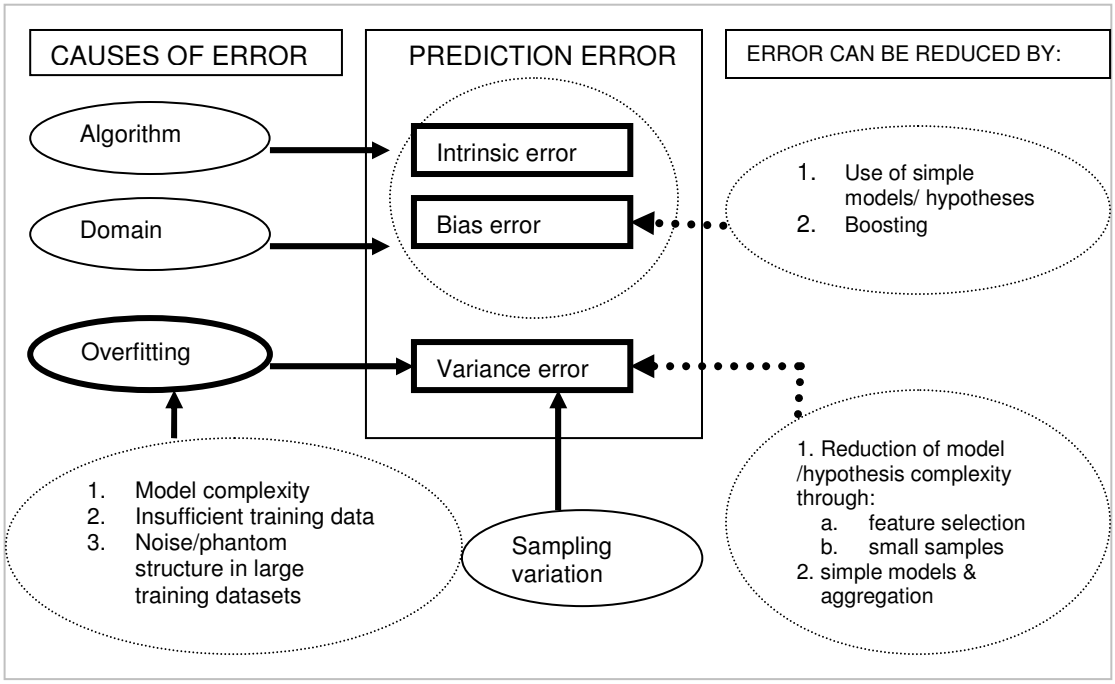


*Figure 2.3: Components of prediction error and factors that influence prediction error*

In statistical data analysis, the terms *massive search* and *data dredging* refer to the practice of processing as much data as possible in order to uncover evidence in support of a hypothesis (Hand et al, 2001; Smyth, 2001; Hand, 1998). The following quote is taken from Hand et al (2001):

> In the 1960s, as computers were increasingly applied to data analysis problems, it was noted that if you searched long enough, you could always find some model to fit a dataset arbitrarily well.

Smyth (2001) has warned against problems of *massive search* as practiced, for example, in association rule mining. Smyth (2001) has argued that even on purely random data where each item's values are generated randomly and independently of other items, a massive search for item associations will 'discover' significant associations between the items. These observations can also be extended to predictive modeling. The main problem here is that it becomes more difficult to distinguish between noise and real structure in the data when datasets are very large (Smyth, 2001; Hand et al, 2001; Cohen, 1995). It is argued in this thesis that one of the objectives of training dataset selection from large datasets should be to minimize the effects of noise and phantom structure in the modeling process. This in turn will lead to a reduction in the *variance* component of the prediction error.

### 2.8.3 Selection of training data to reduce classification error

Figure 2.3 also depicts the methods for prediction error reduction as reported in the literature. Van der Putten and van Someren (2004) have argued that *variance error* can be reduced through the use of methods that select the best predictive features. Methods for feature selection are discussed in chapter 3. The impact of *overfitting* due to noise and/or phantom structure can be reduced through the use of relatively small samples from a dataset. Cohen (1995) has advised that sampling reduces the effects of noise. The use of relatively small training datasets should lead to the reduction of variance error as long as the samples provide good coverage of the instance space. Several researchers have conducted studies to demonstrate that aggregate models based on bagging (bootstrap aggregation) (Breiman, 1996) and boosting (Freund & Schapire, 1997) achieve variance reduction (Friedman, 1997; Kohavi & Wolpert, 1996; Dietterich & Kong, 1995). Dietterich and Kong (1995) have also demonstrated that bias reduction can be achieved through the use of simple models plus increased representation of decision boundary instances as is done for boosting algorithms.

## 2.9 The limitations of current methods of dataset selection

The dataset selection methods discussed in this chapter for selecting training data from large datasets may be divided into three categories. The methods in the first category select and use all of the data in the belief that maximum accuracy will be achieved by processing all the data (Chawla et al, 2001; Hall et al, 2000). For the implementation of these methods, partitioning has been used in order to achieve parallel execution and fast computation of classification algorithms on massively parallel supercomputers. One obvious problem with this approach is that overfitting will occur when millions of records are used to create a predictive model. A second problem is that there is no clear explanation in the reported studies on how this approach is expected to reduce prediction error. It is the author's opinion that the objective here is to provide a very high coverage of the instance space. However, given the caution by Smyth (2001) concerning chance structure in very large datasets, one is led to conclude that high coverage of the instance space has its limits.

The methods in the second category select a subset of the data with the expectation that there is a minimum sample size, $n_{min}$, beyond which no further gains in predictive accuracy are possible (Lutu & Engelbrecht, 2006; Provost et al, 1998; John & Langley, 1996; Toivonen, 1996). The rationale behind this approach is that small training sets are preferred when it is prohibitively expensive to process very large datasets in reasonable time. This approach works well for single model construction. However, given the strong evidence of the superior performance of aggregate models, there is a need in the field of data mining to direct more research effort towards training dataset selection for aggregate model construction.

The methods in the third category attempt to create training datasets with balanced class distributions (Chan & Stolfo, 1998). These methods support training dataset selection for aggregate model construction. Additionally, the methods are aimed at solving the specific problem of creating predictive models from large datasets with skewed class distributions.

## 2.10 Proposed approach to selection of training data from very large datasets

It is the author's opinion that when large amounts of data are available, it is productive to use as much data as possible, while at the same time avoiding the problems of overfitting and the modeling of chance or phantom structure in the large datasets. The discussions in this chapter have revealed that, by reducing the bias and variance components of the prediction error, a good predictive model is obtained. This assertion is strongly supported by the success of bootstrap aggregation (Breiman, 1996) and boosting (Freund & Schapire, 1997) for small datasets. These methods are known to reduce the bias and variance components of the prediction error (van der Putten & van Someren, 2004; Friedman, 1997; Kohavi & Wolpert, 1996; Dietterich & Kong, 1995). Additionally, at the present time, there are many research efforts being undertaken in the area of aggregate model construction. These research efforts are largely motivated by the success of bootstrap aggregation. This section provides a discussion of the training dataset selection approach that was studied for this thesis, for purposes of achieving bias and variance reduction. The proposed methods for variance reduction are discussed in section 2.10.1. The proposed methods for bias reduction are discussed in section 2.10.2.

### 2.10.1 Variance reduction methods

Variance reduction can be achieved through at least four methods. The first method of variance reduction involves the use of many training datasets to create base models for aggregation through voting. For large datasets, this can be achieved by obtaining many randomly selected training samples from the large dataset. Each training sample is then used to create one base model. The second method of variance reduction is to provide as much coverage as possible of the decision boundary regions, as is done in boosting. For large datasets, this can be achieved by ensuring that the training datasets have many instances drawn from the decision boundary regions. The third method of variance reduction is through the avoidance of overfitting. For large datasets, the use of relatively small randomly selected training samples results in the reduction of the amount of noise (incorrect data values) and the effects of chance structure in the data. The fourth method of variance reduction is to select a good set of predictive features (van der Putten & van Someren, 2004).

The combination of the above four methods, namely: selection of many training datasets for the base models, provision of high coverage of the decision boundary regions, and the usage of relatively small training samples for the base models and, feature selection should lead to a significant reduction of the variance component of the prediction error. This approach to dataset selection was adopted for this thesis. For this proposed approach, productive usage of large amounts of data is achieved by ensuring that each of the training datasets for the base models is taken from a different region of the instance space. This approach should result in the usage of large amounts of data in the training process, without creating the problems of overfitting.

### 2.10.2 Bias reduction methods

Bias reduction can be achieved through at least three methods. The first method of bias reduction is through sampling to reduce the effects of noise in the training data. The second method of bias reduction is through making improvements to the algorithm for purposes of reducing bias. The third method of reducing bias is due to Dietterich and Kong (1995). Dietterich and Kong (1995) have argued that the

decomposition of a *k*-class problem into a number of 2-class problems whose solution is then converted back (combined) into the *k*-class solution, results in the correction for bias errors in the classification algorithm (Dietterich & Bakiri, 1995). Two of the three methods discussed above were combined for the proposed methods of training dataset selection. The two methods used for bias reduction were boosting of training datasets and decomposition of *k*-class problems into 2-class problems and *j*-class ($j < k$) problems.

## 2.11 Conclusions

The need for dataset selection has been made explicit, using examples of several application domains where data is collected in massive quantities. The examples have covered both business and scientific application areas. Methods for predictive modeling for classification using very large datasets have been discussed. These include the use of a single model and the use of aggregate models for prediction. The discussion has revealed that the methods available for aggregate model construction may result in an increase in prediction performance, but this is not guaranteed for every domain. Methods for training dataset selection have been discussed. The methods include single sample selection to obtain one dataset for training, dataset partitioning, and, a combination of partitioning and sampling to obtain several training datasets for base models. Additionally, for a given dataset, there may be other objectives, such as balancing the class distribution, which will determine the data selection method.

A discussion of the problems associated with the use of very large training datasets has been given, and reasons have been given on why it is not desirable to use very large training datasets. The various sources of classification error have been discussed. Prediction error is traditionally decomposed into two components: bias and variance. Methods of reducing bias and variance through dataset selection have been discussed. Finally, the proposed general approach to training dataset selection from large datasets in order to reduce bias and variance has been given in the last section. The next chapter presents a discussion of feature selection from very large datasets. The research methods that were used for the studies reported in this thesis are presented in chapter 4.

# Chapter 3

# The Feature Selection Problem

Feature selection should be treated as an integral part of dataset selection. It was pointed out in the last chapter that the use of a good set of predictive features leads to the reduction of the variance component of the prediction error. This chapter provides an overview of the feature selection problem for classification tasks in predictive data mining. A review of the available methods for feature selection from small datasets is provided. The methods discussed fall into two categories. The methods in the first category have been studied by researchers in the context of small datasets. The methods in the second category have been studied to specifically address feature selection for data mining for high dimensional datasets and for large datasets. The dangers of using a single sample to determine relevant features are highlighted. An analysis is conducted of commonly used methods of measuring class-feature correlations and more robust measures of class-feature correlations are discussed. Existing methods for validation of correlation values are also discussed.

This chapter is organised as follows: The need for feature selection is discussed in section 3.1. Methods for implicit and explicit feature selection are respectively discussed in sections 3.2 and 3.3. Merit measures for heuristic feature subset search are given in section 3.4. Sections 3.5 and 3.6 respectively provide a discussion of correlation measurement and validation methods for correlations. Section 3.7 concludes the chapter.

## 3.1 The need for feature selection

The problem of feature subset selection is concerned with finding a subset of the original features of a dataset, such that an induction algorithm running on data containing only the selected features will generate a predictive model that has the highest possible accuracy. It is essential to select a subset of those features which are most relevant to the prediction problem and are not redundant (Hand et al, 2001; Hall, 1999, 2000; Liu & Motoda, 1998; Blum & Langley, 1997; Aha & Bankert, 1996).

Feature selection is central to training dataset selection since one of the motivating factors for training dataset selection is to improve predictive accuracy through variance reduction. This section provides a discussion of the need for feature selection as well as definitions of relevance and redundancy as reported in the literature on feature selection for machine learning and data mining. Section 3.1.1 provides definitions of feature relevance and redundancy. Section 3.1.2 discusses a major problem for predictive modeling called the curse of dimensionality.

### 3.1.1 Feature relevance and redundancy

It is generally agreed that a predictive model should be constructed using a subset of features which are most relevant to the prediction problem and are not redundant (Hand et al, 2001; Hall, 1999, 2000; Liu & Motoda, 1998; Blum & Langley, 1997; Aha & Bankert, 1996). Blum and Langley (1997) have provided definitions of *relevance*, *strong relevance* and *weak relevance*. A feature $f_i$ is *relevant* if a change in the feature's value can result in a change in the value of the predicted (class) variable. A feature $f_i$ is *strongly relevant* if the use of $f_i$ in the predictive model eliminates the ambiguity in the classification of instances. A feature $f_i$ is *weakly relevant* if $f_i$ becomes *strongly relevant* when a subset of the features is removed from the set of available features. By implication, a feature is *irrelevant* if it is not *strongly relevant* and it is not *weakly relevant*. Koller and Sahami (1996) have provided a definition of *redundancy* for a feature. A feature $f_i$ is *redundant* relative to the class variable *C* and a second feature $f_j$ if $f_i$ has stronger predictive power for $f_j$ than for the class variable *C*. Koller and Sahami (1996) have used the term *Markov blanket* to refer to the above relationship between the features $f_j$ and $f_i$, that is $f_j$ is a *Markov blanket* for $f_i$.

For purposes of making feature selection decisions however, many researchers (e.g. Ooi et al, 2007; Yu & Liu, 2004; Blum & Langley, 1997; Hall, 1999,2000) have interpreted a *relevant* feature as one which is highly correlated with the class variable. Ooi et al (2007) and Hall (1999, 2000) have interpreted a redundant feature as one which is highly correlated with all the other features. Yu and Liu (2004) and,

Koller and Sahami (1996) have however implemented the above definition of *redundancy* (based on the *Markov blanket* property) in feature selection.

### 3.1.2 The curse of dimensionality

Hand et al (2001) have discussed the problem of the *curse of dimensionality*. This problem is defined as the exponential rate of growth of the number of unit cells in the instance space as the number of predictive features increases. The curse of dimensionality reduces the density of instances in the instance space. Recall from section 2.2.1 that the instance space is the *d*-dimensional space defined by the *d* predictor variables. Reduction of the density of instances in the instance space causes instances to appear to be very far away from each other. This makes it difficult for discriminative classification algorithms to establish decision regions and decision boundaries for classes. It also becomes more difficult for probabilistic classification models to estimate probability densities in the different regions of the instance space.

The reduction of the number of features reduces the size of the instance space, and therefore also decreases the complexity of the prediction problem. Secondly, according to the PAC theory (Mitchell, 1997), as the hypothesis space size decreases in size, so does the sample complexity.

## 3.2 Implicit feature selection

Decision tree algorithms have the capability to implicitly identify the most predictive features as the tree is constructed. In addition, a decision tree is normally pruned so that it retains only those features which provide statistically significant predictive power (Osei-Bryson, 2004, 2007; Breiman et al, 1984). Kohavi and John (1997) have reported feature selection studies which have revealed that decision tree algorithms are not always able to eliminate irrelevant features. The studies reported by Kohavi and John (1997) on credit approval and diabetes datasets from the UCI Machine Learning repository (Ascuncion & Newman, 2007; Blake & Merz, 1998) have shown that the performance of decision trees constructed by the C4.5 algorithm deteriorates significantly when a single irrelevant feature is added to the dataset. Langley (1994) has observed that artificial neural networks (ANNs) and the Naïve Bayes (NB)

algorithms also perform an implicit ranking as they build classifiers. ANNs and NB assign larger weights to the more relevant features and smaller weights to the less relevant ones. A very important point to emphasize here is that, even though many inductive algorithms perform implicit feature selection, all induction algorithms do benefit from explicit feature selection, before the algorithm is presented with the data.

## 3.3 Explicit feature selection

Explicit feature selection involves the use of a separate step to select those features that are considered relevant for a predictive modeling task. Specific to data mining there may be hundreds or thousands of features in a dataset. All potentially relevant features must be identified first. The identification is typically done by a domain expert (Guyon & Elisseef, 2003). The task of the domain expert is to select those variables that are known to have a bearing on the domain for the prediction task. As an example, van der Putten and van Someren (2004) have quoted the winner of the COIL 2000 competition who stated that only the variables representing wealth and personal behaviour of individuals were useful for the competition dataset. After the initial selection of features, a second step is conducted so that the most effective (predictive) features are selected from the pool of potentially relevant features (Guyon & Elisseef, 2003). Section 3.3.1 presents the categories of feature selection methods. Wrapper methods are discussed in section 3.3.2. Methods that use pure ranking are presented in section 3.3.3. Heuristics search methods and relevance and redundancy analysis methods are respectively discussed in sections 3.3.4 and 3.3.5. Feature selection methods for large datasets are discussed in section 3.3.6.

### 3.3.1 Categories of feature selection methods

Feature selection methods may be categorized as wrapper or filtering methods (Hall, 1999; Kohavi & John, 1997). Wrapper methods incorporate model construction with feature selection, and select that subset of features which provides a model with the highest predictive performance (Blum & Langley, 1997; Kohavi & John, 1997). Filtering methods on the other hand, select feature subsets without constructing predictive models from these features (Ooi et al, 2007; Yu & Liu, 2004; Guyon & Eliseeff, 2003; Hall, 1999; Blum & Langley, 1997). Three filtering methods are discussed in this section. The first method called *pure ranking*, involves sorting the

list of features in descending order of relevance and then selecting the top *w* features or selecting features whose level of relevance is above a user specified threshold (Yu & Liu, 2004). The second method, called *feature subset search*, involves a forward search or backward search based on a list of ranked features in order to determine the best subset of features which maximises relevance and minimises redundancy (Ooi et al, 2007; Hall, 1999, 2000; Blum & Langley, 1997). The third filtering method involves *relevance* and *redundancy analysis* as two distinct steps (Yu & Liu, 2004). The rest of this section provides a discussion of wrapper and filtering methods for feature selection.

### 3.3.2 Feature selection using wrapper methods

Wrapper methods incorporate model construction with feature selection (Blum & Langley, 1997; Kohavi & John, 1997). For wrapper methods, different feature subsets are selected, a predictive model is constructed for each feature subset and the feature subset which produces the model with the highest predictive performance is selected. The accuracy for different feature subsets is measured using 10-fold cross validation (Blum & Langley, 1997). Wrapper methods have typically been used for small datasets with a small number of features. It has been argued that wrapper methods are not suitable for large datasets as encountered in data mining (Hall, 1999) or datasets of high dimensionality (Yu & Liu, 2004) due to the intensive computational requirements. Even though the research reported in this thesis focussed on filtering methods, it is the author's opinion that when many samples are used for model construction and testing for the wrapper approach then more reliable feature selection should be achieved.

### 3.3.3 Feature selection based on pure ranking

Feature ranking involves two steps. In the first step, a value is assigned to each feature to indicate its level of relevance to the prediction task. In the second step, the list of features is sorted and the top *w* features are selected. A commonly used measure of relevance is the correlation of the feature to the class. To compute the correlation between a numeric-valued feature and the class variable, Pearson's correlation coefficient is commonly used (Ooi et al, 2007; Hall, 1999, 2000). To compute the correlation between a qualitative feature and the class variable, the

symmetrical uncertainty coefficient may be used (Yu & Liu, 2004; Hall, 1999, 2000). Guyon and Elisseeff (2003) and Bekkerman et al (2003) have observed that various feature selection algorithms include feature ranking as a preliminary step. The purpose of this preliminary step is to identify those features that have the potential to appear in the final subset of selected features. Various feature selection methods, on the other hand, simply use feature ranking as the selection method (Guyon & Elisseeff, 2003).

### 3.3.4 Feature selection based on heuristic search

Heuristic search (Luger & Stubblefield, 1993; Pearl, 1984) is the process of intelligently narrowing the search through a potentially very large search space of solutions in order to identify a satisfactory solution. At every decision point in the search, a heuristic search procedure employs a merit (heuristic) measure to determine the best path to expand in the search space. For the problem of feature subset search the space of all possible problems is the set of all possible combinations (the power set) of the features in the set of candidate features. The candidate features are those features that have been pre-selected through a process of ranking as discussed in section 3.3.1. Each state in the search space specifies a possible subset of features (Blum & Langley, 1997).

Algorithms for feature subset search are classified as forward selection or backward selection algorithms. The initial state for forward selection algorithms is one where no feature has been selected. For backward elimination algorithms on the other hand, the initial state is one where all features are selected. A hill-climbing search is commonly conducted. The state which currently maximises the measure of merit is selected for further expansion. Commonly used measures include information gain and merit measures based on the class-feature and feature-feature correlations (Ooi et al, 2007; Hall, 1999, 2000). For forward search, stopping criteria include stopping when addition of a new feature does not result in any significant increase in the employed measure (Hall, 1999, 2000), or when a pre-specified number of features has been selected (Ooi et al, 2007).

### 3.3.5 Feature selection using relevance and redundancy analysis

Yu and Liu (2004) have proposed the method of *relevance* and *redundancy analysis* for feature selection aimed at datasets of very high dimensionality. Yu and Liu (2004) have argued that heuristic subset search and wrapper methods are not feasible for high dimensional datasets due to the quadratic time complexity of heuristic search algorithms. The feature selection method proposed by Yu and Liu (2004) consists of two distinct steps, namely *relevance analysis* and *redundancy analysis*. For relevance analysis class-feature correlations are used as a basis to eliminate all features whose level of correlation to the class variable is below a user-specified threshold. The features selected in the *relevance analysis* step are used as input to the *redundancy analysis* step. *Redundancy analysis* aims to select those features that are relevant with respect to the class variable and are not redundant with respect to any other relevant feature. For each relevant feature $f_i$, every feature $f_j$ which has a smaller class-feature correlation than $f_i$ (lower relevance than $f_i$), but has a high feature-feature correlation with $f_i$ (more strongly correlated to $f_i$ than to the class variable) is eliminated.

Yu and Liu (2004) have demonstrated that even though this method has quadratic time complexity in the worst case, in practice the time complexity is close to linear time when many redundant features are present. The studies reported in this thesis were limited to datasets of moderately high dimensionality. It will be useful in future to study how the validation methods proposed in this thesis can be adapted to feature selection for very high dimensional datasets.

### 3.3.6 Feature selection for large datasets

For feature selection for small datasets all the instances in the dataset are used in the selection process. Feature selection from large datasets poses new challenges for feature selection. A dataset may be large because it consists of a large number of instances, or a large number of potentially predictive features, or both. From a computational perspective, the time complexity of feature selection algorithms makes it infeasible to use all of the data in a large dataset (Liu & Setiono, 1998a, 1998b). From a statistical perspective, the problems of massive search (Smyth, 2001) which were discussed in chapter 2 make it infeasible to use all of the data. Liu and Setiono

(1998a, 1998b) have proposed a stochastic (probabilistic) method of feature selection for large datasets of high dimensionality. The method employs random feature subset generation and evaluation in conjunction with dataset sampling. At each step of the process a sample of dataset instances is created and a subset of randomly selected features is generated and evaluated. The process stops when the selected feature subset is established to be optimal.

## 3.4 Merit measures for heuristic search of feature subsets

Heuristic search for feature selection was discussed in section 3.3.2. Suppose that at the current step of heuristic search, *w-1* features $f_1..,f_{w-1}$ have been selected from the candidate set of features $W$ and $u = \mid W \mid -w-1$ features are still unselected. In order to select the next feature $f_w$, feature subsets $FS_i = \{f_1..,f_{w-1},f_w\}, i = 1..u$ are created so that for each subset $FS_i$ the feature $f_w$ is one of the $u$ features that are still unselected. A mathematical function is typically used to compute a measure of merit which guides the heuristic search in the selection of the best feature subset $FS^*$. The correlation-based feature selection (CFS) method proposed by Hall (1999, 2000) uses the merit measure defined as

$$Merit_{CFS} = \frac{w.\overline{corr}_{cf}}{\sqrt{w + w(w-1)\,\overline{corr}_{ff}}} \qquad (3.1)$$

where $\overline{corr}_{cf}$ is the mean correlation between each feature and the class variable, $\overline{corr}_{ff}$ is the mean correlation between the features in subset *FS* and, *w* is the number of features in the subset *FS*. The numerator on the right hand side of equation (3.1) measures the level of relevance of the feature subset, while the denominator measures the level of redundancy of the feature subset. The differential Prioritisation (DP) method, proposed by Ooi et al (2007) uses the merit measure defined as

$$Merit_{DP} = (\overline{corr}_{cf})^{\alpha}.(RD)^{1-\alpha} \qquad (3.2)$$

where

$$RD = \frac{1}{w^2} \sum_{fi, fj \in F, fi \neq fj} (1 - |corr_{fifj}|) \tag{3.3}$$

and $|corr_{fifj}|$ is the magnitude of the correlation between two features. The first term on the right hand side of equation (3.2) measures the level of relevance, while the second term measures the level of redundancy of the feature subset. The parameter $\alpha$ is used to control the levels feature relevance and redundancy based on the user's preference. The merit measures of equations (3.1) and (3.2) both reflect the fact that the subset of selected features should have a high level of relevance and a low level of redundancy. The main difference between the two equations is that the relative importance of relevance and non-redundancy are fixed in equation (3.1) while equation (3.2) allows the analyst to specify the relative importance of relevance and non-redundancy through the parameter $\alpha$.

The correlation coefficients $corr_{cf}$ and $corr_{ff}$ are computed using either Pearson's correlation coefficient for quantitative features or the symmetrical uncertainty coefficient for qualitative features. For two quantitative features *X* and *Y*, the correlation is measured using Pearson's correlation coefficient, which is defined as

$$r_{XY} = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{(n-1) S_X S_Y} \tag{3.4}$$

where $\bar{x}$ and $\bar{y}$ are the sample means for *X* and *Y* respectively, $S_X$ and $S_Y$ are the sample standard deviations for *X* and *Y*, and *n* is the size of the sample used to compute the correlation coefficient.

In general, the level of association between two qualitative variables *X* and *Y* can be established using measures derived from Pearson's $\chi^2$ statistic, such as the $\phi$ statistic and Cramer's *V* statistic (Giudici, 2003). These measures of association have a similar interpretation as a correlation coefficient for quantitative features (Giudici, 2003). The symmetrical uncertainty (*SU)* coefficient derived from the entropy function is an alternative measure of association between two qualitative features and also has a similar interpretation as a correlation coefficient for quantitative variables (Yu & Liu, 2004; Hall, 1999, 2000). The symmetrical uncertainty coefficient *SU* given in equation (3.5) is defined in terms of *E(X)* and *E(Y)*

(the entropy in *X* and Y respectively) and *E(X|Y)* (the entropy of *X* conditioned on Y). The definitions for *E(X), E(X|Y)* and other related definitions of entropy are given in appendix B. The *SU* coefficient is defined as

$$SU = 2.0x\left[\frac{E(X) - E(X \mid Y)}{E(X) + E(Y)}\right] \tag{3.5}$$

The details for the computation of Pearson's $\chi^2$ statistic, the $\phi$ statistic and Cramer's *V* statistic are given in appendix B. The *SU* coefficient was used in the experiments for this thesis as it is more commonly used in feature selection studies (Yu & Liu, 2004; Hall, 1999, 2000).

When one feature *X* is qualitative and the other feature *Y* is quantitative, a weighted Pearson's correlation is used. If the qualitative feature *X* has *V* levels, $L_1 \ldots L_V$, then *V* binary features $B_1 \ldots B_V$ are created through a process called binarisation. Each of the binary features is then correlated with the quantitative feature *Y*. The binary feature $B_i$ is assigned the value *1* when *X* has level $L_i$ and *0* otherwise. The weighted correlation coefficient between *X* and *Y* is computed as

$$corr_{XY} = \sum_{i=1}^{V} P_r(X = L_i).corr_{Bi,Y} \tag{3.6}$$

where $P_r(X = L_i)$ is the prior probability that *X* has the level $L_i$ and $corr_{Bi,Y}$ is the correlation coefficient between a binary feature and the variable *Y*.

The computation of correlation coefficients using equations (3.6) is feasible for qualitative features with few distinct levels. If a qualitative feature has many levels (e.g. 20 and above) then the number of binary features to be created becomes excessively large, which in turn increases the computational time for the correlation coefficients. Equation (3.6) is especially useful for computing correlations between quantitative features and the class variable. Since many classification tasks have few classes the computations for equation (3.6) do not pose a problem.

## 3.5 Measuring correlations

In general, there are three common methods of measuring the correlation between two quantitative random variables *X* and *Y*: Pearson's correlation coefficient, Spearman's *rho,* and Kendall's *tau* (Wilcox, 2001). Each of these correlation measures is exactly zero when *X* and *Y* are independent, and have values that range between *-1* and *+1* to indicate the level and direction of the correlation. Pearson's correlation coefficient is commonly used to estimate the magnitude of the association between the features and the class for a dataset (Ooi et al, 2007; Hall, 1999, 2000). The main advantage of Pearson's correlation is that it is very efficient to compute, compared to the other correlation measures. However, for many datasets used in data mining the meaning of Pearson's correlation coefficient may not be what one expects. The problems associated with Pearson's correlation coefficient and the advantages of using robust measures of correlation are discussed in this section. The problems associated with Pearson's correlation coefficient and robust measures of correlation are respectively presented in sections 3.5.1 and 3.5.2.

### 3.5.1 Problems with Pearson's correlation coefficient

Pearson's product moment correlation coefficient for a data sample is computed as shown in equation (3.4). The computation involves the sample mean, sample variance and sample covariance. Furthermore, the sample mean, variances and covariances can be computed in a single pass of the dataset. Wilcox (2001) has observed that Pearson's correlation coefficient is the best estimator of the correlation between the random variables *X* and *Y* when *X* and *Y* have normal probability distributions. Wilcox (2001) has defined the *finite sample breakdown point* of a statistic computed from a sample as the smallest proportion of outliers in the sample required to make the value of the statistic arbitrarily large or arbitrarily small. Wilcox (2001) has demonstrated that the *finite sample breakdown point* for the sample mean and sample variance is *1/n*, where *n* is the sample size. This means that a single outlier can cause these measures to be arbitrarily large or small. For Pearson's correlation coefficient, Wilcox (2001) has also demonstrated that a single outlier can mask an otherwise strong association between *X* and *Y*.

The above observations by Wilcox (2001) have serious implications for feature selection methods based on Pearson's correlation coefficient. First of all, highly

predictive features may be discarded, simply because the presence of outliers causes the computed sample correlation coefficient to be small, or worse still, to be insignificant. Secondly, non-linear associations will produce very small correlation coefficients, which will cause otherwise relevant features to be discarded. In a nutshell, in the presence of outliers and non-linear associations, and this should be expected in data mining, Pearson's correlation coefficient will provide a feature ranking which is incorrect. When this is the case, there is an increased risk of creating a model that has poor predictive performance. The reader will recall that the use of poor predictors results in an increase in the inherent error or variance component of the prediction error. Wilcox (2001) has made a strong point that*: 'if we are told r (Pearson's sample correlation coefficient), and nothing else, we cannot deduce much about the details of how X and Y are related'.* A third problem with Pearson's correlation (and other correlation measures) is that the two random variables *X* and *Y* may be strongly associated for some of the values and not for the whole range of values. When this is the case, computing the correlation coefficient between *X* and *Y* based on the whole range of each of the variables, will provide very small correlation coefficient values which will lead to the assumption that there is no association between the variables.

### 3.5.2 Robust measures of correlation

Wilcox (2001) has discussed three ways of handling outliers when computing sample correlations. The first method is to compute a winsorised Pearson's correlation coefficient, the second method is to use Spearman's *rho* correlation coefficient, and the third method is to use Kendall's *tau* correlation coefficient. To winsorise the values of a random variable *X*, the smallest *z%* and largest *z%* of values in the sample are altered. The alteration involves replacing each of the small values with the smallest of the unaltered values, and replacing the large values with the largest unaltered value (Wilcox, 2001). For correlation computations, the values of both *X* and *Y* must be winsorised, prior to computing the sample means, variances and covariance. The problem with computing the trimmed means and winsorised variances is that the values of the variables must be sorted first. For a multivariate dataset with *d* variables $2d + d^2$ sorting operations must be conducted for the computation of the class-feature and feature-feature correlations. These intensive computations can be avoided by using Spearman's *rho* or Kendall's *tau* correlation coefficients.

For Spearman's *rho*, the values of *X* and *Y* are converted to ranks, *1,..n*. Spearman's *rho* is then computed with Pearson's correlation formula using the rank values. This way, the effect of outliers is avoided. Even though this method eliminates the problem of outliers, its computational requirements are not modest. It is still necessary to sort the values of *X* and *Y*. For multivariate data, $2d + d^2$ sorting operations are still needed for the *X,Y* pairs. For Kendall's *tau* computations are needed for the probabilities $\pi_c$ and $\pi_d$. $\pi_c$ is the probability that, given two random pairs of values $(x_1, y_1)$ and $(x_2, y_2)$, when $x_1 > x_2$ then $y_1 > y_2$, and when $x_1 < x_2$ then $y_1 < y_2$. $\pi_c$ is called the probability of *concordance* between the random variables *X* and *Y*. $\pi_d$ is the probability that the opposite is the case, and is called the *discord*ance between *X* and *Y*. The value of Kendall's *tau* is computed as $\tau = \pi_C - \pi_D$. The probabilities $\pi_c$ and $\pi_d$ are estimated by comparing all possible sets of pairs of values of the variables *X* and *Y*, that is, $(n-1)/2$ pairs. Kendall's *tau* is computationally more efficient than Spearman's *rho* since Kendall's *tau* does not require sorted data. The method is also a good alternative to Pearson's coefficient when outliers and non-linearity are present. However, the computational time complexity for Kendall's *tau* is still quadratic in *n*, the size of the sample used to estimate the correlations.

## 3.6 Validation methods for feature selection

Guyon and Elisseeff (2003) have defined feature validation methods as those methods that are used to determine the number of significant features, guide and halt the feature subset search or, evaluate the final performance of the models based on the selected features. The discussion in this section is concerned with methods for determining the validity of the decision to select a given feature for inclusion in the set of predictive features. Section 3.6.1 discusses the need for validation of class-feature and feature-feature correlation coefficients. The practical significance of correlation coefficients is discussed in section 3.6.2. Validation methods based on hypothesis testing and based on fake variables are respectively discussed in sections 3.6.3 and 3.6.4.

### 3.6.1 The need for validation of correlation coefficients

In general, filtering methods rank features based on the correlation or some other measure, with the class. The higher the measure the more predictive a feature is assumed to be. Smyth (2001) has argued that a large correlation coefficient between the random variables *X* and *Y* in a given data sample could be purely due to chance. If the feature-class correlations were measured on a different sample, the correlation coefficient would take on different values. The same argument applies to any measurement that is taken on a data sample. Measures such as the class-feature correlation coefficient $corr_{cf}$ and the feature-feature correlation coefficient $corr_{ff}$, which appear in equations (3.1), (3.2) and (3.3) are therefore random variables with associated probability distributions. It is essential to establish the validity of these correlation measures before they are used in feature ranking and subset selection. One validation method for feature correlations that has been reported in the literature is the use of fake variables (Guyon & Elisseeff, 2003; Bi et al, 2003; Stoppiglia et al, 2003). A fake variable or probe, is a variable whose values are generated purely at random. Such values should not have any correlation with the class variable. When measuring correlations using either Pearson's *r* or Kendall's *tau*, any features with a correlation value that is lower than that of the fake variables should be discarded.

### 3.6.2 Practical significance of correlation coefficients

It was pointed out in section 3.1 that feature relevance and redundancy are typically defined in terms of the strength of correlations. Blum and Langley (1997) have defined a relevant feature as one which is *highly* correlated with the class variable. Hall (1999) and Ooi et al (2007) have defined a redundant feature as one that is *highly* correlated with other features. Cohen (1988) has observed that different fields of study and research define the quantitative adjectives for correlations, namely *small, medium, large*, differently. In the physical sciences where the variable values are obtained from high precision instruments, a correlation coefficient of *0.9* is considered small (Cohen, 1988). In Economics, a correlation coefficient of *0.6* is considered small (Coetsee, 2007).

For the field of Behavioural Sciences research, Cohen (1988) has suggested the following approach to interpreting the magnitude of a correlation. A value in the range [*0.10*, *0.30*) indicates a small/weak correlation. A value in the range [*0.30*, *0.50*)

indicates a medium correlation. A value in the range [*0.50*, *1.00*] indicates a large/strong correlation. Cohen (1988) has argued that these criteria are suitable for the social sciences, since for this field of research there is always a large number of complicating factors in the experimental setup and the measuring instruments used to collect data. One implication of Cohen's (1988) criteria for interpreting correlations is that correlation values of less than *0.10* have no practical significance, even though such correlation coefficients might appear to be statistically significant, especially for very large samples.

### 3.6.3 Validation based on hypothesis testing for correlation coefficients

In many fields of scientific enquiry, it is common practice to establish the statistical significance of the correlation coefficient, *r*, using Student's *t-test* for correlations (Wilcox, 2001; Kanji, 1999). One can then test the null hypothesis: $H_0$*: 'the correlation between the two variables is zero',* and the alternative hypothesis $H_a$*: 'the correlation between the two variables is not zero'.* If the null hypothesis $H_0$ is rejected, then one concludes that the two variables have a statistically significant (linear) relationship indicated by the direction and magnitude of the correlation coefficient *r*. The *T* statistic used for testing $H_0$ and $H_a$ as defined above is

$$T = r\sqrt{\frac{n-2}{1-r^2}} \tag{3.7}$$

where *n* is the sample size used to estimate *r*. Under normality and when the population correlation $\rho = 0$ the quantity *T* has a Student's *t* distribution with *n-2* degrees of freedom. Even though this is a fairly popular test in many research areas, the author is not aware of any reported usage of this test in feature selection for computational data mining.

To test the hypothesis $H_0 : r = c$, that is, the correlation coefficient is some value *c* other than zero, Fisher's transform is used to convert the correlation coefficients into the *Z* statistic as follows (Cohen, 1995; Cohen, 1988):

$$Z = \frac{Z(r) - Z(c)}{\sigma_{z(r)}} \tag{3.8}$$

In equation (3.8) *Z(r)* is Fisher's *Z* transform of *r* and is computed as

$$Z(r) = \frac{1}{2}\ln\left(\frac{1+r}{1-r}\right) \tag{3.9}$$

where $\sigma_{z(r)}$ is the estimated standard error of the sampling distribution of *Z(r)* and is computed as $1/\sqrt{(n-3)}$, where *n* is the number of values used to compute *r*. *Z(c)* is Fisher's Z transform of *c* and can be similarly computed.

### 3.6.4 Validation based on fake variables

Stoppiglia et al (2003) have proposed the use of probes (fake variables) for determining the cut-off point between relevant and irrelevant features. A probe is a random variable whose values may be generated from any probability distribution. Stoppiglia et al (2003) have argued that, since such a random variable should not have any significant correlation to the target function, it should be ranked last if an infinitely large amount of data were available. However, since the amount of data is finite, the probe should appear somewhere in the ranked list and all features that are ranked below the probe should be discarded. Since the probe is a random variable, its rank in the list of features is also a random variable. The decision to keep or discard features based on the probe's value should be based on the probability that this feature's ranking is higher or lower than that of the probe. Stoppiglia et al (2003) have recommended that the designer of the model should choose a tolerable risk of selecting or discarding the feature based on the ranking of the probes.

Bi et al (2003) have reported studies on feature selection for micro-array datasets. Bi et al (2003) have observed that the feature selection process can be very unstable in the sense that each time a feature set is selected it consists of totally different features. Since micro-array datasets are typically small, Bi et al (2003) have used bootstrap samples and merged the results from these samples. Bi et al (2003) have used 3 fake variables drawn randomly from Gaussian distributions and have discarded all variables that are less relevant than one of the fake variables. It should

be noted however, that since fake variables are also random variables, one should expect that the ranking of the fake variable will vary from sample to sample.

## 3.7 Conclusions

The need for feature selection for predictive data mining has been discussed in this chapter. Feature selection methods for implicit and explicit feature selection have been presented. Implicit feature selection is performed by the induction algorithm, while explicit feature selection is performed by an algorithm whose sole purpose is to select the best features for a given prediction task. Methods for measuring class-feature correlations have been discussed and the problems inherent in these methods have been highlighted.

Filtering methods are heavily dependent on the class-feature and feature-feature correlation measures. Many researchers have used Pearson's correlation coefficient to establish class-feature and feature-feature correlations. Even though this is the most reliable and efficient way of measuring linear correlations, it is not the most appropriate measure when correlations are non-linear, and when outliers are present. It is useful to study more robust measures of correlation for feature selection.

The validation methods for selected feature subsets that have been reported are based on the use of fake variables. These methods have been studied in the domain of micro-array datasets, where the datasets are typically small: less than 200 instances. Given that fake variables are random variables, it is useful to conduct studies on how probes will perform in the presence of much larger datasets, and to devise methods of using probes to select reliable feature subsets. In many fields of research, hypothesis testing is used to establish the statistical significance of a correlation coefficient. There are no reported studies of this nature for feature selection for computational predictive data mining. It is the author's belief that, when large amounts of data are available, opportunities arise for researchers to conduct studies of this nature.

Filtering methods conduct feature subset selection based on a general definition of relevance, redundancy and correlation strength, even though different application domains have different interpretations of what it means for two variables to be highly

correlated. Algorithms are needed that can incorporate domain-specific definitions of feature relevance and redundancy. Even though many studies have been reported on feature selection for small datasets, to the author's knowledge there are very few reported studies (e.g. Liu & Setiono, 1998) that specifically address feature selection from very large datasets. It is the author's opinion that it is useful to conduct more studies of feature selection methods that can make use of the large amounts of available data in order to perform reliable measurement and validation of class-feature correlations and as a result, provide reliable feature subsets.

Chapter 2 presented a discussion of current methods of training dataset selection from large datasets. It was argued that in the presence of very large datasets it is useful to conduct studies of dataset selection methods aimed at reducing the bias and variance components of the prediction error, without having to re-use the training data. One method of reducing variance errors is the selection of a good set of predictive features. In this chapter, current methods of feature selection have been discussed and it has been argued that it is useful to conduct studies of feature selection methods that make use of the large amounts of available data to perform reliable measurements and validation of class-feature correlations. The next chapter presents a discussion of the research methods used for the studies in this thesis. The studies on feature selection methods are presented in chapter 5. The studies on training dataset selection are presented in chapters 6, 7, 8 and 9.

# Chapter 4
# Research Methods

> 'It is better to have an approximate answer to the right question than a precise answer to the wrong question which can be made…precise' (John Tukey)

The discussion in chapters 2 has made it clear that, first of all, it is necessary to conduct training dataset selection from large datasets for purposes of computational efficiency. Secondly, it is beneficial to study methods for selection of training data based on the characteristics of the instance space. Thirdly, the point has been made that the use of aggregate models has the potential to increase predictive accuracy since aggregate models are aimed at the reduction of the variance component of the prediction error. The use of training dataset selection methods aimed at the reduction of the bias and variance components of the prediction error should result in predictive models with a higher level of performance, compared to models created from data selected purely at random. The discussion of chapter 3 has made it clear that many samples should be used for the measurement and validation of the correlations for the dataset features in order to ensure reliable feature selection for large datasets.

The purpose of this chapter is to explain how methods that address the above issues were studied. Detailed discussion of the research questions and objectives, the central argument of the thesis and, the research paradigm that was followed, are given in sections 4.1, 4.2 and 4.3 respectively. The datasets used for the experiments and the sampling procedures used are discussed in sections 4.4 and 4.5 respectively. The data mining algorithms used in the experiments, the methods used to evaluate model performance and, the software used for the experiments are given in sections 4.6, 4.7 and 4.8 respectively. Section 4.9 gives a summary of the chapter.

## 4.1 Research questions and objectives

The discussions in chapters 2 and 3 led the author to pose the following main research question:

*What methods of training dataset selection can be used to obtain as much information as possible from large datasets while at the same time using training datasets of small sizes to create predictive models that have a high level of predictive performance?*

Based on the main research question and the literature review given in chapters 2 and 3, the primary objectives for conducting the research were to investigate promising methods for the following:

*(1) Reliable feature selection from large datasets using as much data as is feasible.*

*(2) Design of aggregate models which can make use of large amounts of training data while avoiding the problem of modeling phantom structure (i.e. structure that occurs purely due to chance as discussed in section 2.8.2).*

*(3) Design and selection of training datasets for base models aimed at increasing predictive accuracy through the reduction of bias and variance of the prediction error.*

*(4) Creation of a theoretical model that helps to explain the factors that affect the quality of selected features and the relationships between these factors.*

*(5) Creation of a theoretical model that helps to explain the factors that affect the performance of aggregate models and the relationships between these factors.*

## 4.2 The central argument for the thesis

The central argument of this thesis is that, for predictive data mining, it is possible to systematically select many dataset samples and employ different approaches (different from current practice) to feature selection, training dataset selection, and model construction. When a large amount of information in a large dataset is utilised in the modeling process, the resulting models will have a high level of predictive performance and should be more reliable.

Feature selection has been traditionally done based on a single randomly selected sample of the data. In the presence of very large amounts of data, many samples can be used in order to more reliably measure and validate the correlations between

the potential predictive features and the class (predicted) variable. The construction of base models that make up an aggregate model requires the use of a separate training dataset for each base model. It has been argued that syntactic diversity in the base models is a key factor in increasing aggregate model performance. For small datasets it is difficult to create sufficiently large training datasets that are also highly diverse. Breiman (1996) has attempted to achieve diversity through bootstrap sampling. With large amounts of data, it is easier to create diverse training datasets, since there is plenty of data to choose from. Through the use of boosting, Freund and Schapire (1997) have attempted to replicate the training instances that come from those regions of the instance space that are difficult to predict correctly. With large amounts of data, it is easier to obtain many instances that come from the difficult regions. Very large datasets provide far better coverage of the instance space, compared to small datasets. Examination of the structure of the instance space should lead to a better understanding of the prediction task at hand. This understanding should lead to better decisions for the sample composition of the training datasets for base models.

## 4.3 The research paradigm and methodology

The research paradigm used for this research is design science research as described by March and Smith (1995), Hevner et al (2004), Vaishnavi and Kuechler (2004/5) and Manson (2006). In this section, the design science research paradigm and methodology are briefly discussed. The design science research paradigm and the outputs of design science research are respectively presented in sections 4.3.1 and 4.3.2. Artifact evaluation and theory building, and the justification for adopting design science research for this thesis are respectively discussed in section 4.3.3 and 4.3.4. The different types of theories for data mining are discussed in section 4.3.5.

### 4.3.1 The design science research paradigm

The design science research paradigm originates from engineering and the physical sciences (March & Smith, 1995). Design science (Simon, 1996) and design science research (March & Smith, 1995) are concerned with the design and study of artifacts.

Hevner et al (2004) have provided the following definition for Information Systems artifacts:

> '.. innovations that define ideas, practices, technical capabilities, and products, through which the analysis, design, implementation, and use of information systems can be effectively and efficiently accomplished.'

Design science research involves two distinct steps as depicted in figure 4.1. In the first step, an artifact is created. In the second step, an analysis of the usage and performance of the artifact is conducted. The purpose of the analysis is to understand, explain, and possibly improve on one or more aspects of the artifact (Vaishnavi & Kuechler, 2004/5). According to Hevner et al (2004), in the context of information systems, artifacts may be models (abstractions and representations), methods (algorithms and practices) and instantiations (implemented and prototype systems). Design science research is a problem solving paradigm which seeks to create innovations in terms of ideas, practices, technical capabilities, and products. Through these innovations, the analysis, implementation, and usage of information systems can be effectively accomplished. Another view of design science research is due to March and Smith (1995). March and Smith (1995) have defined design science research and design science as activities aimed at the creation of things that serve human purposes. Design science and design science research are therefore technology-oriented and their outputs are assessed against criteria of value/utility.

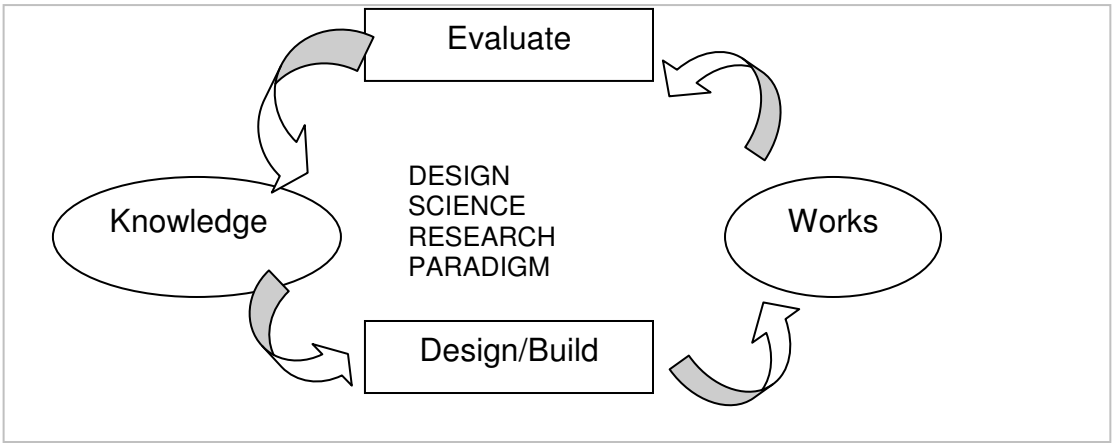


*Figure 4.1: A general model for generating knowledge in design science research (adopted from Vaishnavi & Keuchler (2004/5) and Manson (2006) )*

Manson (2006) has summarised these views by observing that design science research is a process of using knowledge to design and create useful artifacts, and then using rigorous methods to analyse why, or why not, a particular artifact is

effective. Figure 4.1 shows a general model for generating knowledge in design science research, and reflects Manson's (2006) observations.

### 4.3.2 The outputs of design science research

Table 4.1 gives a list of the outputs of design science research. Scientific research is about generating knowledge. In terms of generating new knowledge, for design science research new knowledge is generated in terms of the new constructs, new models, new methods (the how-to knowledge), and the better theories that arise out of the design and evaluation activities. Constructs are the core vocabulary that is used to express the concepts of a field. Knowledge is created when statements or propositions are made to express the relationships between various constructs of the field. Better theories, in terms of the models, will result if the models are rigorously tested in order to establish the existence of the relationships.

*Table 4.1: Outputs of design science research: Adapted from Vaishnavi & Kuechler (2004/5)*

| | Output | Description |
|---|---|---|
| 1 | constructs | Conceptual vocabulary of a domain. Constructs make up the language used to define and communicate problems and solutions. |
| 2 | models | A set of propositions or statements expressing relationships between constructs |
| 3 | Methods | a set of steps used to perform a task: **how-to knowledge** |
| 4 | Instantiations | Operationalisation of constructs, models and methods. Demonstration that the models and methods can be implemented in a working system. |
| 5 | **Better theories** | Artifact construction as analogous to experimental natural science |

### 4.3.3 Artifact evaluation and theory building

March and Smith (1995) have defined theories as 'deep, principled explanations of phenomena'. Cohen (1995) has argued that theories may also be 'propositions from which we can derive testable hypotheses'. Table 4.1 shows that one of the outputs of design science research should be 'better theories', that is, some improvements should be made to the existing theories of the field. Cohen (1995:ch.9) has provided guidelines for generalisation and theory building in Artificial Intelligence (AI) research. Cohen (1995) has stated that, for AI research there are six possible types of contributions as shown in figure 4.2. The cells 3,4,5,6 (P-S, P-G, E-S, E-G) in figure 4.2 represent research activities that result in the creation of new scientific theories. Cells 3 and 4 (P-S and P-G) represent the creation of predictive theories. Predictive

theories attempt to predict (hypothesize on) the behaviour of a specific system or a class of systems. According to Cohen (1995), system behaviour is typically predicted in terms of the features of the system architecture (structure), the features of the tasks that the system can perform, and the features of the environment in which the system operates. Cells 5 and 6 (E-S and E-G) represent the creation of explanatory theories. Explanatory theories attempt to explain the hypothesized behaviour of a specific system or class of systems.

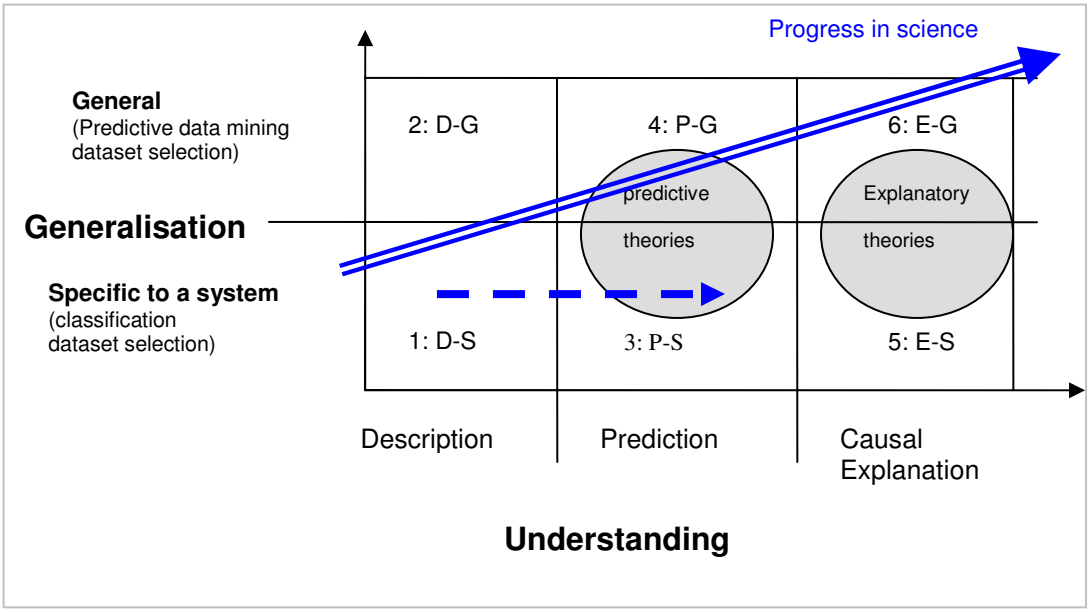


*Figure 4.2: Relationship between understanding, generalisation and scientific theories. Adapted from Cohen (1995)*

According to Cohen (1995) progress in science is gradually achieved by moving from descriptions of specific systems to providing causal explanations for systems in general as depicted in figure 4.2. Specific to design science research, March and Smith (1995) have observed that progress in design science is achieved when existing technologies are replaced by more effective ones. For the scope of this research, 'general' systems were viewed as systems for dataset selection for predictive data mining. A system for dataset selection for discriminative classification modeling was viewed as a 'specific' system as depicted in figure 4.2. The dashed line in figure 4.2 indicates the scope of scientific progress addressed in this thesis based on Cohen's (1995) definitions. The scope of design science progress claimed in this thesis is described in detail in chapter 11.

In the process of formulating predictive and explanatory theories, the Scientific Method of Peirce and Popper (Ngwenyama & Osei-Bryson, 2010; Oates, 2006) may

be followed for purposes of building theories based on empirical studies. This method involves observation, hypothesis generation, experiment design, and testing the validity of the hypotheses. Figure 4.3 shows the steps of the scientific method based on the discussion by Ngwenyama and Osei-Bryson (2010). Empirical observation involves the gathering of data/information about the phenomenon of interest. Hypothesis generation is when the researcher puts forward several hypotheses that could explain the phenomenon. Experiment design involves the design of experiments to test the logical consequence (validity) of the hypotheses. In the empirical testing step, the experiments are conducted in order to collect observations/data which is then analysed in order to establish whether or not the hypotheses are valid. The building of new theories arises from the outcomes of the empirical testing step. The empirical research reported in this thesis resulted in the formulation of a number of predictive theories which are presented in chapter 11. The scientific method was followed in the design and evaluation steps within the design science research paradigm.

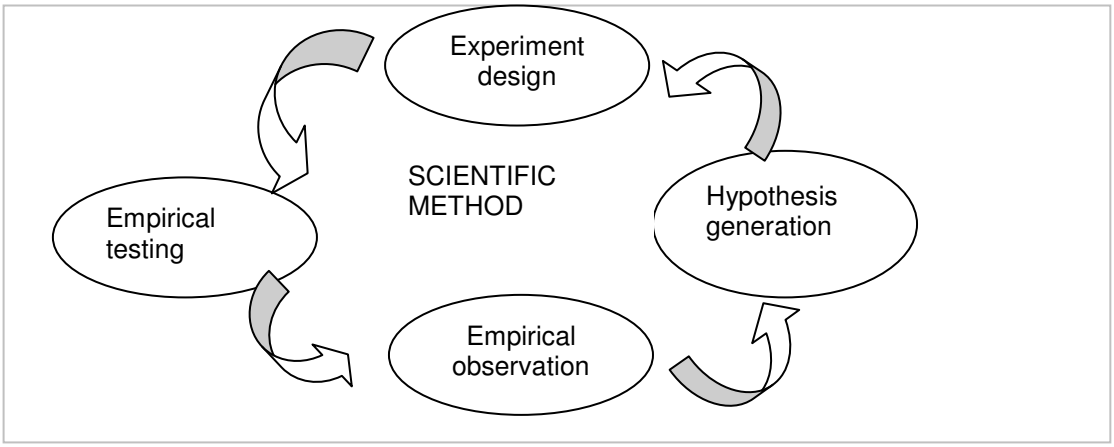


*Figure 4.3: Steps of the scientific method.*

### 4.3.4 Justification for adopting the design science research paradigm

This thesis is concerned with the investigation of methods for selecting features and training datasets from large amounts of data and the use of the selected data to create predictive models which can achieve a high level of accuracy and stability. The design and evaluation activities for the research are therefore the design and evaluation of feature selection methods, training set selection methods, and associated methods for model construction and testing. The use of design science

research enabled the author to systematically evaluate the hypothesised methods of feature selection, dataset selection, and model construction, and to systematically test hypothesised relationships between factors that affect the quality of selected features and training datasets. Based on existing literature, the author was able to construct models of known factors that affect the quality of selected features and training datasets, and to extend these models based on the results obtained from the experiments that were conducted for this research.

### 4.3.5 Theories for data mining

The foregoing discussion is aimed at the development of empirically derived theories for data mining. It was noted in chapter 1 that the parent fields of data mining are Computer Science and Statistics (Smyth, 2001). More recently, Olafsson et al (2008) have discussed the contributions by Operations Research to the field of data mining. While Statistics and Operations Research are founded on mathematical theories, in general for Computer Science, there are two types of possible theories: mathematical theories and empirical theories (Simon, 1996). Simon (1996) has observed that there are many aspects of computer systems that are so complex that there are no feasible mathematical theories that can be developed to describe their design and behaviour. Specific to machine learning, there are many mathematical theories that have been developed. However, Dietterich (1997) has observed that many problems in machine learning will only be solved through empirical studies, and not through mathematical formulations. The theories discussed in chapter 2, on sample complexity for inductive algorithms are a case in point. It was stated in chapter 2 that these theories provide unrealistic estimates for the sample complexity. Cohen (1995) has provided comprehensive guidelines on how to conduct empirical research in artificial intelligence and how to generate empirical theories from the empirical studies.

## 4.4 The datasets used in the experiments

The datasets used for the experiments were obtained from the UCI KDD Archive (Bay et al, 2000; Hettich & Bay, 1999), and the UCI Machine Learning Repository (Ascuncion & Newman, 2007; Blake & Merz, 1998). This section provides the motivation for the choice of datasets, brief descriptions of the datasets, past usage of the datasets, and pre-processing that was performed on two of the datasets. The

descriptive statistics for the selected datasets are provided in appendix C. The reasons for selecting the datasets for the experiments of this thesis are presented in section 4.4.1. Dataset pre-processing is discussed in sections 4.4.2 and 4.4.3.

### 4.4.1 Choice of datasets and past usage

Typical empirical studies on aggregate modelling for small datasets have been conducted using small numbers of datasets. The exception is Ali and Pazzani's (1996) studies where 29 datasets have been used. Table 4.2 provides some examples of studies where small datasets have been used. Experimental studies on dataset selection and aggregate modelling from large datasets have also been conducted using small numbers of datasets.

*Table 4.2: Examples of datasets used in data mining and machine learning studies*

| Author(s) | Nature / scope of study | Dataset description |
|---|---|---|
| Ooi et al (2007) | OVA classification and feature selection | 8 small datasets of sizes between 60 and 257 instances, and large number of dimensions ranging between 1,741 and 12,011 |
| Chawla et al (2001) | Dataset partitioning and aggregate modeling using massively parallel super computers | 4 small datasets of size less than 20,000 instances and **2 large datasets of** size 299,186 **and** 3.6 million instances |
| Hall et al (2000) | Dataset partitioning and aggregate modelling using massively parallel super computers | **4 very large datasets** of sizes 1.6, 3.2, 6.4 and 51 million instances |
| Chan and Stolfo (1998) | Dataset partitioning, sampling and aggregate modelling | **1 large dataset** for credit card fraud detection. 500,000 instances |
| Ho (1998) | Dataset partitioning with random feature subsets | 4 small datasets of sizes between 3,186 and 14,500 instances |
| Ali and Pazzani (1996) | Factors that affect performance improvements for aggregate models | 29 small datasets of sizes between 150 and 8,200 instances |
| Breiman (1996) | Bootstrap aggregation for classification and regression | 12 small datasets of sizes between 351 and 1,395 instances |
| Kwok and Carter (1990) | aggregate modeling for decision trees | 2 small datasets of sizes 446 and 5,516 instances |

The examples given in table 4.2 indicate that studies have been conducted using one, two or four large datasets. Studies on extremely large datasets with instances in excess of one million have been conducted using supercomputers with massively parallel architectures (Chawla et al, 2001; Hall et al, 2000). Based on the foregoing observations and the time and computational resources available to the author, a

decision was made to use two small datasets and two large datasets for the experiments for this thesis.

Table 4.3 gives the characteristics and past usage of the datasets used in the experiments for this thesis. The mushroom and abalone datasets (Ascuncion & Newman, 2007; Blake & Merz, 1998) are very commonly used in machine learning research. The wine quality dataset (Cortez et al, 2009; Ascuncion & Newman, 2007) has been used by Cortez et al (2009) for predictive modeling of wine quality. The mushroom dataset has originally been used for concept learning research by Schlimmer (1987) and Iba et al (1988). The mushroom dataset was selected for this research as the dataset which consists of only qualitative features, in order to study the behaviour of correlation measures for qualitative features. The abalone dataset has originally been used by Waugh (1995) for cascade-correlation. The original abalone dataset has 29 classes. Clark et al (1996) have created a three-class version of this dataset for their comparative study of artificial neural network algorithms. The three-class version of the abalone dataset was used for the experiments reported in this thesis. The wine quality dataset was selected for purposes of establishing whether the proposed training instance selection methods can also be applied to small datasets. A second reason for selecting the abalone and wine quality datasets was because these datasets have low levels of classification accuracy and can therefore be used to demonstrate increases in predictive performance (Cohen, 1995).

The two large datasets that were used for the experiments are forest cover type and KDD Cup 1999 (Bay et al, 2000; Hettich & Bay, 1999). These two datasets were chosen for the experiments because they are large datasets, and have large numbers of features. Tables 4.3, 4.4 and 4.5 show the important statistics for these datasets. The forest cover type dataset is a good example of data mining for a scientific application. This dataset consists of data describing the forest cover type for each of 581,012 forest cells, each measuring 30x30 meters. The prediction task for the forest cover type dataset is to predict one of seven forest cover types based on the soil type, wilderness area type, elevation (altitude) and other variables. Blackard (1998) has used this dataset to study the differences in predictive performance between artificial neural networks and discriminant analysis. The KDD Cup 1999 dataset is a typical example of data for forensic data mining. This dataset is a common benchmark for the evaluation of computer network intrusion detection

systems (IDS). The dataset consists of a wide variety of network intrusions, simulated for a military computer network environment.

*Table 4.3: The datasets used for the experiments*

| Dataset | Description | | Past usage |
|---|---|---|---|
| | Size | Features & classes | |
| Forest cover type | 581,012 | 54 features<br>10 continuous, 44 binary, 7 classes | Comparison of ANNs and discriminant analysis (Blackard, 1998) |
| KDD Cup 1999 Training dataset (10% version) | 494,022 | 41 features<br>34 continuous, 7 qualitative, 23 classes | Intrusion detection (Stolfo et al, 2000) |
| KDD Cup 1999 Test dataset | 311,029 | 41 features<br>34 continuous, 7 qualitative, 40 classes | |
| Wine quality (white) | 4,898 | 11 continuous-values features,<br>7 classes | Prediction of wine quality scores assigned by wine tasters. (Cortez et al, 2009) |
| Abalone (3 class) | 4,177 | features: 8 features<br>7 continuous, 1 qualitative, 3 classes | Prediction of the age of abalone |
| Mushroom | 8,146 | 22 qualitative features<br>2 classes | Prediction of edibility of mushrooms |

*Table 4.4: Class counts for the forest cover type dataset*

| Class | Type of forest cover | Count |
|---|---|---|
| 1 | Spruce / Fir | 211,840 |
| 2 | Lodgepole pine | 283,301 |
| 3 | Ponderosa pine | 35,754 |
| 4 | Cottonwood / Willow | 2,747 |
| 5 | Aspen | 9,493 |
| 6 | Douglas - fir | 17,367 |
| 7 | Krummholz | 20,510 |
| **TOTAL** | | **581,012** |

The dataset was provided by the USA DARPA and MIT Lincoln Labs (Lee et al, 2002), and was later pre-processed for the KDD Cup 1999 competition by the Columbia IDS Group (Stolfo et al, 2000). Two versions of this dataset are provided at the UCI KDD archive. The smaller version, which consists of ten percent of the instances of the original version, was used for the experiments. Four main categories of attacks are present in the dataset: denial-of-service (DOS), unauthorized access from a remote machine (R2L), unauthorized access to super-user privileges (U2R), and probing attacks (PROBE) (Laskov et al, 2005; Lee & Stolfo, 2001; Stolfo et al, 2000).

*Table 4.5: Class counts for the KDD Cup 1999 training (10% version) and test sets*

| | Class count | | | Class count | |
|---|---|---|---|---|---|
| Attack Type | Training set | Test set | AttackType | Training set | Test set |
| apache2 | | 794 | portsweep | 1,040 | 354 |
| back | 2,203 | 1,098 | processtable | | 759 |
| buffer_overflow | 30 | 22 | ps | | 16 |
| ftp_write | 8 | 3 | rootkit | 10 | 13 |
| guess_passwd | 53 | 4,367 | saint | | 736 |
| httptunnel | | 158 | satan | 1,589 | 1,633 |
| imap | 12 | 1 | sendmail | | 17 |
| ipsweep | 1,247 | 306 | smurf | 280,790 | 16,4091 |
| land | 21 | 9 | snmpgetattack | | 7,741 |
| loadmodule | 9 | 2 | snmpguess | | 2,406 |
| mailbomb | | 5,000 | spy | 2 | |
| mscan | | 1,053 | sqlattack | | 2 |
| multihop | 7 | 18 | teardrop | 979 | 12 |
| named | | 17 | udpstorm | | 2 |
| neptune | 107,201 | 58,001 | warezclient | 1,020 | |
| nmap | 231 | 84 | warezmaster | 20 | 1,602 |
| normal | 97,278 | 60,593 | worm | | 2 |
| perl | 3 | 2 | xlock | | 9 |
| phf | 4 | 2 | xsnoop | | 4 |
| pod | 264 | 87 | xterm | | 13 |
| | | | **TOTALS** | **494,021** | **311,029** |

## 4.4.2 Dataset pre-processing to balance class distributions

The KDD Cup 1999 dataset is not amenable to classifier construction without pre-processing (Laskov et al 2005; Leung & Leckie, 2005). Laskov et al (2005) have observed that the KDD Cup 1999 dataset suffers from two major flaws in the distribution of the classes in the dataset. First of all, approximately 80% of instances correspond to attacks. Secondly, the distribution of the attack instances is highly unbalanced. Laskov et al (2005) have observed that *Probes* and *DOS* attacks dominate the class distribution, while more dangerous attacks such as *phf* and *imap* are severely under-represented. Researchers who have used the KDD Cup 1999 dataset (e.g. Shin & Lee, 2006; Laskov et al, 2005; Leung & Lecki, 2005) have typically pre-processed the dataset to balance the distribution of the attack types and service types, and to reduce the number of instances for attacks in comparison to normal connections. Laskov et al (2005), for example, have reduced the number of attack instances to five percent (5%).

*Table 4.6: Reduction of the over-representation of (service, attack type) values in the KDD Cup 1999 training and test datasets*

| Dataset | Service name | Class name | Instance type | Count before | Count after |
|---|---|---|---|---|---|
| Training set | private | neptune | attack | 101,317 | 500 |
| | ecr_i | smurf | attack | 280,790 | 1,000 |
| | http | normal | normal | 61,887 | 5,000 |
| | smtp | normal | normal | 9,598 | 5,000 |
| Test set | private | neptune | attack | 54,739 | 500 |
| | private | smurf | attack | 164,091 | 1,000 |
| | private | snmpgetattack | attack | 7,733 | 2,500 |
| | smtp | mailbomb | attack | 5,000 | 2,500 |
| | private | normal | normal | 12,808 | 2,500 |

Two approaches have been used by researchers to construct classification models from the KDD Cup 1999 dataset. For the first approach, the attack types that appear in the data are used as the classes (Laskov et al, 2005; Lee & Stolfo, 2000; Fan et al, 2000). For the second approach, the main categories: NORMAL, DOS, PROBE, R2L and U2R are used as the classes (Shin & Lee, 2006). The main problem with using the first approach is that there are attack types that are severely under-represented as can be seen in table 4.6. Secondly, there are attack types that appear in the test set but not in the training set. The problem of under-representation is slightly reduced in the second approach. The problem of classes which appear in the test set and not the training set is partially eliminated when the second approach is used, since all such attacks belong to the R2L category.

For the experiments reported in this thesis, pre-processing of the dataset was done as follows. The instances for the *(service, attack type)* values that are severely over-represented were reduced as shown in table 4.6. The objective of the reduction was to ensure that the frequency of attacks for each over-represented attack type is reduced to make that frequency comparable to the other attack types for that service. The reduction was achieved using sequential random sampling of the instances that are over-represented. The training and test datasets were further pre-processed to add a new class variable with values NORMAL, DOS, PROBE, R2L and U2R. Table 4.7 shows the resulting attack type and class distributions after the pre-processing, for the dataset used for training. The test dataset was further pre-processed to remove all attack types that do not appear in the training data. This was motivated by the following observations as stated by Lee and Stolfo (2000).

The two main intrusion detection techniques are *misuse detection* and *anomaly detection*. Misuse detection systems use patterns of well known attacks to identify

known intrusions. On the other hand, anomaly detection systems detect and report activities that significantly differ from established normal usage profiles (Lee & Stolfo 2000). Since classification modeling is based on inductive learning, classification models created for intrusion detection systems should be created for misuse detection. For this reason, attack types that do not appear in the training data were removed from the test data. Table 4.8 shows the resulting class distribution of the test dataset after this phase of pre-processing. The entries for TestA in column 7 of table 4.8 indicate the number of instances that were removed because the attack type does not appear in the training data.

*Table 4.7: Class counts for the final version of the KDD Cup 1999 training dataset*

| Class | Type of connection | AttackType | AttackType count | Class count |
|---|---|---|---|---|
| DOS | Denial of service | back | 2,203 | |
| | | land | 21 | |
| | | neptune | 6,384 | |
| | | pod | 264 | |
| | | smurf | 1,000 | |
| | | teardrop | 979 | 10,851 |
| NORMAL | normal | normal | 35,794 | 35,794 |
| PROBE | Probing prior to attack | ipsweep | 1247 | |
| | | nmap | 231 | |
| | | portsweep | 1,040 | |
| | | satan | 1,589 | 4,107 |
| R2L | Unauthorised access from a remote machine | ftp_write | 8 | |
| | | guess_passwd | 53 | |
| | | imap | 12 | |
| | | multihop | 7 | |
| | | phf | 4 | |
| | | spy | 2 | |
| | | warezclient | 1,020 | |
| | | warezmaster | 20 | 1,126 |
| U2R | Unauthorised access to superuser privileges | buffer_overflow | 30 | |
| | | loadmodule | 9 | |
| | | perl | 3 | |
| | | rootkit | 10 | 52 |
| **TOTALS** | | | **51,930** | **51,930** |

*Table 4.8: Class counts for the final version of the KDD Cup 1999 test dataset*

| Class | AttackType | AttackType count | Class count | Class | AttackType | Attack count | Class count |
|---|---|---|---|---|---|---|---|
| DOS | apache2 | 794 | 10023 | R2L | httptunnel | TestA: 158 TestB: 0 | TestA: 11,114 TestB: 5,995 |
| | back | 1,098 | | | imap | 1 | |
| | land | 9 | | | multihop | 18 | |
| | mailbomb | 2,500 | | | named | TestA: 17 TestB: 0 | |
| | neptune | 3,762 | | | phf | 2 | |
| | pod | 87 | | | sendmail | TestA: 17 TestB: 0 | |
| | processtable | 759 | | | snmpgetattack | TestA : 2508 TestB: 0 | |
| | smurf | 1,000 | | | snmpguess | TestA: 2406 TestB: 0 | |
| | teardrop | 12 | | | warezmaster | 1,602 | |
| | udpstorm | 2 | | | worm | 2 | |
| NORMAL | normal | 50,285 | 50285 | | xlock | TestA: 9 TestB: 0 | |
| PROBE | ipsweep | 306 | 4166 | | xsnoop | TestA: 4 TestB: 0 | |
| | mscan | 1,053 | | U2R | buffer_overflow | 22 | 70 |
| | nmap | 84 | | | loadmodule | 2 | |
| | portsweep | 354 | | | perl | 2 | |
| | saint | 736 | | | ps | 16 | |
| | satan | 1,633 | | | rootkit | 13 | |
| R2L | ftp_write | 3 | | | sqlattack | 2 | |
| | guess_passwd | 4,367 | | | xterm | 13 | |
| | | | | **TOTALS (TestA)** | | **75,658** | **75,658** |
| | | | | **TOTALS (TestB)** | | **70,539** | **70,539** |

## 4.4.3 Dataset pre-processing to normalise feature values

The KDD Cup 1999 dataset contains features from various numeric-valued domains. Table 4.9 shows a selected sample of features as well as the minimum and maximum values of the features for the KDD Cup 1999 dataset. As can be seen in table 4.9, the KDD Cup 1999 dataset has features with a narrow value range (e.g. [0,1] for *DstHostSrvSerrorRate*) as well as features with a very wide value range (e.g. [0, 693375640] for *SrcBytes*). K-Nearest neighbour (KNN) is one of the classification algorithms that were used in the experiments. The KNN algorithm computes distances between instances using a distance measure from the class of the Minkowski norms (Doherty et al, 2007) of which the Euclidean distance measure is the most common. The computation of the Euclidean distance using features from very widely varying ranges of values such as found in the KDD Cup 1999 dataset will result in the large-valued features dominating the result of the computed distance, and so masking the effects of the small-valued features.

*Table 4.9: Range of values for features in the KDD Cup 1999 dataset*

| Feature | Minimum value | Maximum value |
|---|---|---|
| NumCompromised | 0 | 884 |
| WrongFragment | 0 | 3 |
| DstHostSrvSerrorRate | 0 | 1 |
| Hot | 0 | 30 |
| DstHostSerrorRate | 0 | 1 |
| NumRoot | 0 | 993 |
| Counted | 0 | 511 |
| DstBytes | 0 | 5,155,468 |
| **SrcBytes** | **0** | **693,375,640** |
| SrvCount | 0 | 511 |
| NumFailedLogins | 0 | 5 |
| NumFileCreations | 0 | 28 |
| Duration | 0 | 58,329 |

Doherty et al (2007) have conducted experiments which show that normalisation of data values for a dataset may eliminate this problem. For this reason, the numeric-valued features of the KDD Cup 1999 dataset were normalised in the pre-processing step for the KNN algorithm. Secondly, the normalised values were mapped into the range [0, 1000] to avoid the effects of rounding when fractional values are multiplied together.

## 4.5 Sampling methods

All the experiments reported in this thesis involved the use of simple random sampling. Simple random sampling is the process of selecting a sample of the population units while giving every member of the population an equal chance of being selected (Rao, 2000). Simple random sampling may be done with replacement (SRSWR) or without replacement (SRSWOR). For SRSWOR, every population unit gets only one chance of being considered for selection. For SRSWR, every population unit gets many chances of being considered for selection. Sequential random sampling, described in the next section, was used to implement both SRWR and SRWOR for the large datasets used in the experiments. Sequential random sampling is discussed in section 4.5.1. The shuffling of data prior to sampling is discussed in section 4.5.2.

### 4.5.1 Sequential random sampling

Olken and Rotem (1995) and Olken (1993) have studied the use of sequential random sampling from databases. For sequential random sampling, the problem is to draw a random sample of size *n* without replacement, from a file containing *N* records. The simplest sequential random sampling method is due to Fan et al (1962) and Jones (1962), and proceeds as follows: An independent uniform random variate from the uniform interval [0,1] is generated for each record in the file to determine whether the record should be included in the sample. If $n_t$ records have already been chosen from among the first *t* records in the file, the *(t+1)*$^{st}$ record is chosen with probability $(RQ_{size} / RM_{size})$, where $RQ_{size} = (n - n_t)$ is the number of records that still need to be chosen for the sample, and $RM_{size} = (N - t)$ is the number of records in the file still to be processed. Olken (1993) has used these methods to study database sampling.

### 4.5.2 Obtaining random samples from datasets

The records for each of the large datasets used in the experiments were randomised (shuffled) prior to sampling. The reason for shuffling the data was to remove any possible ordering in the dataset and to maximise the randomness of the order in which the instances appear. Sequential random sampling was then used to achieve simple random sampling, either from the whole dataset or from partitions of the dataset. In order to create bootstrap samples, the sequential random sampling procedure was repeated several times on the dataset, with a different random seed for each iteration. The shuffling and sampling from the datasets were implemented using stored procedures in a Microsoft SQL Server database.

## 4.6 The data mining algorithms used in the experiments

The two classification algorithms used for the experiments are decision tree for classification and K-Nearest Neighbour (KNN) classification. Decision tree classification (Quinlan, 1993; Quinlan, 1986; Breiman et al, 1984), which constructs classification models, has the desirable property that it attempts to identify the most relevant features. The KNN algorithm (Cover & Hart, 1967) is very different from the

decision tree algorithms, as it does not perform feature selection of any kind, and therefore benefits the most from feature selection. Wu et al (2008) have reported that the decision tree algorithms as implemented in C4.5 (Quinlan, 1993; Quinlan, 1986) and CART (Breiman et al, 1984) as well as the KNN classification algorithm (Cover & Hart, 1967), are among the top ten algorithms used in data mining research. This section provides a brief description of the classification tree and KNN classification algorithms. Classification tree algorithms are discussed in section 4.6.1. KNN classification is discussed in section 4.6.2.

### 4.6.1 Classification trees

A classification tree algorithm creates a tree-structured model for the prediction of a qualitative variable called the class variable. In the classification tree model each leaf node provides information about the class to be assigned to instances that fall in that node. A classification tree algorithm recursively partitions a set of training data, using one predictive feature at a time, to create training dataset partitions. A classification tree is constructed, along with the partitioning process, based on the generated training dataset partitions. The heuristic used to guide the partitioning process uses a *class impurity* measure. At each decision point (for partitioning), all remaining features are evaluated. The feature that produces the partitions with the lowest *class impurity* is selected for partitioning. The selected feature then becomes the test for the decision/classification tree node with its values labeling the branches of the node. Commonly used class impurity measures are the chi-square (CHAID) criterion (Giudici, 2003), the two-ing criterion (Breiman et al, 1984), the Gini index of diversity (Breiman et al, 1984), and the entropy function (Quinlan, 1986).

The partitioning process should ideally stop when each partition is pure, that is, it consists of training instances all of the same class. In practice, however, pruning methods are used to halt the partitioning when it is no longer statistically valid to continue (Quinlan, 1993; Breiman et al, 1984). Breiman et al (1984) have observed that the tree growing procedures result in trees that are much larger than the data warrant. For example, if splitting is carried out to the point where each terminal node contains only one data case, then each node is classified by the case it contains, and the error on the training data is zero. This is an extreme case of overfitting which was discussed in chapter 2. On the other hand, when a tree is too small, then useful classification information in the training data has been ignored. This results in a high

rate of classification error on the training data and a high predictive error rate on future instances. To determine the optimally-sized tree, a three step procedure is used (Osei-Bryson, 2004; Breiman et al, 1984). The first step is to grow a tree that is as large as possible. In the second step, the tree is pruned upward from the leaf nodes until the root is reached. In the third step, an independent sample of test data is used to estimate the predictive accuracy of all the pruned trees. The tree with the highest accuracy on the test data is selected as the optimally-sized tree. Optimisation methods from Operations Research have also been proposed for the selection of the optimal-size classification tree based on multiple objectives (Osei-Bryson, 2004). For the final classification tree that is used for classification, each leaf node has an assigned posterior probability for each class. In the prediction process, when a query instance lands at a given leaf node, the class with the highest probability at that node is predicted for the query instance (Osei-Bryson, 2004; Quinlan, 2004).

### 4.6.2 K-Nearest Neighbour classification

The K-Nearest Neighbour (KNN) classification algorithm originates from the field of statistical pattern recognition (Cover & Hart, 1967). The *inductive bias* of the KNN algorithm corresponds to an assumption that the classification of an instance $\mathbf{x}_q$, will be most similar to the classification of other instances that are nearby in terms of Euclidean distance. K-nearest neighbour classification uses a lazy algorithm which only constructs a classifier in the form of a target function, only when a new instance for classification is presented. The target function may be either discrete or real valued. If the target function is discrete valued then it is of the form $f : R^d \rightarrow C$, where *d* is the number of predictive features and *C* is the finite set $\{c_1,...,c_k\}$ of the classes in the training data. For the simplest implementation of the KNN algorithm, the target function is estimated by computing a score for each class and returning that class that most frequently occurs among the K-nearest instances, based on the Euclidean distance. The score computed by the KNN algorithm is also the posterior probability $P_r(c_i \mid \mathbf{x}_q)$ that the query instance $\mathbf{x}_q$ belongs to class $c_i$. The computation of the Euclidean distance between query (test) instance $\mathbf{x}_q$ and training instance $\boldsymbol{x}$ is

$$dist(\boldsymbol{x}, \boldsymbol{x_q}) = \left( \sum_{i=1}^{d} (x_i - x_{qi})^2 \right)^{\frac{1}{2}} \qquad (4.1)$$

where *d* is the number of predictive features for the instance space. Since many datasets have qualitative features, special treatment is needed for the qualitative nominal and qualitative ordinal values when computing Euclidean distance. A common approach is to define the quantity $(x_i - x_{qi})$ for qualitative (nominal and ordinal) values as follows (Mitchell, 1997):

$$(x_i - x_{qi}) = \begin{cases} 0 \text{ if the qualitative values are identical} \\ 1 \text{ if the qualitative values are different} \end{cases} \quad (4.2)$$

## 4.7 Measures of model performance

Evaluation is a crucial part of design science research. The measures for evaluating predictive model performance are discussed in section 4.7.1. Statistical methods for comparing two models on performance are discussed in section 4.7.2. ROC and lift chart analysis are discussed in section 4.7.3.

### 4.7.1 Measures of predictive performance

It was stated in chapter 1 that statisticians have invented effective methods of model construction, validation and testing for small datasets. Model validation and testing, using small amounts of data, has typically been done in the past using cross validation, the hold out method, or the bootstrap method (Mitchell, 1997; Hand, 1997; Moore & Lee, 1994). These methods were discussed in chapter 2. Even though the predictive performance of a model is very commonly reported in terms of predictive accuracy, especially in machine learning literature, various measures exist for more detailed analysis of the predictive capabilities of a model (Giudici, 2003; Hand et al, 2001; Hand, 1997). By generating a confusion matrix, performance measures can be computed for a given classification model. Table 4.9 shows a theoretical confusion matrix for a 2-class problem with two class labels *positive* and *negative* (Giudici, 2003; Hand, 1997). For a given validation dataset or test dataset, the value *TP* represents the number of positive instances that are correctly predicted as positive instances. The value *FN* represents the number of positive instances that are incorrectly predicted as negative instances. The value *FP* represents the number of

negative instances that are incorrectly predicted as positive instances. The value *TN* represents the number of negative instances that are correctly predicted as negative instances. In general, a confusion matrix can be created for any *k*-class (*k* > *1*) classification model. From the *TP, FN, FP* and *TN* values, various performance measures can be derived (Giudici, 2003; Kubat & Matwin, 1997; Hand, 1997).

Table 4.10 gives the definitions and computation of the performance measures for a 2-class model. These performance measures provide useful information which can be used to compare classification models and to select the model that has the best predictive performance on the test data. The counts *FN* and *FP* represent levels of *class confusion*. The value *FN* represents the level to which instances of the *positive* class are mis-classified as *negative* instances and *FP* represents the level to which instances of the *negative* class are mis-classified by the model as positive instances.

*Table 4.9: Theoretical confusion matrix for a 2-class model*

| **Actual class** | **Predicted class** | | **Totals** |
|---|---|---|---|
| | **positive** | **negative** | |
| positive | $TP$ | $FN$ | *Pos = TP + FN* |
| negative | $FP$ | $TN$ | *Neg = FP + TN* |
| **Totals** | *TP + FP* | *FN + TN* | ***Pos + Neg*** |

*Table 4.10: Measures of performance derived from a confusion matrix*

| **Measure** | | | **Computation (in terms of table 4.9)** |
|---|---|---|---|
| **Name** | **Description** | **symbol** | |
| Error | error rate | *error* | *(FN + FP) / (Pos + Neg)* |
| Accuracy | Accuracy | *accuracy* | *(TP + TN) / (Pos + Neg)* |
| Sensitivity | True positive rate | *TPRATE* | *TP / (TP + FN)* |
| Specificity | True negative rate | *TNRATE* | *TN / (FP + TN)* |
| Precision | Correct positive prediction rate | *Precision* | *TP / (TP + FP)* |
| Type I error rate | False negative rate | *FNRATE* | *FN / (TP + FN)* |
| Type II error rate | False positive rate | *FPRATE* | *FP / (FP + TN)* |
| Y rate | Positive prediction rate | *YRATE* | *(TP + FP) / (Pos + Neg)* |

The concepts of *positive instances* and *negative instances* for *k*-class prediction tasks were interpreted as follows in this thesis. Each class $c_i$ was treated as the positive class in contrast to all the other *k-1* classes which were treated as the

negative classes. This resulted in the creation and analysis of *k* confusion matrices with one 2 x 2 confusion matrix for each (positive) class.

The *error* and *accuracy* measures have a straight forward interpretation. In this thesis, the accuracy (rather than the *error*) is reported for all experiments on predictive performance. For a *2*-class problem, the *sensitivity* or *true positive rate* is the error rate on the test instances that belong to the *positive* class. For *2*-class problems, the *specificity* or *true negative rate* is the error rate on the test instances that belong to the *negative* class. The *false negative rate (type I error rate)* is the rate at which a model fails to classify *positive* instances as *positive*. The *false positive rate* (*type II error rate)* is the rate at which a model fails to classify *negative* instances as *negative*. The *YRATE* is used for lift analysis as discussed in section 4.7.3.

## 4.7.2 Statistical test to compare model performance

For purposes of comparing the performance of two predictive models $M_A$ and $M_B$, a common approach is to establish the performance of each model on several test problems and compute the values of selected measures, or, all of the measures presented in the last section. Most commonly, in machine learning, the predictive accuracy or error rate are computed. Statistical tests are then used to compare the values of the measures on the test problems in order to establish if one model provides a higher level of predictive performance. Suppose that models $M_A$ and $M_B$ are each tested on a set of *n* problems, $PSet_A = \{problem_{A1},..., problem_{An}\}$ for model $M_A$ and $PSet_B = \{problem_{B1},..., problem_{Bn}\}$ for model $M_B$. For statistical testing, when the sample size *n* is large (*n* ≥ *30*), the *Z* test for normal distributions is used to compare the mean values of the performance measures. When *n* is small (*5*≤ *n<30*), Student's t test is used to compare the mean values (Cohen, 1995).

There are two types of t-tests for comparing sample means. For the first t-test, called the *two sample t test* (Cohen, 1995), the problem sets $PSet_A$ and $PSet_B$ are different. For the second t-test, called the *paired sample t test* (Cohen, 1995), the problem sets $PSet_A$ and $PSet_B$ are identical. The models $M_A$ and $M_B$ are each tested on each problem, $problem_{Ai}$, and the statistical test is based on the

difference in performance on each of the test problems. The *paired sample t-test* has more statistical power than the *two sample t-test* since it controls for (minimises) the variance due to the test problems (Cohen, 1995). The *paired sample t-test* was used for all the experiments in this thesis to compare model performance. In order to establish whether model $M_A$ provides a higher level of predictive performance compared to model $M_B$, the following null hypothesis $H_0$ and *two-tail* alternative hypothesis $H_a$ were tested:

$$H_0 : \mu_\delta = 0, H_a : \mu_\delta \neq 0 \tag{4.3}$$

where $\mu_A$ and $\mu_B$ represent the hypothesised mean values for one of the performance measures presented in the last section and $\mu_\delta = \mu_A - \mu_B$ represents the mean difference.

The F-test for variances (Cohen, 1995) was used in this thesis to compare the single and aggregate models in terms of variability of predictive performance. Cohen (1995: pg 205) has advised that comparison of performance variance for two models can be used to establish whether one model exhibits more erratic (or more coherent) behaviour compared with the other model. When two models have equal mean predictive performance then the model with more coherent performance should be preferred (Cohen, 1995). For the F-test of variances, the null hypothesis $H_0$ is that there is no significant difference in the performance variances of both models.

When an experiment is conducted, the probability of obtaining a particular sample result given the null hypothesis $H_0$ is called the *p value*. There are two methods of conducting statistical inference with p values. With the first, more traditional method, the researcher decides on the level of significance at which the null hypothesis will be rejected. Conventionally, a significance level of 0.05 (p = 0.05) is used. If the *p* value for a test is less than 0.05, then the null hypothesis is rejected (Montgomery et al, 2004; Cohen, 1995). For the second method, various levels of the *p* value are used to determine the outcome of the test, as shown in table 4.11 (Stirling, 2008). The second method of interpreting *p* values was adopted for the experiments of this thesis.

*Table 4.11: Interpretation of p values for statistical tests*

| ***p* value** | **Interpretation** |
|---|---|
| $p < 0.01$ | Strong evidence for the rejection of $H_0$ |
| $0.01 < p \leq 0.05$ | Moderate evidence for the rejection of $H_0$ |
| $0.05 < p \leq 0.1$ | Marginal or weak evidence for the rejection of $H_0$ |
| $p > 0.1$ | No evidence to support the rejection of $H_0$ |

### 4.7.3 Analysis of performance using ROC curves and lift charts

Receiver Operating Characteristic (ROC) curves and lift charts are commonly used as graphic representations of predictive model performance for 2-class prediction tasks (Giudici & Figini, 2009; Witten & Frank, 2005; Giudici, 2003; Berry & Linoff, 2000). A probabilistic classification model will typically assign a class and a score for the class. Most commonly, the score is the posterior probability that a test instance belongs to the predicted class (Giudici & Figini, 2009; Witten & Frank, 2005; Giudici, 2003; Berry & Linoff, 2000). ROC analysis and lift analysis are concerned with the selection of the model with the optimal performance based on the cut-off point $\lambda$ that is used to decide when an instance should be declared positive or negative. A cut-off point is the score value $conf(\mathbf{x})$ for which $conf(\mathbf{x}) \geq \lambda$ implies that the predicted class for instance $\mathbf{x}$ is the positive class. ROC and lift analysis can also be used to determine which of two models provides a higher level of predictive performance as discussed below.

The Receiver Operating Characteristic (ROC) curve construct originates from signal detection applications where there is a signal transmitter and a signal receiver for a given (possibly noisy) transmission channel. A ROC curve is used to specify the relationship between the hit rate (correct detection) and the miss rate (false alarm rate) for the signal receiver (Witten & Frank, 2005). For classification modeling, a ROC curve is created using the information in a 2-class confusion matrix. More precisely, a ROC curve is a plot on a 2-dimensional Cartesian plane with the *x* and *y* values defined as (Vuk & Curk, 2006; Fawcett, 2001, 2004, 2006; Ferri et al, 2003; Hand & Till, 2001):

$$x = FPRATE(\lambda), y = TPRATE(\lambda) \tag{4.4}$$

where $FPRATE(\lambda)$ and $TPRATE(\lambda)$ are respectively the false positive and true positive rates obtained when the cut-off value of $\lambda$ is used. In order to understand the purpose of ROC analysis for classification modeling, it is useful to make a distinction between a *discrete classifier* and a *probabilistic classifier*. A discrete classifier assigns a class label to a test (or query) instance for a fixed threshold value (Fawcett, 2001, 2004, 2006). A probabilistic classifier on the other hand has the ability to assign a class label and a (probabilistic) score to a test (or query) instance for different threshold values. Stated differently, a probabilistic classifier operates in ROC space (Fawcett, 2001, 2004, 2006) which is the 2-dimensional plane defined by equation (4.4). A discrete classifier corresponds to exactly one point in the ROC space of a probabilistic classifier.

Figure 4.5 shows the Cartesian plane for the ROC space with a ROC curve example. A ROC curve represents relative tradeoffs between the benefits (true positives) and the costs (false positives) of using a given probabilistic classifier (Fawcett, 2006).

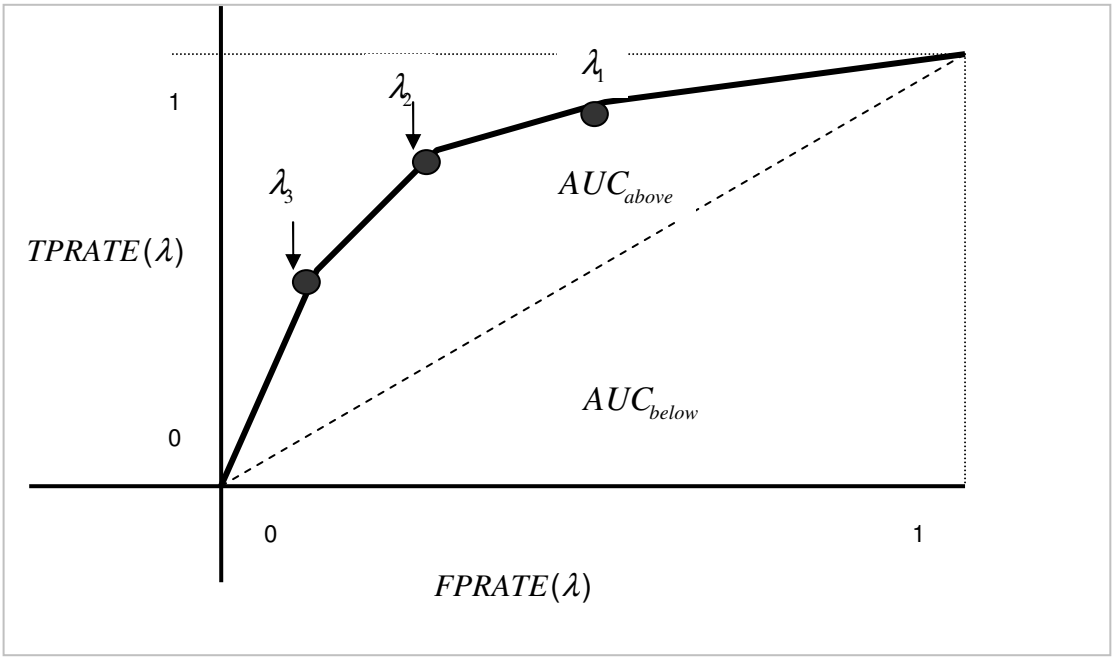


*Figure 4.5: ROC space and AUC*

For a given probabilistic classifier, each cut-off value of $\lambda$ corresponds to a single point in the ROC space as defined in equation (4.4). The ROC curve joins these points for $-\infty < \lambda < \infty$. The point (0,0) represents a classifier which never gives a positive prediction. The point (1,1) represents a classifier which always gives a

positive prediction. The point (0,1) represents a perfect classifier which never issues incorrect predictions. For the ROC curve example shown in figure 4.5, the relationship between the cut-off values is: $\lambda_3 > \lambda_2 > \lambda_1$, that is, the higher the cut-off value, the lower the FPRATE and TPRATE. A 45 degree line is normally plotted on the ROC plane to represent the ROC curve for classification in the absence of a model (random guessing). Any ROC point which lies below the 45 degree line represents a model which performs worse than random guessing.

An important statistic provided by the ROC curve is the Area Under the ROC curve (AUC). The AUC is the area between the x-axis, y-axis and the ROC curve (Fawcett 2001, 2004, 2006; Vuk & Curk, 2006; Ferri et al, 2003). This is the sum of the areas labelled $AUC_{above}$ and $AUC_{below}$ in figure 4.5. The area $AUC_{below}$ has a fixed value of 0.5. Fawcett (2006) has observed that the area $AUC_{above}$ is related to the Gini concentration coefficient (Breiman et al, 1984) as:

$$Gini = 2 \text{x} AUC_{above} \tag{4.5}$$

Hand and Till (2001) have observed that the total area under the curve is related to the Gini concentration coefficient as:

$$Gini + 1 = 2 \text{x} (AUC_{below} + AUC_{above}) \tag{4.6}$$

The definition of the Gini concentration coefficient is given in appendix B. The AUC is equivalent to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance (Fawcett, 2006). The AUC is also equivalent to the statistical Wilcoxon test of ranks (Fawcett, 2006; Hand & Till, 2001; Hanley & McNeil, 1982). Given two classifiers, the classifier with the larger AUC value provides a higher level of predictive performance. When the ROC curves of the two classifiers lie above the 45 degree line the performance difference is determined by the $AUC_{above}$ area. For this reason, all discussions of the AUC provided in chapter 9 refer to the $AUC_{above}$ area.

Two-class ROC analysis is concerned with the computation of the AUC, which is computed in a straight-forward manner by calculating the area under the ROC curve

in the 2-diminsional Cartesian plane defined by equation (4.4). For *k*-class ($k > 2$) prediction tasks, ROC analysis is concerned with the computation of the Volume Under the ROC Surface (VUS). Computation and visualisation of the VUS is a non-trivial task. Two surrogate measures for the VUS, which have been proposed by Hand and Till (2001) and Provost and Domingos (2001) are discussed in chapter 9 of this thesis. The ROC (VUS) analysis results for the models studied in the experiments for this thesis are also presented in chapter 9.

The lift chart construct originates from the domain of predictive modeling for marketing and sales. For purposes of targeting customers in Marketing, the *lift factor* represents the expected increase in response rates when a model is used compared to the situation when no model is used to determine the customers to be targeted (Witten & Frank, 2005; Berry & Linoff, 2000). In order to plot a lift chart, the scores (probability values) assigned by the model on the test data are sorted into ascending (or descending) order and then grouped into deciles. A score for each group (decile) is then computed as the mean score within each group (Giudici & Figini, 2009; Witten & Frank, 2005; Giudici, 2003; Berry & Linoff, 2000). More precisely, a lift chart is a plot on a 2-dimensional Cartesian plane with the *x* and *y* values defined as (Vuk & Curk, 2006):

$$x = YRATE(\lambda), y = TPRATE(\lambda) \tag{4.7}$$

The lift factor for each decile is computed as the ratio between the score assigned by the model and the score when no model is used (random guessing). The lift chart is plotted with the deciles on the horizontal axis and the cumulative lift factor values on the vertical axis. A baseline line that represents random guessing is also plotted. As for ROC curves, the area between the base line and the cumulative lift curve indicates the quality of the model. The larger the area, the better the model. A discussion of why lift analysis was not used is provided in chapter 6.

## 4.8 Software used for the experiments

Various software packages were used for the experiments as shown in table 4.12. The datasets were stored in a Microsoft SQL Server database. Storing the datasets in a database made it especially easy to establish the composition of each dataset, and to pre-process the KDD Cup 1999 dataset, using SQL statements and stored

procedures. Dataset sampling was implemented using stored procedures implemented in the Microsoft SQL Server procedural language.

*Table 4.12: Software used for the experiments*

| Task / Activity | Software |
|---|---|
| Dataset storage and retrieval | MS SQL Server 2000 |
| Dataset sampling | Stored procedures implemented in the MS SQL Server procedural language |
| Measurement of correlation coefficients | Specialised code implemented in Borland C++ Builder 5 |
| Feature subset search | Specialised code implemented in Borland C++ Builder 5 |
| KNN classification (modeling) | Specialised code implemented in Borland C++ Builder 5 |
| Classification tree modeling | See 5 – Windows version of the C5.0 classifier |
| Aggregate modeling | Specialised code implemented in Borland C++ Builder 5 |
| ROC analysis | Specialised code implemented in Borland C++ Builder 5 |
| Statistical hypothesis testing | SPSS versions 15 and 17 |
| Generation of descriptive statistics for datasets | SPSS versions 15 and 17, MS SQL Server 2000 SQL, Ms Excel 2003 |
| Various activities | MS Excel 2003 |

The See5 classifier (Quinlan, 2004), which is the MS Windows version of the C5.0 classifier for Unix, was used for classification tree construction. SPSS versions 15 and 17 for MS Windows were used for conducting the Student's t-tests for the statistical analysis of model performance. Specialised applications were created in Borland C++ Builder 4 and 5 for measuring class-feature and feature-feature correlation coefficients, feature selection, the KNN classifiers, aggregate model classifiers, and ROC analysis. It should be pointed out that statistical software provides functions for correlation measurement and ROC analysis. However specialised software was implemented in order to speed up the experiments.

## 4.9 Chapter summary

The main research questions, central argument of this thesis, and research methods have been presented and justified in this chapter. The design science research paradigm in conjunction with the scientific method were used as a conceptual framework for the research. The datasets used for the experiments were obtained from the UCI KDD Archive and UCI Machine Learning repository. The descriptive

statistics of the datasets, as well as the pre-processing that was done on the datasets have been discussed. Sequential random sampling was used to obtain random samples from large datasets. The algorithms that were used for modeling, namely: classification tree and K-Nearest Neighbour, have been presented. The measures of predictive performance that were used in the experiments have been discussed. Finally, the software used for the experiments has been presented. In the next four chapters, the experiments that were conducted, the results that were obtained and the proposed methods for feature and dataset selection are presented. The theoretical models that were deduced from the experimental results are discussed in chapter 10.

# Chapter 5
# Feature Selection for Large Datasets

> .. the object of data analysis is not to model the fleeting random patterns of the moment, but to model the underlying structures which give rise to consistent and replicable patterns. ..' (Hand, 1998)

It was stated in chapters 2 and 3 that the selection of a good subset of predictive features results in the reduction of the variance component of a predictive model. To the author's knowledge, there are very few reported studies on research that addresses feature selection in the presence of large datasets. One such study has been reported by Liu and Setiono (1998a, 1998b). Research has been reported on validation of class-feature correlation coefficients using fake variables (Stoppiglia et al, 2003; Bi et al, 2003). Since this method of validation has only been applied to small datasets, it is useful to establish whether the use of fake variables for validation can be effectively applied to feature selection from large datasets. It was also argued in chapter 3 that algorithms that conduct feature subset search should use clearly specified definitions of feature relevance.

The purpose of this chapter is to report the experimental results of the study of feature subset selection in the presence of sampling from large datasets. Experimental results are reported on studies that were conducted on two correlation measures, two validation methods for correlations, and three algorithms for feature subset selection. Hand (1998) has made the insightful observation about data analysis as quoted at the beginning of this chapter, and it is in the spirit of this observation that experiments for this chapter were designed and conducted. It is argued in this chapter that statistical methods can be used to make inferences on the expected values of the feature correlations for large datasets when many samples are used. The use of many samples for correlation measurement should lead to better decisions for feature selection since the correlation values obtained are more reliable. It is further argued that features that are selected when domain-specific definitions of feature relevance are incorporated into the feature selection procedures are the best features for the prediction task at hand. In the context of processing large datasets in data mining, the following research questions are answered in this chapter:

*1. How can class-feature correlations be measured in order to produce a reliable ranking of features for a dataset?*
*2. What methods of validation for feature correlations result in reliable feature selection?*
*3. How can domain-specific definitions of feature relevance be incorporated into feature selection procedures?*

The rest of this chapter is organised as follows. Section 5.1 gives a summary of the feature selection problem. Section 5.2 presents the different approaches to feature selection that were studied. Empirical studies of feature ranking, feature subset search and predictive performance of selected feature subsets are respectively discussed in sections 5.3, 5.4 and 5.5. The discussion of the experimental results and conclusions are respectively given in sections 5.6 and 5.7.

## 5.1 The feature selection problem revisited

It was stated in chapter 3 that the initial selection of features is typically done by a domain expert, based on the data mining task at hand. Subsequent to this, a process of selecting the most relevant features and eliminating redundant features must be conducted. It is this process which is addressed in this chapter. Further, Guyon and Elisseeff (2003) have observed that there is not just the one method of feature selection that suits all datasets, all algorithms and all data mining tasks. With Guyon and Elisseeff's (2003) observations in mind, the methods discussed in this chapter were directed at large datasets of moderately high dimensionality.

It was also stated in chapter 3 that filtering methods are preferred to wrapper methods for data mining for reasons of efficiency. The measurement of class-feature and feature-feature correlations is at the core of many filtering methods for feature selection. In the experiments reported for this chapter, many of the correlation measures commonly used in filtering methods for feature selection (Yu & Liu, 2004; Hall, 1999, 2000) were adopted to establish feature relevance and redundancy in the presence of sampling. For feature relevance, these measures are based on the strength of the correlation between a feature and the class variable. For redundancy, the measures are based on the strength of the correlations between the features. The correlation measures were presented in chapter 3. While studies of feature

selection most commonly use one sample (i.e. the whole dataset) to establish the feature correlation values, the studies reported in this chapter were directed at using many samples to establish the correlation values for the dataset features.

## 5.2 Alternative approaches to feature selection for large datasets

When large datasets are available the question arises as to whether relevant features should be selected based on the whole dataset, one sample from the dataset, or several samples taken from the dataset. When a dataset is very large, it is not feasible to compute correlation values using all the available data. If only one sample is taken there is a great risk of making the wrong decisions about which features are the most relevant. Based on Hand's (1998) observations as quoted at the beginning of this chapter, the purpose of feature selection should not be to identify the best features for the specific training sample that has been chosen (or happens to be available) for model creation, but rather to identify the best features for model creation regardless of the specific training sample that is chosen. In other words, the objectives of feature selection should be directed at the data generating process and not solely at the data sample that happens to be available. In attempting to answer the question:

*How can class-feature correlations be measured in order to produce a reliable ranking of features for a dataset?*

the author hypothesised that the use of many samples to measure class-feature and feature-feature correlations should provide reliable estimates of these correlations. In order to provide evidence to support this hypothesis, the following alternatives were considered and studied:

**Alternative 1**

Use one small sample (e.g. 100 or 500 instances) to measure correlations and select the features, and assume that these will be the relevant features for the instance space and prediction task regardless of the specific training sample used for classification model construction. The motivation for this alternative is that, first of all, the computation of correlation values from small samples is faster than for large samples. Secondly, statistical theory points to the fact that relationships that appear

to be strong in small samples are generally stronger than relationships which only appear in large samples of data. The foregoing observation can be used to argue that features that have strong correlations in small samples are the strongest predictors and will have globally predictive power. The term *globally predictive* is used to mean that a feature will have predictive power in all regions of the instance space.

**Alternative 2**

Use one large sample (e.g. 1000 instances) to measure correlations and select the features and assume that these will be relevant features for the instance space and prediction task regardless of the specific training sample used for classification model construction. The rationale here is that those features which are not strongly predictive may be eliminated when a small sample is used. The use of a large sample increases the chances of identifying more features for the prediction task.

**Alternative 3**

Use many small samples of one size to select the features and assume that these will be relevant features for the instance space and prediction task regardless of the specific training sample used for classification model construction. The rationale here is the same as for alternative 1. Additionally, taking the mean values of the correlations measured on many samples, and using statistical inference to select features is more reliable than the use of a single sample correlation.

**Alternative 4**

Use many large samples of one size to select the features and assume that these will be relevant for the instance space and prediction task regardless of the specific training sample used for classification model construction. Again, taking the mean values of the correlations measured on many samples, and using statistical inference to select features is more reliable than using a single sample correlation.

The next section provides the experimental results for the investigation of the above four alternatives.

## 5.3 Empirical study of feature ranking methods for large datasets

For feature ranking the selection criteria are applied to each feature, without any consideration of the contribution of the other features to the prediction performance. The ranking criteria reported in this section are based on feature correlation measures. The results of the experiments that were conducted on feature selection based on *pure ranking* of features are reported in this section. The experimental procedures that were used are given in section 5.3.1. A comparison of Pearson's and Kendall's correlation measures is given in section 5.3.2. Sections 5.3.3 and 5.3.4 respectively provide the experimental results and discussions for feature ranking based on a single samples and feature ranking based on many samples.

### 5.3.1 Experimental procedure for the study of feature ranking

The datasets presented in chapter 4 were used for the experiments. The sequential random sampling method (SRS), described in chapter 4, was used to obtain random samples. Probes (fake variables) with values drawn from both Gaussian and uniform distributions were used. Probes may be added to the datasets prior to taking samples for feature selection. However, adding probes to a very large dataset is a computationally lengthy and unnecessary process. The probes can be added during or after the sampling step. The generation of pseudo-random numbers is a process of sampling from the specified distribution (Thomas et al, 2007). The sampling approach used for the experiments was to first take samples from the large dataset and then sample from the chosen probability distributions for the probes (fake variables). Three types of probes were used with values drawn from a Gaussian distribution, a uniform distribution, and uniform binary distribution. For the Gaussian probe, the Marsaglia-Bray algorithm was used to generate the pseudo-random numbers (Thomas et al, 2007). For the uniform probes, the Borland C++ function for generating random numbers was used. The datasets that were used in the experiments for this thesis contain quantitative (discrete and continuous), and qualitative (nominal and ordinal) features. The forest cover type and KDD Cup 1999 also contain binary (quantitative discrete) features. Furthermore, even though the correlation values (for quantitative features) and symmetrical uncertainty (SU) coefficient values (for qualitative features) are comparable, the functions used to

compute the correlations are different. The functions for computing Pearson's correlation Kendall's correlation and SU coefficients were presented in chapter 3. It is statistically meaningful to compare a true binary feature with a fake binary feature and a true qualitative feature with a fake qualitative variable. For this reason one Gaussian and two uniform probes were used. Table 5.1 shows the characteristics of the probes used for the datasets.

*Table 5.1 Characteristics of the probes for the datasets*

| Probe name | Value range | Description |
|---|---|---|
| Probe1GaussCont | 0- 999 | Gaussian distribution with mean = 500, stdev = 100 |
| Probe2UniformCont | 0-999 | uniform distribution |
| Probe3UniformBin | 0,1 | uniform distribution with binary values |

Class-feature and feature-feature correlation coefficients were computed from samples drawn from a large dataset, for both the true features and the probes (fake variables). Two methods were studied for computing correlations: Pearson's correlation coefficient and Kendall's *tau* correlation coefficient. Two criteria were studied for feature ranking selection: statistical significance with the t-test on mean values for correlations and symmetrical uncertainty (SU) coefficients, and statistical significance based on probes. Two algorithms, See5 for classification trees and Nearest Neighbours (5NN) were used for comparison. It should be noted that these two classification algorithms differ significantly in their treatment of predictive features during model construction. The 5NN algorithm does not have the ability to rank features or select relevant features, while the classification tree algorithm performs an implicit ranking of features and also performs tree pruning to ensure that only statistically significant information provided by the features is used in the final classification tree.

## 5.3.2 Comparison of Pearson's and Kendall's correlation measures

For the comparison of Pearson's and Kendall's correlation coefficients, experiments were conducted to compare the mean values of the class-feature correlations using 10 samples to compute each mean value. Correlation values for the top 10 variables are shown in table 5.2. Table 5.2 shows the class-feature correlation values for the three datasets: Forest cover type, KDD Cup 1999 and Abalone3C. For each dataset, the top 10 features as ranked by Kendall's *tau* are shown. It should be noted that the

forest cover type dataset has 54 features, the KDD Cup 1999 dataset has 41 features, and the abalone3C dataset has eight features. Only the top 10 features for the forest cover type and KDD Cup 1999 datasets are shown in table 5.2 for purposes of concise presentation, and for illustration of the differences between the Pearson's *r* and Kendall's *tau* coefficients.

*Table 5.2 Comparison of mean values for Kendall's tau and Pearson's r*

| Dataset | Top 10 features as ranked by Kendall's tau | Mean values for correlation coefficients for 10 test samples | | | |
|---|---|---|---|---|---|
| | | Sample size = 1000 | | Sample size = 500 | |
| | | Kendall's tau | Corresponding Pearson's r | Kendall's tau | Corresponding Pearson's r |
| Forest cover type | WildernessArea4 | 0.86 | 0.22 | 0.81 | 0.22 |
| | SoilType12 | 0.70 | 0.14 | 0.72 | 0.16 |
| | SoilType1 | 0.69 | 0.08 | 0.44 | 0.06 |
| | SoilType38 | 0.68 | 0.12 | 0.60 | 0.12 |
| | SoilType39 | 0.68 | 0.11 | 0.58 | 0.11 |
| | SoilType2 | 0.64 | 0.07 | 0.58 | 0.09 |
| | SoilType4 | 0.64 | 0.10 | 0.57 | 0.11 |
| | SoilType6 | 0.60 | 0.08 | 0.56 | 0.09 |
| | SoilType22 | 0.59 | 0.14 | 0.57 | 0.14 |
| | SoilType10 | 0.58 | 0.13 | 0.47 | 0.11 |
| | | | | | |
| KDDCup99 | SerrorRate | 0.92 | 0.51 | 0.87 | 0.45 |
| | NumCompromised | 0.92 | 0.23 | 0.85 | 0.26 |
| | SrvSerrorRate | 0.91 | 0.50 | 0.90 | 0.43 |
| | WrongFragment | 0.90 | 0.21 | 0.81 | 0.18 |
| | DstHostSrvSerrorRate | 0.85 | 0.50 | 0.83 | 0.43 |
| | DstHostSrvRerrorRate | 0.85 | 0.34 | 0.76 | 0.27 |
| | SrvRerrorRate | 0.85 | 0.35 | 0.80 | 0.28 |
| | Hot | 0.84 | 0.11 | 0.78 | 0.14 |
| | DstHostSerrorRate | 0.84 | 0.51 | 0.81 | 0.44 |
| | RerrorRate | 0.82 | 0.34 | 0.76 | 0.27 |
| | | | | | |
| Abalone 3C (all features) | Diameter | 0.50 | 0.41 | 0.50 | 0.41 |
| | Shellweight | 0.52 | 0.40 | 0.53 | 0.40 |
| | Height | 0.51 | 0.37 | 0.52 | 0.39 |
| | WholeWeight | 0.49 | 0.38 | 0.50 | 0.38 |
| | VisceraWeight | 0.49 | 0.38 | 0.49 | 0.38 |
| | ShuckledWeight | 0.45 | 0.34 | 0.45 | 0.34 |
| | Length | 0.17 | 0.14 | 0.18 | 0.15 |
| | Gender (qualitative) | 0.12 | 0.12 | 0.13 | 0.13 |

The mean correlation values shown in table 5.2 for sample sizes of 500 and 1000 indicate that for the forest cover type and KDD Cup 1999 datasets the class-feature correlations as measured by Kendall's *tau* are generally much larger than the class-feature correlations measured using Pearson's correlation coefficient. Secondly,

even among the top 10 features out of 54 features for forest cover type, the two features *SoilType1* and *SoilType2* have strong class-feature correlations based on Kendall's *tau* but have insignificant correlations based on Pearson's *r*. A feature ranking method based on Pearson's *r* would eliminate the features *SoilType1* and *SoilType2*. Thirdly, for both forest cover type and KDD Cup 1999 the feature rankings based on Kendall's *tau* are different from the rankings based on Pearson's *r*.

The reader will recall from chapter 3 that the point was made that Wilcox (2001) has cautioned against the interpretation of Pearson's *r* for measuring correlations when there is no guarantee that the correlation between two variables is linear, and when outliers have not been given special treatment. A small Pearson's correlation coefficient between two variables does not necessarily mean that the two variables are not strongly correlated. It could be the case that the correlation is not linear or the correlation is masked by the presence of outliers in the data. On the other hand, Kendall's *tau* is a robust measure of correlation which will provide reliable correlation values even when the correlation is not linear and even when outliers are present in the data (Wilcox, 2001). For the Abalone3C dataset, the results of table 5.2 indicate that the differences between the class-feature correlations measured with Kendall's *tau* and Pearson's *r* are marginal and the feature rankings based on both correlation measures are nearly the same. Based on Wilcox's (2001) observations, it can be deduced that Pearson's *r* is a suitable correlation measure for the Abalone3C dataset because there are no outliers in the data and the predictive features are linearly correlated to the class variable. It can be deduced from table 5.2 that Pearson's *r* is not a suitable correlation measure for the forest cover type and KDD Cup 1999 datasets because the datasets either have outliers or the correlations between the features and the class variables are non-linear.

Table 5.3 and tables D.1, D.4 and D.7 of appendix D respectively give the number of features that would be selected for the forest cover type, KDD Cup 1999 and Abalone 3C datasets based on the Students t-test of means. The test was conducted to determine the features whose mean class-feature correlation coefficient is greater than or equal to *0.1*. The reader will recall from the discussion of chapter 3 that Cohen (1998) has advised that a correlation value with a magnitude in the interval [0, 0.1) has no practical significance in any domain for data analysis and a correlation value with a magnitude in the interval [0.1, 1.0] may have practical significance. For the forest cover type dataset only 6 out of 54 features would be selected based on

Pearson's *r*. Based on the foregoing observations all subsequent experiments for feature selection were based on Kendall's *tau* as the correlation measure.

*Table 5.3 Comparison of the number of selected features for Kendall's tau and Pearson's r*

| Dataset (no. of features) | Sample size | Number of features with a mean $corr_{cf}$ or mean *SU* coefficient that is significant ($corr_{cf}$ >= 0.1, significance level 0.01) | |
|---|---|---|---|
| | | Kendall's *tau* | Pearson's *r* |
| Forest cover type (54) | 500 | 35 | 6 |
| | 1000 | 38 | 6 |
| KDDCup99 (41) | 500 | 36 | 26 |
| | 1000 | 30 | 21 |
| Abalone (3 class) (8) | 500 | 6 | 5 |
| | 1000 | 7 | 5 |

## 5.3.3 Feature ranking based on a single sample

The problems and consequences of using a single small or large sample are investigated and made explicit in this section. Ten small samples, 10 medium samples, and 10 large samples were taken from the forest cover type dataset using sequential random sampling (SRS). Table 5.4 shows the number of features selected for each sample by the Gaussian probe and Z-test based on class-feature correlations measured using Kendall's *tau*. For the probes, the selection criterion is a class-feature correlation coefficient greater than that of the Gaussian probe. The number of features selected by the uniform probe and uniform-binary probe are given in tables D.1, D.4 and D.7 of appendix D. Since only one correlation value is available for each predictive feature for these experiments, the Z-test for a single correlation value was used to test the hypothesis that the class-feature correlation value is greater or equal to 0.1, that is, features that have a correlation value which is of practical significance (Cohen, 1988). The Z-test for a single correlation measurement was discussed in chapter 3. The first problem that can be deduced from table 5.4 is that sample sizes of 100 result in very few features being selected. The second problem is that the number of selected features varies from sample to sample. Smyth (2001) has argued that if a single sample is used to measure correlations between variables, then features may be lucky (or unlucky) in the sample and get selected (or eliminated) based on the single correlation measurement.

It could be argued that as sample sizes get larger the variability in the measured correlation coefficient will decrease. However, even for sample sizes of 1000 which is large for statistical hypothesis testing, one can see from table 5.4 that the variability in the number of features selected is still high. A second problem that arises when a

single sample is used for feature selection is illustrated in table 5.5. Table 5.5 shows the class-feature correlation values for four of the features in the KDD Cup 1999 dataset, as measured using Kendall's *tau* with samples of size 1000. It can be deduced from table 5.5 that a feature (e.g. *NumFailedLogins*) can have no correlation, small correlation, medium correlation, or high correlation with the class variable depending on the sample that is used, even when the sample size for correlation measurements is large. Alternatives 1 and 2 as stated in section 5.2 were discarded due to the three problems discussed above and no further studies of correlation measurement with small sample sizes (size = 100) were conducted.

*Table 5.4: Number of selected features based on single samples for forest cover type*

| Sample ID | **Number of relevant features** with a significant class-feature Kendall's tau correlation selected by the Gaussian probe and Z-test for forest cover type | | | | | |
|---|---|---|---|---|---|---|
| | size = 100 | | size = 500 | | size = 1000 | |
| | Gaussian Probe | Z-test $\|corr_{cf}\| >= 0.1$ | Gaussian Probe | Z-test $\|corr_{cf}\| >= 0.1$ | Gaussian Probe | Z-test $\|corr_{cf}\| >= 0.1$ |
| S1 | 35 | 22 | 46 | 31 | 46 | 35 |
| S2 | 34 | 14 | 46 | 37 | 43 | 40 |
| S3 | 30 | 13 | 46 | 34 | 43 | 35 |
| S4 | 35 | 18 | 47 | 36 | 48 | 39 |
| S5 | 39 | 16 | 46 | 34 | 49 | 41 |
| S6 | 30 | 23 | 42 | 32 | 49 | 37 |
| S7 | 34 | 15 | 42 | 32 | 48 | 38 |
| S8 | 35 | 21 | 47 | 33 | 46 | 34 |
| S9 | 34 | 16 | 46 | 34 | 48 | 37 |
| S10 | 35 | 19 | 39 | 32 | 49 | 41 |

*Table 5.5: Kendall's correlations for four features for KDD Cup 1999*

| Feature | Sample ID | | | | | | | | | | $corr_{cf}$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 | S9 | S10 | Mean | Stdev |
| NumFailedLogins | 0.33 | 0.58 | 0 | 0 | 0.58 | 0.39 | 0 | 0.33 | 0.49 | 0.33 | 0.30 | 0.23 |
| NumShells | 0.21 | 0.21 | 0.42 | 0 | 0.34 | 0.32 | 0 | 0 | 0.2 | 0.35 | 0.20 | 0.16 |
| NumAccessFiles | 0.35 | 0 | 0 | 0.33 | 0.43 | 0.22 | 0 | 0 | 0 | 0.44 | 0.18 | 0.20 |
| SUAttempted | 0 | 0.21 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.02 | 0.07 |

## 5.3.4 Feature ranking based on many samples

The rationale behind using many samples is that the use of one sample will lead to misleading conclusions as demonstrated above in section 5.3.3. Taking the mean over the correlation values for many samples should provide a more reliable estimate of the correlation values. Smyth (2001) has argued that a feature may be highly correlated with the class for a given sample, simply because it is lucky in that particular sample. In fact, the results of section 5.3.3 have illustrated this point precisely. The use of many samples for correlation measurement also enables the

validation of selected feature subsets using more robust (less prone to error) statistical methods. For 4 datasets, 10 medium sized samples (size = 500) and 10 large samples (size=1000) were used to compute the Kendall's *tau* and the *SU* coefficient for the class-feature associations. Two criteria were used for feature selection. The first criterion was to select features based on the confidence interval of the mean correlation value of the probe. If a feature has a confidence interval whose lower and upper values are both greater than the lower and upper values of the confidence interval for the probe then the feature is selected, otherwise it is rejected. The second criterion was to use Student's t-test on the mean value of Kendall's *tau* or *SU* coefficient at the 0.01 significance level.

*Table 5.6: Number of selected features based on 10 samples*

| Dataset (no. of features) | Sample size | Number of relevant features with mean $corr_{cf}$ (Kendall's tau) or mean *SU* that is statistically significant. Number of samples = 10 | | | |
|---|---|---|---|---|---|
| | | Selection based on probes | | | t-test for ($corr_{cf}$>=0.1 or SU>=0.1 at the 0.01 level) |
| | | Probe1 (Gaussian) | Probe2 (uniform-cont) | Probe3 (uniform-bin) | |
| Forest cover (54) | 500 | 47 | 47 | 44 | 35 |
| | 1000 | 49 | 48 | 47 | 38 |
| KDDCup99 (41) | 500 | 36 | 36 | 36 | 34 |
| | 1000 | 36 | 36 | 35 | 30 |
| Abalone (8) | 500 | 8 | 8 | 8 | 6 |
| | 1000 | 8 | 8 | 8 | 7 |
| Mushroom(22) | 500 | 21 | 15 | 14 | 2 |

Table 5.6 shows the results for the number of selected features based on two criteria. The two uniform probes selected approximately the same number of features. The Gaussian probe selected approximately the same number of features as the uniform probes, except in the case of the mushroom dataset. The t-test is very strict as it selects the smallest number of features. The details of the features selected by the Gaussian probe for the forest cover type and KDD Cup 1999 are given in tables D.2 and D.5 of appendix D.

For the experiments of this section many samples were used to measure class-feature correlations and to conduct validation for the selected features using probes and the t-test for mean correlation values. The experimental results demonstrated that for medium sized samples (size = 500) and large sized samples (size = 1000) each validation method selects nearly the same number of features. However, different validation methods select different numbers of features. The Gaussian

probe is the least strict of all the methods as it generally selects more features. The t-test is the most strict as it generally selects the smallest number of features.

Based on the results of table 5.6, alternatives 3 and 4 as stated in section 5.2 provided useful options for correlation measurement, feature ranking and validation. For validation based on probes the variability in the number of selected features is low for both medium size (500) and large size (1000) samples even though the Gaussian probe does not work well for the mushroom dataset (all features are qualitative). Performance of the feature subset search algorithms based on the features selected in this section as inputs, are discussed in the next section.

## 5.4 Empirical study of feature subset search

Feature subset search is the process of searching for an optimal subset of features based on specified criteria. A common criterion is to select that subset of features (from a set of identified relevant features) that maximises relevance and minimises redundancy in the selected subset. Feature subset search methods and examples of the merit measures that are employed in heuristic search for feature subsets were discussed in detail in chapter 3. The experiments reported in this section are for feature subset selection using forward search. Forward search algorithms that employ the correlation-based feature selection (CFS) merit measure (Hall, 1999) and differential prioritisation (DP) measures (Ooi et al, 2007) were implemented and tested using the features selected in the last section as inputs. Section 5.4.1 provides a discussion and analysis of the implementation of feature relevance and redundancy definitions by the CFS (Hall, 1999) and DP (Ooi et al, 2007) search procedures. The weaknesses of the merit measures employed by the CFS and DP search procedures are made explicit. A new algorithm for feature subset search is proposed in section 5.4.2 and the algorithm's feature selection performance is compared to the CFS and differential prioritisation methods.

### 5.4.1 Implementation of feature relevance and redundancy definitions

A good feature ranking method should be followed by a good search procedure. A good feature subset search procedure should not have a *search bias* which forces it to prefer an irrelevant feature to a relevant one.

When the *search bias* is based on precise and domain-specific definitions of *weak*, *medium*, and *strong* feature correlations then the selected feature subset should be the best for that application domain (Lutu & Engelbrecht, 2010). If fake variables are included in the initial feature set, then they should only be used to indicate when the search should stop. In other words, if the search procedure finds that the best feature to select at a given point is a fake variable, then the search procedure should terminate. Possible terminating criteria in the presence of fake variables (probes) should then be: (1) *stop when a pre-specified number of features have been selected* or (2) *stop when a probe is encountered as the next best choice*.

Definitions of feature relevance (Blum & Langley, 1997) and feature redundancy (Koller & Sahami, 1996) were given in chapter 3. It was also stated in chapter 3 that many implementations of feature selection implement the meanings of relevance and redundancy using the level of class-feature and feature-feature correlations. For feature selection implementations it is generally accepted that a relevant feature is one which is *highly correlated* with the class variable and a redundant feature is one that is *highly correlated* with other features (Ooi et al, 2007; Yu & Liu, 2004; Hall, 1999, 2000). Table 5.7 provides a summary of common interpretations of levels of class-feature and feature-feature correlations for purposes of identifying relevant and redundant features. One problem with heuristic procedures for feature subset search, for example DP (Ooi et al, 2007) and CFS (Hall, 1999, 2000) is that the merit measures they use do not have sufficient precision to distinguish between *high correlation* as opposed to *not-high correlation*. It is demonstrated later in this section that there are several situations where the CFS search procedure (Hall,1999, 2000) and DP search procedure (Ooi et al,2007) prefer features with very low feature-feature correlations at the cost of eliminating features with high class-feature correlations.

*Table 5.7: Interpretation of levels of feature correlations for heuristic search*

| Situation | class-feature correlation for feature *f* | mean feature-feature correlation of selected features if *f* is added to selected features | Interpretation according to the literature |
|---|---|---|---|
| s1 | not high | not high | *f* is irrelevant |
| s2 | not high | High | *f* is redundant |
| s3 | High | not high | *f* is relevant |
| s4 | High | High | *f* is redundant |

Experiments for feature subset selection were conducted on the forest cover type and KDD Cup 1999 datasets since these are large datasets with large numbers of features as commonly encountered in predictive data mining (Hand et al, 2001; Hand, 1998). The purpose of the experiments was to establish the behaviour of the CFS and differential prioritization algorithms for feature subset search. Table 5.8 shows a partial trace of the computations of the CFS search procedure for the datasets. For one iteration of the CFS algorithm (column 5), the CFS algorithm selects the best feature (column 2) based on the value of the CFS merit measure (column 6). The CFS merit measure (discussed in chapter 3) is computed using the mean values of the class-feature and feature-feature correlations for the candidate feature subsets. For each iteration, columns 3 and 4 of table 5.8 show the value of the class-feature correlation for the selected feature and total feature-feature correlation for the subset of selected features.

For the situations depicted in table 5.7, when making a choice between a feature whose situation is *s1* and one whose situation is *s3*, CFS chooses the situation *s1* feature. For the forest cover type features, at iteration number 26, it would be preferable to choose one of *SoilType13* or *SoilType39* instead of the binary-valued probe since each of these features has a high class-feature correlation and its selection would result in a low level of feature-feature correlation for the selected features. At iteration 39 it would be better to choose *SoilType13* or *SoilType39* instead of *SoilType 25.* Similarly, at iteration number 21 for the KDD Cup 1999 dataset, it should be preferable to choose one of *DstHostSrvSerrorRate, DstHostSrvRerrorRate* or *Count*, instead of the Gaussian probe for the same reasons as stated above. The *search bias* of the CFS search procedure forces it to choose the Gaussian probe instead, since CFS does not have sufficient information to make the distinctions that are made in table 5.7.

*Table 5.8: Trace of the CFS search procedure for the forest cover type and KDD Cup 1999*

| Dataset | Selected feature F | class-feature correlation ($corr_{cf}$) for *f* | Total feature-feature correlation ($corr_{ff}$) for selected features | Iteration | Merit |
|---|---|---|---|---|---|
| Forest cover type | Probe3UniformBin | 0.051 | 10.752 | 26 | 1.518 |
| | Probe2UniformCont | 0.044 | 10.752 | 27 | 1.509 |
| | Probe1GaussCont | 0.037 | 10.752 | 28 | 1.499 |
| | SoilType20 | 0.161 | 11.352 | 29 | 1.489 |
| | SoilType25 | 0.08 | 11.569 | 30 | 1.48 |
| | SoilType13 | 0.527 | 14.095 | 31 | 1.471 |
| | SoilType15 | 0.028 | 14.095 | 32 | 1.462 |
| | SoilType39 | 0.676 | 17.99 | 33 | 1.447 |
| | | | | | |
| KDD Cup 1999 | Probe1GaussCont | 0.032 | 12.589 | 21 | 1.309 |
| | Probe2UniformCont | 0.028 | 12.589 | 22 | 1.299 |
| | SUAttempted | 0.021 | 12.589 | 23 | 1.289 |
| | Land | 0.001 | 12.589 | 24 | 1.276 |
| | IsHostLogin | 0 | 12.589 | 25 | 1.263 |
| | DstHostSrvSerrorRate | 0.855 | 17.715 | 26 | 1.25 |
| | DstHostSrvRerrorRate | 0.847 | 23.547 | 27 | 1.237 |
| | Count | 0.631 | 28.713 | 28 | 1.22 |

Figures 5.1 and 5.2 respectively show the plots of the merit measures when the forward search procedure was implemented with the CFS merit measure, and when it was implemented with the differential prioritization (DP) measure. For the DP measure figures 5.1 and 5.2 show the plots of merit values for $\alpha = 0.5$ (DP050 Merit), $\alpha = 0.75$ (DP075 Merit) and $\alpha = 0.95$ (DP095 Merit). The plots show the values of the merit measures for each iteration of the search procedure until all features have been processed. A disturbing observation is that it is not obvious from the plots when the search procedure should terminate. Hall (1999) has stated that in the presence of feature interactions, CFS may fail to select the optimal subset of features. The discussion of detailed executions of the CFS search procedure provided earlier in this section demonstrated that the CFS merit measure can favour noise over true features. For the differential prioritisation measure, the stopping criterion is easier to determine, since Ooi et al (2007) have stated that one objective of the search procedure which uses this measure is to identify a pre-specified number of features. For the differential prioritisation measure it is difficult to justify the choice of the $\alpha$ value in relation to any precise definition of relevance and redundancy unless domain-specific information is available.

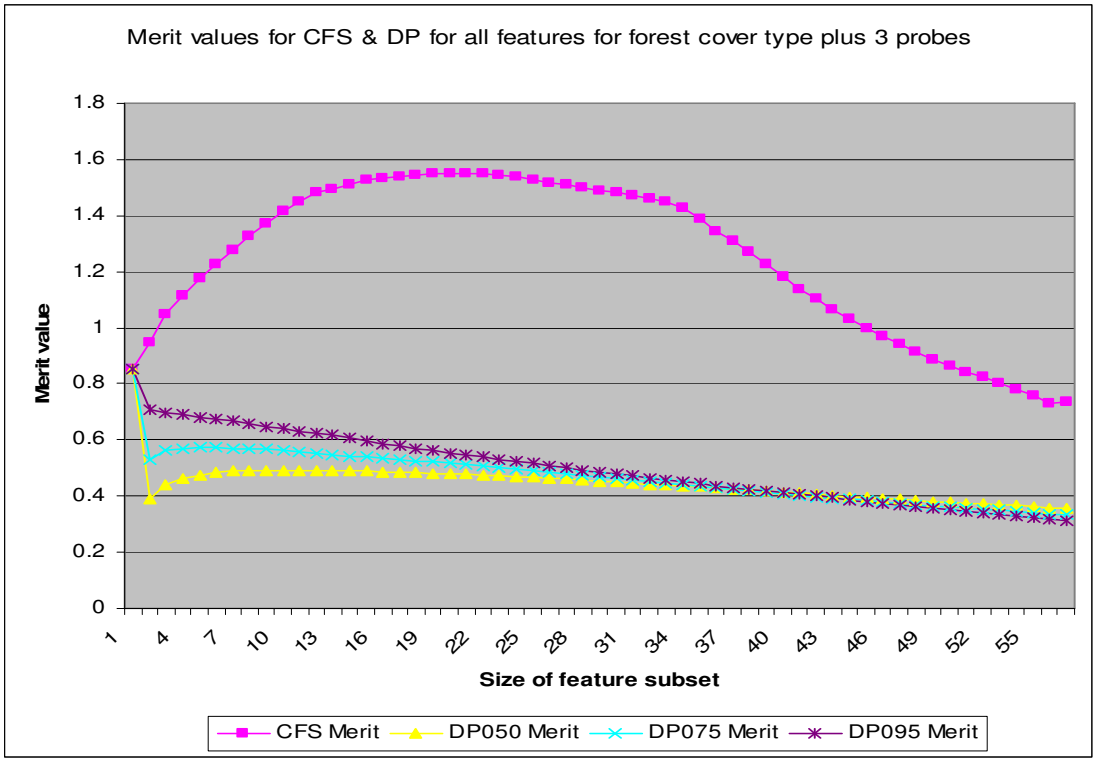


*Figure 5.1: Merit values for the forest cover type dataset without pre-selection*

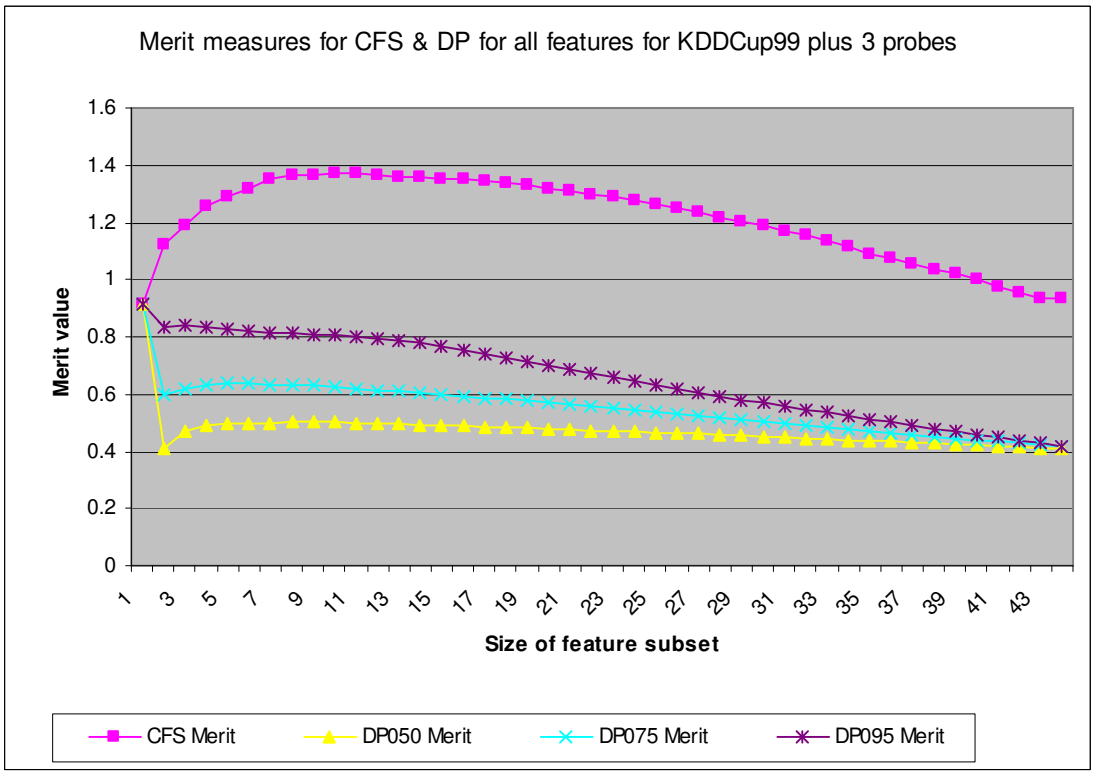


*Figure 5.2: Merit values for the KDD Cup 1999 dataset without feature pre-selection*

The experimental results reported in this section have revealed two weaknesses of the merit measures employed by the CFS and differential prioritisation search

procedures. Firstly, the mathematical functions used as merit measures sometimes select pure noise in preference to predictive features. Secondly, the stopping criteria for the two search procedures can be difficult to establish for some datasets. This was found to be the case for the forest cover type dataset and for the KDD Cup 1999 dataset. Based on the foregoing observations the author was led to hypothesise that the use of more precise definitions for interpretation of correlation values should eliminate the above problems that arise with the CFS and differential prioritisation merit measures.

### 5.4.2 A reliable search procedure for feature subset search

One possible solution to the problems exhibited by the CFS and DP merit measures is to use a feature selection criterion that precisely implements a given definition of relevance and redundancy. The definition of feature relevance should be supplied by domain experts in terms of what values of correlations are considered to be *low*, *medium* and *high*. The idea of incorporating user supplied domain knowledge in model construction is not new. Osei-Bryson (2004) has proposed the incorporation of user-specified preferences and value functions in the post pruning of classification trees. Yu and Liu (2004) have proposed the incorporation of user-specified threshold values of class-feature correlations for feature relevance analysis and selection. The method of differential prioritisation proposed by Ooi et al (2007) enables a user to control the levels of feature relevance and redundancy for the selected feature subset.

Formal definitions of feature relevance and redundancy were given in chapter 3. For feature selection based on relevance and redundancy analysis, Yu and Liu (2004) have defined four categories of features, namely (1) *irrelevan*t, (2) *weakly relevant and redundant,* (3) *weakly relevant and non-redundant*, and (4) *strongly relevant*. Yu and Liu (2004) have argued that the optimal subset of features should consist of features that fall in categories 3 and 4, that is, *weakly relevant and non-redundant*, and *strongly relevant*. The four categories of features proposed by Yu and Liu (2004) are based on Blum and Langley's (1997) definition of feature relevance and Koller and Sahami's (1996) definition of redundancy as discussed in chapter 3.

A feasible refinement of the feature relevance and redundancy definitions of table 5.7 is shown in table 5.9 for purposes of heuristic feature subset search. The refinement

is based on all possible combinations of the levels *insignificant, low, medium* and *high* correlation for class-feature and feature-feature correlations. Columns 5 and 6 respectively show the interpretation of each combination and the source of motivation for the interpretation. Column 2 shows the symbols used to label the distinct interpretations of the correlation level combinations. The categorisation suggests that unselected features fall into one of six categories at the time when the search algorithm needs to make a decision as to which feature to select next for inclusion in the set of already selected features. The six categories denoted by A,B,C,D,E and F correspond to the interpretations *strongly relevant* (category A), *relevant* (category B), *weakly relevant* (category C), *weakly redundant* (category D), *redundant* (category E), and *irrelevant* (category F).

Yu and Liu (2004) have defined four categories of features for relevance and redundancy analysis, as discussed earlier in this section. For the proposed categorisation of table 5.8 two categories (*weakly redundant* and *redundant*) are used to represent redundant features and two categories (*relevant* and *strongly relevant*) are used to represent strongly relevant features. The motivation for using six categories was to provide the heuristic search procedure with the ability to make higher precision distinctions between features compared to the level of precision provided by the CFS and DP measures.

As an example of the interpretation of the correlation levels *insignificant*, *low, medium and high* shown in table 5.9, Cohen's (1988) proposal for the interpretation of correlation coefficients for behavioural sciences research could be used. The reader will recall that according to Cohen's (1988) definitions, a correlation coefficient in the range [0,0.1) indicates a correlation with no practical significance (*insignificant*). A correlation coefficient in the range [0.1, 0.3) indicates a *low* correlation. A correlation coefficient in the range [0.3, 0.5) indicates a *medium* correlation, and a correlation coefficient in the range [0.5, 1.0] indicates a strong (*high*) correlation. The categorisation of table 5.9 can then be used as follows. When the input variables to the search procedure are pre-selected, for example, using the t-test or a probe, then (situation sp1, category F) will not arise during the search. If there is no pre-selection of features, then the situation (situation sp1, category F) may arise. The merit measure should then be replaced by clear logic which implements the interpretation of class-feature and feature-feature correlations based on table 5.9 It should be noted that the categories are dynamic, that is, the category of a given feature will

change based on the currently selected features since the mean correlation with already selected features (column 4 of table 5.9) is not a static quantity.

*Table 5.9: Proposed definition of feature relevance and redundancy based on user specified levels*

| Situation (new interpre-tation) | Category | class-feature correlation for *f*: $corr_{cf}(f)$ | mean correlation with selected features: $\overline{corr_{ff}}(f)$ | Proposed new interpretation | Source of motivation for interpretation of category |
|---|---|---|---|---|---|
| sp1 | F | insignificant | any level | irrelevant | (Blum & Langley, 1997) and (Yu & Liu,2004) |
| sp2 | C | Low | insignificant | weakly relevant | |
| sp3 | C | Low | Low | weakly relevant | |
| sp4 | C | Low | Medium | weakly relevant | |
| sp5 | F | Low | High | irrelevant | |
| sp6 | B | Medium | insignificant | relevant | |
| sp7 | B | Medium | Low | relevant | |
| sp8 | D | Medium | Medium | weakly redundant | (Koller & Sahami,1996) and (Yu & Liu,2004) |
| sp9 | E | Medium | High | redundant | |
| sp10 | A | High | Insignificant | strongly relevant | (Blum & Langley, 1997) and (Yu & Liu,2004) |
| sp11 | A | High | Low | strongly relevant | |
| sp12 | D | High | Medium | weakly redundant | (Koller & Sahami,1996) and (Yu & Liu,2004) |
| sp13 | E | High | High | redundant | |

A new search algorithm was designed to use the categorisation shown in table 5.9 to conduct a search for the best subset of features. In general, a heuristic search procedure creates a search tree whose nodes represent various states of the search space (Luger & Stubblefield, 1993; Pearl, 1984). The heuristic search procedure will expand that node which is most promising based on the value of a heuristic (merit) measure. Typical implementations of heuristic search employ several lists to record the current state of the search tree. The new algorithm, which is given in figure 5.3, uses a list called FEATURES to hold all currently unselected features, a list called CHILDREN to hold all the nodes for features that are candidates for selection, and a list called SELECTED to hold all currently selected nodes. Initially, the SELECTED list holds the feature with the highest $corr_{cf}$ value. When making a decision on the next feature to include in the SELECTED list, the algorithm will prefer a strongly relevant feature, if such a feature exists. If there are no strongly relevant features at that point, the algorithm will prefer a relevant feature. If there are no strongly relevant

or relevant features at that point, the algorithm will prefer a weakly relevant feature. If there are no strongly relevant, relevant or weakly relevant features at that point, the algorithm will prefer a weakly redundant feature. If there is no feature which falls in one of the categories strongly relevant, relevant, weakly relevant or redundant, the algorithm terminates. The motivation for allowing the algorithm to select weakly redundant features is due to the fact that Guyon & Elisseeff (2003) have reported experiments which demonstrate that feature interactions are not necessarily harmful.

**Initialise**

**Step 1:** Place all features and their correlations on the FEATURES list

**Step 2:** Create a node for the feature with the highest $corr_{cf}$ and place the node on the SELECTED list

**Process:**

**Step 3**: If the FEATURES list is not empty, create a node for each feature on FEATURES and place it on the CHILDREN list. If the FEATURES list is empty, go to step 11.

**Step 4**: For each node on the CHILDREN list, establish the $corr_{cf}$ and mean $corr_{ff}$ with already selected features on SELECTED.

**Step 5:** Assign a category to each node on the CHILDREN list, based on table 5.9.

**Step 6:** Delete from the CHILDREN list any node which belongs to the E and F categories. This leaves the categories strongly relevant (A), relevant (B), weakly relevant (C ) and weakly redundant (D).

**Step 7:** If the CHILDREN list has only one feature in category A (strongly relevant), put it on SELECTED and delete it from the FEATURES list and go to step 3.

If there is more than one feature in category A, then call function *GetBestInCat* to select the best one. Go to step 3. Otherwise go to step 8.

**Step 8:** If the CHILDREN list has only one feature in category B (relevant), put it on SELECTED and delete it from the FEATURES list and go to step 3.

If there is more than one feature in category B, then call function *GetBestInCat* to select the best one and go to step 3. Otherwise go to step 9.

**Step 9:** If the CHILDREN list has only one feature in category C (weakly relevant), put it on SELECTED and delete it from the FEATURES list and go to step 3.

If there is more than one feature in category C, then call function *GetBestInCat* to select the best one. Go to step 3. Otherwise go to step 10.

**Step 10:** If the CHILDREN list has only one feature in category D (weakly redundant), put it on SELECTED and delete it from the FEATURES list and go to step 3.

If there is more than one feature in category D, then call function *GetBestInCat* to select the best one and go to step 3. Otherwise go to step 11.

**Step 11:** Terminate the search procedure and return the nodes on SELECTED as the selected features.

*Figure 5.3: Decision rule-based algorithm based on definitions of relevance and redundancy*

At the time of selecting a feature for inclusion in the SELECTED list, if more than one feature fall in the preferred category, the algorithm uses the decision rules shown in table 5.10 to choose between two features, $f_1$ and $f_2$. The rules of table 5.10 are implemented in the function *Better_than(f1,f2)* which returns true if $f_1$ is better than $f_2$ (i.e. the decision should be to choose $f_1$). Figure 5.4 shows the algorithm for the function *GetBestInCat(CT)* for searching for the best feature of the *CT* category. This function utilises the function *Better_than(f1,f2).*

```
1.  first =  index of first node in  category CT
2. best = CHILDREN[first]
3. count = number of nodes on the CHILDREN list
4. For (i=first+1; i < count, i++)
        feature = CHILDREN[i]
        if (feature belongs to category CT) and Better_than(feature, best)
              best = feature
  end-for
4.  Return best
```

*Figure 5.4: The algorithm GetBestInCat(CT) to select the best features in one category*

*Table 5.10: Decision rules for choosing between two features of the same category*

| Class-feature correlation | mean feature-feature correlation with selected features | Decision | Reason |
|---|---|---|---|
| $corr_{cf}(f_1) > corr_{cf}(f_2)$ | $corr_{ff}(f_1) \leq corr_{ff}(f_2)$ | choose $f_1$ | prefer feature with higher class-feature & lower feature-feature correlation |
| | $corr_{ff}(f_1) > corr_{ff}(f_2)$ | choose $f_2$ | prefer feature with lower feature-feature correlation |
| $corr_{cf}(f_1) < corr_{cf}(f_2)$ | $corr_{ff}(f_1) < corr_{ff}(f_2)$ | chose $f_2$ | prefer feature with higher class-feature correlation |
| | $corr_{ff}(f_1) \geq corr_{ff}(f_2)$ | choose $f_2$ | |
| $corr_{cf}(f_1) = corr_{cf}(f_2)$ | $corr_{ff}(f_1) < corr_{ff}(f_2)$ | choose $f_1$ | prefer feature with lower feature-feature correlation |
| $corr_{cf}(f_1) = corr_{cf}(f_2)$ | $corr_{ff}(f_1) > corr_{ff}(f_2)$ | choose $f_2$ | |
| $corr_{cf}(f_1) = corr_{cf}(f_2)$ | $corr_{ff}(f_1) = corr_{ff}(f_2)$ | break tie randomly | identical levels of relevance & redundancy |

The results for the feature subset search for the KDD Cup 1999 and forest cover type datasets are respectively listed in tables 5.11 and 5.12. The tables show the search results when the input list consists of all features, including probes. The summary results for the four datasets used in the experiments are given in table 5.13. The results of tables 5.11 and 5.12 show that the search algorithm does not select any probes.

*Table 5.11: Output of the decision rule-based search algorithm without feature pre-selection for KDD Cup 1999*

| Feature | Category | Selection Reason | $corr_{cf}(f)$ | *mean* $corr_{ff}(f)$ | Number of elected features |
|---|---|---|---|---|---|
| SerrorRate | | | 0.916 | 0 | 1 |
| DstHostRerrorRate | A | Strongly relevant | 0.805 | 0.272 | 2 |
| NumRoot | A | Strongly relevant | 0.677 | 0.281 | 3 |
| WrongFragment | A | Strongly relevant | 0.901 | 0.281 | 4 |
| Flag | B | Relevant | 0.428 | 0 | 5 |
| NumFailedLogins | B | Relevant | 0.303 | 0 | 6 |
| DstHostSerrorRate | A | Strongly relevant | 0.835 | 0.278 | 7 |
| DstHostSrvCount | B | Relevant | 0.313 | 0.227 | 8 |
| SrvCount | B | Relevant | 0.423 | 0.268 | 9 |
| DstHostCount | B | Relevant | 0.368 | 0.258 | 10 |
| Hot | A | Strongly relevant | 0.845 | 0.289 | 11 |
| Service | C | Weakly relevant | 0.236 | 0 | 12 |
| NumAccessFiles | C | Weakly relevant | 0.177 | 0 | 13 |
| NumCompromised | A | Strongly relevant | 0.915 | 0.298 | 14 |
| Counted | A | Strongly relevant | 0.631 | 0.281 | 15 |
| ProtocolType | C | Weakly relevant | 0.151 | 0 | 16 |
| SrvDiffHostRate | B | Relevant | 0.455 | 0.286 | 17 |
| SrcBytes | B | Relevant | 0.49 | 0.297 | 18 |
| RootShell | C | Weakly relevant | 0.108 | 0 | 19 |
| NumShells | C | Weakly relevant | 0.204 | 0.098 | 20 |
| NumFileCreations | C | Weakly relevant | 0.297 | 0.139 | 21 |
| DstHostSrvRerrorRate | A | Strongly relevant | 0.847 | 0.295 | 22 |
| DstHostSameSrcPortRate | C | Weakly relevant | 0.284 | 0.262 | 23 |
| DstHostDiffSrvRate | C | Weakly relevant | 0.144 | 0.268 | 24 |
| DstHostSameSrvRate | C | Weakly relevant | 0.224 | 0.296 | 25 |
| Duration | C | Weakly relevant | 0.254 | 0.361 | 26 |
| DstBytes | D | Weakly redundant | 0.584 | 0.332 | 27 |
| DstHostSrvDiffHostRate | D | Weakly redundant | 0.439 | 0.352 | 28 |
| DstHostSrvSerrorRate | D | Weakly redundant | 0.855 | 0.42 | 29 |
| DiffSrvRate | D | Weakly redundant | 0.727 | 0.444 | 30 |
| SrvRerrorRate | D | Weakly redundant | 0.845 | 0.456 | 31 |
| RerrorRate | D | Weakly redundant | 0.822 | 0.482 | 32 |

Secondly, the algorithm never selects irrelevant or redundant features as defined in table 5.9. Thirdly, for forest cover type the algorithm selects nearly the same number of features when pre-selection is done using probes as shown in table 5.13. Table 5.13 provides a summary of the number of features selected by the decision rule-based algorithm for different input feature sets that were generated in the experiments of the last section. The results for the features selected by the validation and ranking methods were given in table 5.6. The results of table 5.13 show that the t-test is far more restrictive compared to the pre-selection of features using probes. A comparison of tables 5.6 and 5.13 shows that the decision rule-based search algorithm selects nearly all the features that are pre-selected by the t-test.

*Table 5.12: Output of the decision rule-based search algorithm without feature pre-selection for forest cover type*

| Feature | Category | Selection Reason | $corr_{cf}(f)$ | mean $corr_{ff}(f)$ | Selected feature count |
|---|---|---|---|---|---|
| WildernessArea4 | | | 0.855 | 0 | 1 |
| SoilType2 | A | Strongly relevant | 0.642 | 0.243 | 2 |
| SoilType40 | A | Strongly relevant | 0.547 | 0.146 | 3 |
| SoilType38 | A | Strongly relevant | 0.676 | 0.162 | 4 |
| SoilType4 | A | Strongly relevant | 0.638 | 0.164 | 5 |
| SoilType1 | A | Strongly relevant | 0.686 | 0.178 | 6 |
| SoilType3 | A | Strongly relevant | 0.548 | 0.185 | 7 |
| SoilType6 | A | Strongly relevant | 0.603 | 0.192 | 8 |
| SoilType13 | A | Strongly relevant | 0.527 | 0.199 | 9 |
| SoilType39 | A | Strongly relevant | 0.676 | 0.283 | 10 |
| SoilType21 | B | Relevant | 0.322 | 0 | 11 |
| SoilType35 | B | Relevant | 0.443 | 0.02 | 12 |
| SoilType12 | A | Strongly relevant | 0.704 | 0.286 | 13 |
| SoilType34 | B | Relevant | 0.4 | 0.021 | 14 |
| SoilType19 | B | Relevant | 0.351 | 0.02 | 15 |
| SoilType22 | A | Strongly relevant | 0.593 | 0.287 | 16 |
| SoilType18 | B | Relevant | 0.44 | 0.04 | 17 |
| SoilType26 | B | Relevant | 0.431 | 0.038 | 18 |
| SoilType17 | B | Relevant | 0.433 | 0.066 | 19 |
| SoilType10 | A | Strongly relevant | 0.579 | 0.295 | 20 |
| SoilType5 | B | Relevant | 0.36 | 0.066 | 21 |
| SoilType16 | B | Relevant | 0.329 | 0.084 | 22 |
| SoilType11 | B | Relevant | 0.476 | 0.162 | 23 |
| WildernessArea2 | B | Relevant | 0.391 | 0.221 | 24 |
| SoilType30 | B | Relevant | 0.34 | 0.266 | 25 |
| SoilType14 | C | Weakly relevant | 0.231 | 0 | 26 |
| SoilType8 | C | Weakly relevant | 0.176 | 0 | 27 |
| SoilType37 | C | Weakly relevant | 0.126 | 0 | 28 |
| SoilType9 | C | Weakly relevant | 0.283 | 0.018 | 29 |
| SoilType28 | C | Weakly relevant | 0.215 | 0.018 | 30 |
| SoilType27 | C | Weakly relevant | 0.147 | 0.017 | 31 |
| SoilType23 | B | Relevant | 0.399 | 0.293 | 32 |
| SoilType20 | C | Weakly relevant | 0.161 | 0.052 | 33 |
| SoilType24 | C | Weakly relevant | 0.259 | 0.203 | 34 |
| SoilType31 | C | Weakly relevant | 0.223 | 0.25 | 35 |
| HorizDistToFire | C | Weakly relevant | 0.156 | 0.263 | 36 |
| HorizDistToRoad | C | Weakly relevant | 0.158 | 0.266 | 37 |
| Slope | C | Weakly relevant | 0.124 | 0.283 | 38 |
| SoilType33 | C | Weakly relevant | 0.183 | 0.323 | 39 |
| SoilType32 | C | Weakly relevant | 0.207 | 0.349 | 40 |
| Elevation | C | Weakly relevant | 0.277 | 0.377 | 41 |
| SoilType29 | C | Weakly relevant | 0.295 | 0.465 | 42 |

The point was made in the last section that the t-test is more strict than the probes at eliminating irrelevant features. When there is no pre-selection of features for the input to the decision rule-based search algorithm, or when the input consists of features

pre-selected using the Gaussian probe, the decision rule-based search algorithm eliminates a larger number of the input features compared to when the input features are pre-selected by the t-test. A tentative conclusion that can be made from this observation is that the probes admit some features that are possibly irrelevant. A more detailed discussion of this issue is given later in this chapter.

*Table 5.13: Features selected by the decision rule-based algorithm for sample sizes of 1000*

| Dataset (number of features) | Number of features selected by the decision rule algorithm when the input is pre-selected using: | | | | |
|---|---|---|---|---|---|
| | no pre-selection | Gaussian probe | Uniform probe | Uniform-bin probe | t-test (sig =0.01) |
| Forest cover type (54) | 42 | 41 | 41 | 41 | 36 |
| KDD Cup 1999 (41) | 32 | 34 | 34 | 34 | 30 |
| Abalone (8) | 3 | 3 | 3 | 3 | 2 |
| Mushroom (22) | 14 | 14 | 14 | 14 | - |

## 5.5 Predictive performance for features selected with different methods

Classifiers were constructed to compare the predictive performance of the features selected by the different methods of feature selection. The 5NN and See5 classification tree algorithms were used for classification. In this section the results and analysis of the predictive performance of the classifiers are reported. Section 5.5.1 provides a description of the experimental procedures. The Predictive performance of the forest cover type and KDD Cup 1999 classifiers are respectively given in sections 5.5.2 and 5.5.3. Predictive performance results for the small dataset classifiers (abalobe3C and mushroom) are presented in section 5.5.4.

### 5.5.1 Experimental procedure for classifier creation and testing

The point was made in section 5.2 that the objective of feature selection should **not be** to identify the best features for the training sample that has been chosen (or happens to be available) but rather to identify the best features for model creation for any sample taken from the instance space for the prediction task. This observation was based on Hand's (1998) advice quoted at the beginning of this chapter. Based on the foregoing observation, training samples much larger than the samples used for feature selection were used for the experiments. Classification models were created for purposes of testing the predictive performance of the feature subsets selected by the feature selection methods presented in sections 5.3 and 5.4. Two

classification algorithms 5NN and See5, were used in the experiments. For each dataset the same training set (samples) and same test set (samples) were used for 5NN and See5 classification. Predictive accuracy (on instances not seen during training) was established on 10 test sets for each classifier.

In section 5.3 feature ranking was reported for four datasets, two sizes of samples (500 and 1000) for measuring correlations and four validation methods (three probes and t-test). In section 5.4 feature subset search was reported for the decision rule-based algorithm for four datasets, five types of input features (no pre-selection, three probes and t-test) selected using sample sizes of 1000 to measure correlations. To conduct experiments to test all the feature ranking methods on two algorithms would require 4 x 2 x 4 x 2 = 64 classifiers to be created. To generate test results for 10 test sets (samples) for each classifier would result in 640 test runs. To conduct experiments to test feature subsets selected with the decision rule-based algorithm would require 5 x 4 x 2 = 40 classifiers. The generation of test results for 10 test sets (samples) for each classifier would result in 400 test runs. Additionally, four classifiers must be created with all the features (no selection) and tested on 10 test sets for comparison with the classifiers created with selected feature subsets, which results in an additional 40 test runs. The total number of test runs would then be 640 + 400 + 40 = 1080. If two types of class distributions are used, as was done for the experiments, this number would double to 2160.

To avoid the factorial explosion in the number of test runs as described above, researchers are advised to sample from the space of all possible factor combinations (Cohen, 1995:pg 88). The decision made for the experiments was as follows: Only feature subsets selected using correlations measured with samples of size 1000 were used. For feature ranking methods feature subsets selected by one probe were used (Gaussian probe for forest cover type, KDD Cup 1999 and abalone3C, and uniform probe for mushroom). These were compared to classifiers constructed with all the features. Classifiers were also constructed for feature subsets selected by the decision rule-based method with one type of input (no pre-selection of features). For the forest cover type dataset (the largest dataset) two sample sizes of 6000 and 12000 instances were used. For the other three (smaller datasets), one sample size was used. Additionally, for the large datasets classifiers were created for two types of class distributions to illustrate the difficulty of establishing the true positive rates (TPRATE) for individual classes when the parent dataset class distribution is used in

the presence of minority classes. This resulted in the number of test runs being reduced to 400.

### 5.5.2 Classification results for forest cover type

Classifiers were constructed to compare the predictive performance obtained when the feature sets selected by the different methods as discussed in section 5.5.1 are used. Tables 5.14 shows the classification results for the forest cover type dataset samples on two class distributions. Column 4 gives the classification results for tests based on the class distribution of the parent dataset. 10-fold cross validation was used to measure the predictive accuracy. Column 5 shows the classification results for tests based on an equal class distribution. Ten test samples were used to measure the predictive accuracy. The results are shown for two sample sizes (6000 and 12000). For each sample size the predictive accuracy on all the classes is shown for all features (54), features selected by the Gaussian probe (49), and features selected by the decision rule based search algorithm (42).

*Table 5.14: Predictive accuracy for forest cover type based on two class distributions*

| Algorithm | Feature selection method (number of features) | Sample size | Mean predictive accuracy and 95% CI of mean | |
|---|---|---|---|---|
| | | | 10-fold cross validation parent dataset distribution | 10 test sets equal class distribution |
| 5NN (nearest neighbours) | All features (54) | 6000 | 71.2 ± 1.1 | 75.1 ± 1.4 |
| | | 12000 | 76.2 ± 0.8 | 80.1 ± 1.0 |
| | Gaussian probe (49) | 6000 | 71.5 ± 1.1 | 75.1 ± 1.1 |
| | | 12000 | 76.1 ± 0.8 | 79.4 ± 0.8 |
| | Decision rule search (42) | 6000 | 68.5 ± 1.4 | 71.6 ± 1.2 |
| | | 12000 | 70.4 ± 1.1 | 74.4 ± 0.9 |
| See5 (classification tree) | All features (54) | 6000 | 74.8 ± 1.2 | 73.6 ± 0.9 |
| | | 12000 | 74.5 ± 0.8 | 76.6 ± 0.9 |
| | Gaussian probe (49) | 6000 | 73.8 ± 1.2 | 73.6 ± 0.9 |
| | | 12000 | 75.4 ± 1.1 | 76.5 ± 0.9 |
| | Decision rule search (42) | 6000 | 72.8 ± 0.8 | 72.3 ± 0.7 |
| | | 12000 | 74.5 ± 0.6 | 76.9± 1.0 |

Table 5.15 gives the results of the statistical tests to compare the predictive performance of the classifiers based on the parent dataset class distribution. The paired samples t-test is not applicable in this case since there are no paired tests, so the independent samples t-test was used to compare the performance of two classifiers. The independent samples t-test revealed that the predictive performance of the 5NN classifiers constructed with all 54 features and those constructed with 49 features as selected by the Gaussian probe do not statistically differ in predictive

accuracy, for all sample sizes. However, the classifiers constructed with 42 features (as selected by the decision rule-based search) have a predictive performance that is significantly lower than that when all 54 features are used. For the classifiers constructed using the See5 classification tree algorithm, there is no statistically significant difference between using all features (54) and features selected by the Gaussian probe (49). There was also no significant difference between using all features (54) and features selected by the decision rule search algorithm (42).

*Table 5.15: Statistical tests to compare the accuracy of forest cover type classifiers for different feature subsets for parent dataset class distribution*

| Algorithm | Groups for independent samples t-test, sample size, number of features | | Student's independent samples t-test (9df) (equal variances not assumed) | | |
|---|---|---|---|---|---|
| | Group A (mean & CI) | Group B (mean & CI) | 95% CI of mean difference | p-value (2 tails) | Group A better than Group B? |
| 5NN (nearest neighbour) | 6000; 54 (71.2 ± 1.1) | 6000; 42 (68.5 ± 1.4) | [1.8, 3.8] | 0.000 | yes |
| | 12000; 54 (76.2 ± 0.8 ) | 12000; 42 (70.4 ± 1.1) | [5.2, 6.4] | 0.000 | yes |
| | 6000; 54 (71.2 ± 1.1) | 6000; 49 (71.5 ± 1.1) | [-1.0, 0.5] | 0.468 | no |
| | 12000; 54 (76.2 ± 0.8 ) | 12000; 49 (76.1 ± 0.8) | [-0.4, 0.6] | 0.640 | no |
| See5 (classification tree) | 6000; 54 (74.8 ± 1.3) | 6000; 42 (72.8 ± 0.8) | [0.3, 3.6] | 0.019 | yes |
| | 12000; 54 (74.5 ± 0.8) | 12000; 42 (74.5 ± 0.6) | [-1.0, 1.0] | 0.952 | no |
| | 6000; 54 (74.7 ± 1.2) | 6000; 49 (73.8 ± 1.2) | [-0.9, 2.8] | 0.283 | no |
| | 12000; 54 (74.5 ± 0.8) | 12000; 49 (75.4 ± 1.1 ) | [-2.4, 0.5] | 0.176 | no |

Table 5.16 shows the results for the Student's paired samples t-test for the predictive accuracy on the 10 test sets. Columns 2 and 3 respectively provide the description of the classifiers that were compared. A specification of the training set size for the classifier, the size of the feature set that was used for the classifier and the mean predictive accuracy of the classifier on 10 test samples are given. Column 4 gives the 95% confidence interval of the mean difference for the predictive accuracy of the two classifiers specified in columns 2 and 3. Columns 5 and 6 respectively give the p-value for the paired samples t-test and the interpretation of the p-value based on the reasoning given in table 4.11 of chapter 4.

For the 5NN classifiers created with training sample sizes of 6000 and 12000 instances, the 54-feature classifiers provided a higher level of predictive accuracy compared to the 42-feature classifiers. The 49-feature classifiers provided a higher level of predictive accuracy than the 42-feature classifiers. These results indicate that the decision rule-based algorithm based on Cohen's (1998) thresholds for *insignificant, low, medium* and *high* correlations eliminates some features which have predictive power for the 5NN algorithm. For the 5NN classifiers created with training samples of 6000 and 12000 instances there is no statistically significant difference in predictive accuracy between the 54-feature classifiers and the 49-feature classifiers. These results indicate that the Gaussian probe eliminates only features with no predictive power for 5NN.

For the See5 classifiers created with training sample sizes of 6000 instances, the 54-feature classifiers provided a higher level of predictive accuracy than the 42-feature classifiers. The 49-feature classifiers also provide a higher level of predictive accuracy that the 42-feature classifiers. For classifiers created with training sample sizes of 12000 instances there was no statistically significant difference in predictive accuracy between the 54-feature and 42-feature classifiers, and between the 54-featute and 42-feature classifiers. These results indicate that for the See5 classifiers the 42 features selected by the decision rule-based algorithm based on Cohen's (1998) guidelines are sufficient for prediction with very large samples (e.g. 12000 instances).

A detailed analysis of the 5NN and See5 classifiers was conducted for the classifiers of sample size 12000 in order to establish the TPRATE values for the individual classes. The analysis results are given in table 5.17. The analysis was done to compare the predictive performance of the 49 features selected by the Gaussian probe and the 42 features selected by the decision rule-based search algorithm. The results of the Student's paired samples t-test for the 5NN classifiers indicate that for three of the classes (1, 2, 7) there is no statistically significant difference between using 49 features and 42 features for 5NN classification. However, for four of the classes (3, 4, 5, 6) there is a statistically significant increase in the TPRATE values when 49 features are used. The results of the Student's paired samples t-test for the See5 classifiers indicate that there is no statistically significant difference between using 49 features and 42 features for five of the classes (2, 3, 4, 5, 6). The TPRATE for the 49-feature classifier is statistically significantly higher than that for the 42-

feature classifier for class 1. The TPRATE for the 42-feature classifier is statistically significantly much higher than that for the 49-feature classifier for class 7.

*Table 5.16: Statistical tests to compare the accuracy of forest cover type classifiers for different feature subsets for equal class distribution*

| Algorithm | Groups for paired tests sample size; number of features | | Student's paired t-test (9df) | | |
|---|---|---|---|---|---|
| | Group A (mean & CI) | Group B (mean & CI) | 95% CI of mean difference | p value (2 tails) | Group A better than Group B? |
| 5NN (nearest neighbours) | 6000; 49 (75.1 ± 1.1) | 6000; 42 (71.6 ± 1.2) | [1.7, 5.2] | 0.002 | yes |
| | 12000; 49 (79.4 ± 0.8) | 12000; 42 (74.4 ± 0.9) | [3.6, 6.3] | 0.000 | yes |
| | 6000; 54 (75.1 ± 1.4) | 6000; 42 (71.6 ± 1.2) | [1.9, 5.0] | 0.000 | yes |
| | 12000; 54 (80.1 ± 1.0) | 12000; 42 (74.4 ± 0.9) | [4.2, 7.2] | 0.000 | yes |
| | 6000; 54 (75.1 ± 1.4) | 6000; 49 (75.1 ± 1.1) | [-1.5, 1.4] | 0.451 | no |
| | 12000; 54 (80.1 ± 1.0) | 12000; 49 (79.4 ± 0.8) | [-0.6, 1.8] | 0.290 | no |
| See5 (classification tree) | 6000; 49 (73.6 ± 0.9) | 6000; 42 (72.3 ± 0.7) | [0.3, 2.3] | 0.014 | yes |
| | 12000; 49 (76.5 ± 0.9) | 12000; 42 (76.9 ± 1.0) | [-1.3, 0.5] | 0.324 | no |
| | 6000; 54 (73.6 ± 0.9) | 6000; 42 (72.3 ± 0.7) | [0.3, 2.3] | 0.014 | yes |
| | 12000; 54 (76.6 ± 0.9) | 12000; 42 (76.9 ± 1.0) | [-1.2,0.6] | 0.490 | no |
| | 6000; 54 (73.6 ± 0.9) | 6000; 49 (73.6 ± 0.9) | no variance | no variance | no |
| | 12000; 54 (76.6 ± 0.9) | 12000; 49 (76.5 ± 0.9) | [-0.01, 0.3] | 0.495 | no |

When samples are randomly selected from a large dataset and the class-feature correlations are measured, the correlation values obtained reflect the predictive power of the features over the whole instance space. This is called *global predictive power* in this thesis. If a large dataset was clustered and samples taken from each cluster to measure the class-feature correlations, then the correlation values obtained would reflect the predictive power of the features for a given cluster. Features that have significant class-feature correlations only for a cluster and not for the whole instance space are said to be locally predictive within that cluster. This is called *local predictive power* in this thesis.

*Table 5.17: Statistical tests to compare TPRATE performance of forest cover type classifiers for different feature subsets for training sample size 12000*

| Algorithm | Groups for paired tests sample size; number of features | | Student's paired t-test (9df) | | |
|---|---|---|---|---|---|
| | Group A (12000;49) | Group B (12000;42) | 95% CI of mean difference | p value (2 tailed) | Group A better than Group B? |
| 5NN (nearest neighbours) | All classes-A (79.4 ± 0.8) | All classes-B (74.4 ± 0.9) | [3.6, 6.3] | 0.000 | yes |
| | Class 1-A (60.6 ± 3.0) | Class 1-B (60.4 ± 4.0) | [-6.3, 6.7] | 0.473 | no |
| | Class 2-A (46.2 ± 3.1) | Class 2-B (47.8 ± 4.4) | [-4.4, 1.2] | 0.224 | no |
| | Class 3-A (67.2 ± 4.0) | Class 3-B (54.4 ± 2.1) | [7.7, 17.9] | 0.000 | yes |
| | Class 4-A (97.4 ± 0.4) | Class 4-B (90.4 ± 3.0) | [3.3, 10.7] | 0.002 | yes |
| | Class 5-A (97.6 ± 0.8) | Class 5-B (93.8 ± 0.9) | [2.5, 5.1] | 0.000 | yes |
| | Class 6-A 82.0 ± 2.9) | Class 6-B (72.2 ± 2.9) | [5.1, 14.5] | 0.000 | yes |
| | Class 7-A (94.8 ± 1.0) | Class 7-B (92.4 ± 4.1) | [-2.0, 6.8] | 0.250 | no |
| See5 (classification tree) | All classes-A (76.5 ± 0.9) | All classes-B (76.9 ± 1.0) | [-1.3,0.5] | 0.324 | no |
| | Class 1-A (61.4 ± 4.1) | Class 1-B (57.4 ± 3.4) | [1.3, 6.7] | 0.008 | yes |
| | Class 2-A (61.2 ± 3.0) | Class 2-B (63.8 ± 3.0) | [-5.2, 0.2] | 0.050 | no |
| | Class 3-A (64.8 ±3.4) | Class 3-B (60.8 ± 3.3) | [-0.4, 8.4] | 0.034 | yes |
| | Class 4-A (96.6 ± 1.0) | Class 4-B (96.8 ± 1.0) | [-2.2, 1.8] | 0.414 | no |
| | Class 5-A (84.0 ± 1.7) | Class 5-B (86.2 ± 2.4) | [-5.2, 0.8] | 0.128 | no |
| | Class 6-A (79.8 ± 2.5) | Class 6-B (77.8 ± 3.3) | [-0.1, 2.1] | 0.062 | yes |
| | Class 7-A (87.6 ± 3.4) | Class 7-B (95.6 ± 1.6) | [-11.2, -4.8] | 0.000 | no |

The observations for the test results of tables 5.16 and 5.17 led the author to hypothesise as follows: If a large dataset has features that only have local predictive power, such features will have small class-feature correlations and will therefore appear to be non-relevant when one of the validation methods (Student's t-test of

means) and decision rule-based search algorithm studied in this chapter are used. This hypothesis was not tested in this thesis, and is left for future work.

### 5.5.3 Classification results for KDD Cup 1999

5NN and See5 classification models were also constructed for the KDD Cup 1999 dataset. The challenge for the KDD Cup 1999 dataset is to achieve a high level of accuracy on the attack classes R2L and U2R on the test dataset. The KDD Cup 1999 test dataset was presented in chapter 4. Classification performance results for this dataset are most commonly presented in terms of the TPRATE values for the classes (Lee et al, 2002; Lee & Stolfo, 2000). The predictive performance results are therefore presented here in terms of the accuracy on all classes as well as the TPRATE values for each of the 5 classes. Table 5.18 shows the predictive performance of the classifiers. The performance results are shown for 10-fold cross validation on the training set, and for 10 test samples drawn from the test dataset.

For 10-fold cross validation, a training sample of 4500 instances was used. For the minority class, U2R, all 52 instances of that class were included in the sample. For the remaining four classes, sequential random sampling was used. For the classifiers based on an equal distribution of the classes, a training sample of 4500 instances was created with 1000 instances from each of the four classes NORMAL, DOS, PROBE and R2L, and 500 instances for the class U2R. The 500 instances of the class U2R were obtained using bootstrap sampling of the 52 instances that appear in the training dataset. The aim was to try as much as possible to achieve an equal distribution, but a decision was made not to bootstrap the U2R class beyond ten times the actual size. The test samples were created by taking all 70 instances of the class U2R in the test dataset and using sequential random sampling to obtain 70 instances for each of the remaining classes.

From table 5.6 it can be deduced that the 3 probes select nearly the same numbers of features for the KDD Cup 1999 dataset. The results of table 5.13 show that all input feature subsets result in nearly the same number of features being selected, with the decision rule-based search algorithm selecting the smallest number of features. Classifiers were constructed with all the 41 features (all features) and with the 32 features selected by the decision rule based search algorithm. Classifiers were not constructed with the 36 features selected by the Gaussian probe as initial

exploratory studies (Cohen, 1995) revealed that the predictive performance of 41 features is not significantly different from that of the 32 features.

*Table 5.18: Predictive performance of KDD Cup 1999*

| Algorithm (training sample size) | Feature selection method (number of features) | Class | Natural distribution mean TPRATE% for 10-fold cross validation | Equal class distribution mean TPRATE% for 10 test sets |
|---|---|---|---|---|
| 5NN (nearest neighbours) (size = 4500) | All features (41) | All classes (accuracy) | 97.0 ± 0.5 | 73.5 ± 0.9 |
| | | NORMAL | 98.4 | 85.6 ± 3.2 |
| | | DOS | 97.5 | 67.3 ± 5.0 |
| | | PROBE | 60.0 | 95.9 ± 1.2 |
| | | R2L | 61.9 | 73.1 ± 2.2 |
| | | U2R | 65.2 | 45.7 ± 0.0 |
| | Decision rule (32) | All classes (accuracy) | 96.4 ± 0.5 | 69.9 ± 1.3 |
| | | NORMAL | 98.2 | 84.7 ± 3.2 |
| | | DOS | 97.0 | 67.1 ± 4.8 |
| | | PROBE | 94.7 | 95.9 ± 1.2 |
| | | R2L | 72.3 | 70.3 ± 3.3 |
| | | U2R | 38.0 | 31.4 ± 0.0 |
| See5 (classification tree) (size = 4500) | no selection (41) | All classes (accuracy) | 98.6 ± 0.5 | 66.3 ± 1.2 |
| | | NORMAL | 99.6 | 97.3 ± 1.1 |
| | | DOS | 99.5 | 74.3 ± 6.2 |
| | | PROBE | 98.2 | 86.2 ± 2.2 |
| | | R2L | 76.5 | 25.4 ± 2.3 |
| | | U2R | 71.2 | 48.6 ± 0.0 |
| | Decision rule (32) | All classes (accuracy) | 97.5 ± 0.5 | 66.3 ± 1.2 |
| | | NORMAL | 99.3 | 97.3 ± 1.1 |
| | | DOS | 96.3 | 74.3 ± 6.2 |
| | | PROBE | 98.4 | 86.2 ± 2.2 |
| | | R2L | 68.4 | 25.4 ± 2.3 |
| | | U2R | 65.4 | 48.6 ± 0.0 |

From the 10-fold cross validation results of table 5.18 it can be deduced that predictive accuracy does not differ significantly for the 41-feature and 32-feature 5NN classifiers. The 5NN classifier TPRATE values for the classes NORMAL and DOS do not differ significantly. There are significant differences in the 5NN classifier TPRATE values for the classes PROBE, R2L and U2R. The 10-fold cross validation results for the See5 classifiers indicate that the TPRATE values for the majority classes (NORMAL, DOS, PROBE) are nearly identical. The See5 TPRATE values for the 41-feature classifiers are significantly higher than those for the 32-feature classifier for the minority classes (R2L, U2R). Overall, it is difficult to establish differences in performance when the parent dataset class distribution of the KDD Cup 1999 dataset is used.

A training dataset of size 4500 randomly selected instances was created to with a class distribution that is very close to an equal class distribution. An equal class distribution is one where instances from all the classes appear in equal proportions in a dataset. One thousand instances were selected for each of the classes, NORMAL, DOS, PROBE and R2L. The 52 instances for the minority class (U2R) were bootstrapped to 500 instances. Test samples of 350 instances with an equal class distribution were created.

*Table 5.19: Statistical tests to compare the performance of KDD Cup 1999 classifiers for different feature subsets*

| Algorithm (training size) | Groups for the paired tests training sample size; number of features | | Student's paired t-test (9df) | | |
|---|---|---|---|---|---|
| | Group A (4500;41) | Group B (4500;32) | 95% CI of mean difference | p value (2 tailed) | Group A better than Group B? |
| 5NN (nearest neighbour) (size=4500) | All classes-A (73.5 ± 0.9) | All classes-B (69.9 ± 1.3) | [3.1, 4.1] | 0.000 | yes |
| | NORMAL-A (85.6 ± 3.2) | NORMAL-B (84.7 ± 3.2) | [0.1, 1.6] | 0.026 | yes |
| | DOS-A (67.3 ± 5.0) | DOS-B (67.1 ± 4.8) | [-0.4, 0.8] | 0.492 | no |
| | PROBE-A (95.9 ± 1.2) | PROBE-B (95.9 ± 1.2) | [-0.01, 0.03] | 0.342 | no |
| | R2L-A (73.1 ± 2.2) | R2L-B (70.3 ± 3.3) | [0.5, 5.1] | 0.020 | yes |
| | U2R-A (45.7 ± 0.0) | U2R-B (31.4 ± 0.0) | no variance | no variance | no variance |
| See5 (classification tree) (size=4500) | All classes-A (66.3 ± 1.2) | All classes-B (66.3 ± 1.2) | no variance | no variance | no |
| | NORMAL-A (97.3 ± 1.1 | NORMAL-B (97.3 ± 1.1) | no variance | no variance | no |
| | DOS-A (74.3 ± 6.2 | DOS-B (74.3 ± 6.2) | no variance | no variance | no |
| | PROBE-A (86.2 ± 2.2 | PROBE-B (86.2 ± 2.2) | no variance | no variance | no |
| | R2L-A (25.4 ± 2.3) | R2L-B (25.4 ± 2.3) | no variance | no variance | no |
| | U2R-A (48.6 ± 0.0) | U2R-B (48.6 ± 0.0) | no variance | no variance | no |

5NN and See5 classifiers were constructed from the training dataset of 4500 instances and tested on 10 test sets. Table 5.18 provides a summary of the classification results for the 41-feature and 32-feature classifiers. Table 5.19 shows

the results of the Student's paired samples t-test to compare the 41-feature and 32-feature classifiers. The results indicate that there is no statistically significant difference in the TPRATE values for both the 5NN and See5 classifiers the DOS and PROBE classes. The TPRATE values for the NORMAL and R2L classes are marginally higher for the 41 feature classifiers compared to the 32 features classifier. The 5NN classifier TPRATE for the U2R class is 14.3% higher for the 41-feature classifier compared to the 32-feature classifier. However, due to lack of variance, the paired t-test could not be applied. The See5 classifier TPRATE values for all classes for the 41-feature and 32-feature classifiers are equal. Again, due to the absence of variance, the paired samples t-test is not applicable.

The foregoing discussion led the author to hypothesise as follows: It is possible that features that are predictive of minority and severely under-represented classes will be eliminated when class-feature correlations are measured for all classes using instances randomly selected to represent the whole instance space. Such features are eliminated because the class-feature correlations cannot be reliably estimated. This is the case for the U2R class. This hypothesis was not tested for this thesis and is left for future work.

### 5.5.4 Classification results for the small datasets

Classifiers were constructed for the abalone3C and mushroom datasets to compare the predictive performance obtained when there is no feature selection and when the features selected by the decision rule-based algorithm are used. The results of table 5.13 show that for the abalone3C dataset the probes did not eliminate any features. For the mushroom dataset the number of features selected by the decision rule-based algorithm is the same as the number of features selected by the uniform-binary probe. 5NN and See5 classifiers were constructed for both datasets using 10-fold cross validation on randomly selected training samples from the datasets. The training sample size used for abalone3C was 3000 instances. Training sample sizes of 600 and 3000 instances were used for the mushroom dataset since training sample sizes of 3000 instances produced a ceiling effect (Cohen, 1995). A ceiling effect is encountered when the performance level of a system on a given task is so high that it is not possible to demonstrate performance improvements with any intervention (Cohen, 1995). Experiments with 10 test sets were not conducted as was done for the large datasets since the abalone3C and mushroom datasets have

balanced class distributions. Tables 5.20 show the classification results for the two datasets.

The predictive accuracy for each dataset is shown for all features and for features selected by the decision rule based search algorithm. The 5NN 8-feature classifiers provided a higher level of predictive accuracy compared to the 3-feature classifiers for the abalone3C dataset. Clearly the 95% confidence intervals for the mean predictive accuracy for these classifiers do not overlap.

*Table 5.20: Predictive accuracy for the small datasets based on the parent dataset class distribution*

| Dataset (training set size) | Classifier | Feature selection method (number of features) | Mean predictive accuracy & 95% CI of mean with 10-fold cross validation |
|---|---|---|---|
| Abalone3C (3000) | 5NN (nearest neighbours) | All features (8) | 59.8 ± 2.1 |
| | | Decision rule search (3) | 52.3 ± 3.2 |
| | See5 (classification tree) | All features (8) | 63.3 ± 1.3 |
| | | Decision rule (3) | 56.7 ± 1.6 |
| Mushroom (3000) | 5NN (nearest neighbours) | All features (22) | 97.9 ± 1.4 |
| | | Decision rule search (14) | 97.6 ± 2.1 |
| | See5 (classification tree) | All features (22) | 100 |
| | | Decision rule search (14) | 99.9 ± 0.0 |
| Mushroom (600) | 5NN (nearest neighbours) | All features (22) | 96.7 ± 2.1 |
| | | Decision rule search (14) | 95.8 ± 2.8 |
| | See5 (classification tree) | All features (22) | 99.2 ± 0.6 |
| | | Decision rule search (14) | 99.2 ± 0.9 |

The See5 8-feature classifiers also provided a higher level of predictive accuracy compared to the 3-feature classifiers for the abalone3C dataset. The 22-feature and 14-feature See5 and 5NN classifiers created with training sample sizes of 600 and 3000 provided the same level of very high predictive accuracy the mushroom dataset. Clearly the 95% confidence intervals of mean accuracy for the classifiers overlap. This is a ceiling effect which makes it difficult to make any conclusions on this dataset. The same ceiling effect was observed for both the mushroom 5NN 22-feature and 14-feature classifiers created with training sample sizes of 600 and 3000.

Table 5.21 gives the results of Student's independent samples t-test for means for the two datasets. The groups compared for each dataset were the 10-fold cross validation results when all features were used and when the features selected by the decision rule-based algorithm were used.

*Table 5.21: Statistical tests to compare the predictive performance of small dataset classifiers*

| Dataset (training sample size) | Classifier | Groups for independent samples tests; number of features | | Student's independent samples t-test (18df, equal variances assumed) | | |
|---|---|---|---|---|---|---|
| | | Group A (mean & CI) | Group B (mean & CI) | 95% CI of mean difference | p value (2 tails) | Group A better than Group B? |
| Abalone3C (3000) | 5NN | All-classes; 8 (59.8 ± 2.1) | All-classes; 3 (52.3 ± 3.2) | [3.4, 11.6] | 0.001 | yes |
| | | Class-young; 8 (74.1 ± 4.5 ) | class-young,3 (73.7 ± 6.2) | [-7.8, 8.7] | 0.912 | no |
| | | Class-middle; 8 (43.1 ± 4.9) | Class-middle;3 (51.1 ± 6.5) | [-16.8, 0.7] | 0.70 | no |
| | | Class-old; 8 (61.5 ± 3.4) | Class-old; 3 (37.2 ± 6.7) | [16.3, 32.4] | 0.000 | yes |
| | See5 | All-classes; 8 (63.3 ± 1.3) | All-classes; 3 (56.7 ± 1.6) | [4.3, 8.8] | 0.000 | yes |
| | | Class-young; 8 (74.0) | Class-young;3 (74.5) | based on arithmetic comparison of TPRATE | | no |
| | | Class-middle; 8 (47.6) | Class-middle;3 (20.4) | based on arithmetic comparison of TPRATE | | yes |
| | | Class-old; 8 (67.3) | Class-young;3 (72.7) | based on arithmetic comparison of TPRATE | | no |
| Mushroom (3000) | 5NN | All-classes; 22 (97.9 ± 1.4) | All-classes; 14 (97.6 ± 2.1) | [-2.5, 3.0] | 0.856 | no |
| | See5 | All-classes; 22 (100 ± 0.0) | All-classes; 14 (99.9 ± 0.0) | [-0.3, 0.1] | 0.230 | no |
| Mushroom (600) | 5NN | All-classes; 22 (96.7 ± 2.1) | All-classes; 14 (95.8 ± 2.8) | [-3.0, 4.7] | 0.652 | no |
| | See5 | All-classes; 22 (99.2 ± 0.6) | All-classes; 14 (99.2 ± 0.9) | [-1.1, 1.1] | 0.970 | no |

Student's paired samples t-test for means is not applicable here as cross validation tests were not paired. The class TPRATE values for the abalone3C classifiers are also given. TPRATE values are not given for the mushroom classes as there were no (interesting) differences in the TPRATE values for the different feature subsets. The statistical tests on the TPRATE values for the abalone3C 5NN classifiers indicate that the performance of the 8-feature and 3-feature classifiers does not differ significantly for the classes *young* and *middle*. For the 8-feature and 3-feature See5

classifiers the TPRATE values for the classes *young* and *old* do not differ significantly, but the TPRATE for the class *middle* is much higher for the 8-feature classifier.

The abalone3C dataset is a major challenge for feature subset search algorithms which attempt to maximise class-feature association and minimise feature-feature association. Table D.8 of appendix D gives the feature-feature correlation values for this dataset. It is clear from table D.8 that generally all the abalone3C features are strongly correlated with each other. The results of table 5.21 clearly indicate that the use of eight features (all features) provides a higher level of predictive performance compared to the use of a subset of the features. Obviously, the feature interactions for abalone3C have predictive power for the class variable.

## 5.6 Discussion

This section provides a discussion of the experimental results of this chapter. The discussion is divided into three subsections covering correlation measurement, feature subset selection and the problems associated with the measurement of class-feature correlations for feature selection. The recommendations for feature selection from large datasets are given in chapter 10 where the general discussion of the research findings is provided. Section 5.6.1 provides a discussion of correlation measures and feature ranking. A discussion of feature subset selection is given in section 5.6.2. Section 5.6.3 discusses the problems associated with the global measurement of class-feature and feature-feature correlations.

### 5.6.1 Correlation measures and feature ranking

When there are no outliers in the data and associations between variables are linear Pearson's correlation coefficients are suitable for measuring correlations (Wilcox, 2001) and determining feature ranking for feature selection. This was demonstrated with the abalone3C dataset experiments of section 5.3.2. When data contains outliers or when the association between variables is non-linear, robust measures of association will provide more accurate estimates of correlation values (Wilcox, 2001). The experimental results of section 5.3.2 and 5.3.4 demonstrated that Kendall's correlation coefficient is a more accurate measure of correlation compared to

Pearson's correlation coefficient for the large datasets used in the experiments. However, this does not exclude the possibility that Pearson's correlation coefficients could work well for some large datasets.

Robust measures for correlations (e.g. Kendall's tau) should be the preferred measure for ranking numeric features in feature selection for purposes of estimating the class-feature and feature-feature correlations, when the expense of removing outliers becomes prohibitive. Given a *d*-dimensional instance space and a sample size of *n* instances for correlation measurement, computation of Kendall's *tau* (and generally any robust correlation measure) has a larger time complexity of $O(d.n^2)$ compared to Pearson's $O(d.n)$ for the computation of class-feature correlations. The computation of feature-feature correlations has a time complexity of $O(d^2.n^2)$ for Kendall's *tau* (and generally any robust correlation measure) and $O(d^2.n)$ for Pearson's correlation coefficients. The extra computation time is worthwhile, since it allows more accurate feature rankings, even when moderately small sample sizes are used for the correlation estimates. If, on the other hand, there is sufficient computing power, then the winsorised Pearson's correlation coefficient (Wilcox, 2001) discussed in chapter 3 may be used for correlation measurement. Computation of winsorised Pearson's correlation coefficients will remove the effect of outliers but will not solve the problem of non-linear associations (Wilcox, 2001).

The experimental results further demonstrated that, a correlation coefficient is a random variable in the presence of sampling. This was demonstrated by the results of table 5.5 for the KDD Cup 1999 dataset and in fact Smyth (2001) has discussed this problem. When feature ranking and validation is based on correlations measured for one sample the feature ranking and number of selected features will vary from sample to sample. This was demonstrated in table 5.4. Based on the foregoing observations, feature rankings should not be based on a single sample, but rather on the mean values of the coefficients measured with many samples.

From a statistical perspective, using many (relatively) small randomly selected samples from very large datasets makes it possible to accurately and more efficiently estimate class-feature and feature-feature correlations. This was demonstrated in section 5.3.4. The samples used for feature ranking for large datasets should not be very small. When samples are small, the variability of the correlation coefficients can

change dramatically from sample to sample causing the confidence intervals of the mean correlation coefficients to be wide. It then becomes difficult to trust the feature ranking even when the rankings are based on mean values. This problem is demonstrated in table D.1 of appendix D for the forest cover type dataset samples of size 100.

Probes (fake variables) provide useful information for elimination of irrelevant features. It was empirically found that the three probes used generally eliminated the same (number of) features for the forest cover and KDD Cup 1999 datasets. The Gaussian probe did not work well for the mushroom dataset (all features are qualitative). However, the uniform and uniform-binary probe both eliminated nearly the same number of features for the mushroom dataset. None of the probes eliminated any features for the abalone3C dataset. Probes are random variables. The use of the confidence interval of the mean for a probe provided a better criterion for feature elimination. It was empirically found that probes also selected several features whose correlation values are not of practical significance, as defined by Cohen (1988). However, these features were found to have a small amount of predictive ability for the forest cover type dataset. Statistical significance with the t-test for means selected features that have predictive power. However, for all datasets the t-test eliminated features with no practical significance, even though these features have a small amount of predictive ability.

### 5.6.2 Feature subset selection

Feature selection methods that search for the best subset of features depend on an initial ranking of features. If this ranking is not accurate, then the search method will not select the best subset of features. The experimental results of section 5.3.4 demonstrated that the use of mean values of correlation coefficients for feature ranking and validation provided more accurate input values for feature subset selection. The experimental results of section 5.4.1 demonstrated that a mathematical function (merit measure) will not necessarily always precisely reflect the definition of what is required for feature subset selection. It was demonstrated that the search procedure can select irrelevant features in preference to more relevant features. The use of decision rules in place of a merit measure provides an alternative method of implementing the definition of feature relevance and redundancy to a search algorithm. The experimental results of section 5.4.2

demonstrated that irrelevant features were not selected when decision rules were used in place of mathematical functions.

The results of section 5.5.4 provided evidence to support the observation that apparently redundant features can have predictive power. This was the case for the abalone3C dataset. Cohen (1995:pg 68) has discussed two types of relationships between random variables. A *simple relationship* between two variables *X* and *Y* can be expressed in the form *X influences Y* (or *X* is correlated with *Y*). An *interaction relationship* involves three variables and is expressed in the form $X_i$ *and* $X_j$ *in concert influence Y*. The feature selection methods proposed in this chapter are not capable of detecting feature interactions.

### 5.6.3 Problems associated with the global measurement of correlations

For the empirical studies reported in section 5.4 some of the eliminated features were those features that are either good predictors for one or more of the classes in the dataset, or good predictors of some local areas of the instance space, or both. It was observed that one or more of the eliminated features for the KDD Cup 1999 dataset were those features that could be good predictors for the minority and severely under-represented class (U2R). It was hypothesised that if a large dataset is pre-processed to create clusters prior to feature selection then the above problems should not arise. The study of this hypothesis was left for future work.

## 5.7 Conclusions

The first research question that was addressed in this chapter was: *How can class-feature correlations be measured in order to produce a reliable ranking of features for a dataset?* The method that was studied and demonstrated to work well is to use many samples to measure correlations coupled with a robust measure of correlation. The samples should be large enough to avoid large variability in the measured correlation values as discussed in section 5.6.1. The mean values of the correlations should then be used to conduct validation and feature ranking for the prediction task.

The second research question that was addressed in this chapter is: *What methods of validation for feature correlations result in reliable feature selection?* The experimental results reported in this chapter have demonstrated that a comparison of mean values of class-probe and class-feature correlations provides useful information for more accurately determining which features have no relevance to the classification task. A second method of validation that was studied is the use of Student's t-test of means to determine the practical significance of class-feature correlations. The experimental results indicated that the t-test method of validation eliminated several features which have predictive power.

The third research question that was addressed in this chapter is: *How can domain-specific definitions of feature relevance be incorporated into feature selection procedures?* The method that was studied was to incorporate domain-specific definitions of the meaning of *insignificant*, *low, medium* and *high* correlation, in terms of the ranges of values that should be interpreted as *insignificant*, *low, medium* and *high* correlations. A new algorithm was designed, implemented and used for the selection of the best subset of features. The algorithm used decision rules based on the definitions of values for *insignificant*, *low, medium* and *high* correlations based on Cohen's (1988) definition. Experiments using the decision rule-based algorithm demonstrated that the algorithm selected good feature subsets which have global predictive power. The experimental results have also demonstrated that selecting features based on the measurement of class-feature correlations for samples obtained from all the regions of the instance space does not necessarily result in the selection of all the features that have predictive power. This problem was left for future research. The next three chapters provide a discussion of the studies that were conducted for training dataset selection.

# Chapter 6

# Methods for Dataset Selection and Base Model Aggregation

> 'Where you lead me I will follow, where you lead me I will follow, wherever you lead me I will follow. I will be with you always. Ngiyakuthanda moya oyingcwele…' (Benjamin Dube, 2007)

It was stated in chapter 2 that a limited amount of research on the combination of dataset partitioning, sampling and aggregate model construction from large datasets has been reported in the literature. To the author's knowledge, only one research effort by Chan and Stolfo (1998) has been reported. Chan and Stolfo's (1998) studies were aimed at improving predictive performance of 2-class datasets with skewed class distributions. It was argued in chapter 2 that, when large datasets are available, training datasets can be designed to achieve bias and variance reduction of the prediction error, without having to re-use training data. It was also argued in chapter 2 that more information is made available to the modeling process when a large amount of data is used in the training process.

The purpose of this chapter is to present the two proposed methods for combining dataset partitioning, sampling and aggregate model construction for large datasets. The methods used in the experiments for the evaluation of aggregate model performance are also presented. The proposed methods are specifically aimed at multi-class prediction tasks. The proposed methods were designed to support two types of base models: One-Versus-All (OVA) models and positive-Versus-negative (pVn) models. OVA modeling (Ooi et al, 2007; Rifkin & Klautau, 2004) is discussed in this chapter and the performance evaluation of this method is presented in chapter 7. pVn modeling is a new method of aggregate model construction, proposed in this thesis. pVn modeling is introduced in this chapter and the performance evaluation of this method is presented in chapter 8. The main difference between OVA and pVn modeling is that each OVA base model is designed to predict one of the *k* classes while each pVn base model is designed to predict more than one of the *k* classes.

It is claimed in this thesis that the proposed methods have the potential to provide a high level of predictive accuracy through the implementation of highly diverse and competent base models that are designed to provide high predictive performance when combined into an aggregate model. Syntactic diversity and high expertise of base models were discussed in chapter 2. Syntactic diversity refers to level of structural differences between the base models that constitute the aggregate model. High expertise refers to the level of predictive accuracy of the base models. The higher the predictive accuracy the higher the expertise. In the context of having large amounts of data available for the modeling process, the methods presented in this chapter are aimed at providing answers to the following questions:

*1. How should training datasets be designed in order to create base models that are syntactically diverse and highly expert at prediction for aggregate models?*

*2. How should training datasets for the base models be designed in order to achieve high accuracy for the aggregate model?*

The rest of this chapter is organised as follows: Problem decomposition for OVA and pVn modeling is discussed in section 6.1. A recap of methods for improving predictive performance is given in section 6.2. Methods for training and test dataset selection are discussed in section 6.3. Methods for creation and testing of OVA and pVn models are presented in sections 6.4. Section 6.5 provides a summary of the chapter.

## 6.1 Problem decomposition for OVA and pVn modeling

Problem decomposition is the process of converting a classification task into several classification sub-problems (Ooi et al, 2007; Dietterich & Kong, 1995). It was stated in the last section that problem decomposition has the potential to reduce the bias component of the prediction error (Dietterich & Kong, 1995). The two methods of problem decomposition that were studied are discussed in this section. The methods are *One-versus-all (OVA)* classification and *positive-versus-negative (pVn)* classification. OVA classification is discussed in section 6.1.1 and pVn classification is discussed in section 6.1.2.

### 6.1.1 Problem decomposition for OVA modeling

OVA classification (Ooi et al, 2007; Rifkin & Klautau, 2004) is a method of classification where a *k*-class prediction problem is decomposed into *k* sub-problems for classification. OVA classification is commonly used for (binary) support vector machines (SVMs) (Boser et al, 1992) for creating classifiers from multi-class datasets. Given a classification problem with *k* classes, $c_1,\ldots,c_k$, OVA classification involves the creation of *k* sub-problems $ova_1,\ldots, ova_k$. For each sub-problem, $ova_i$, the task is to create a base classifier, $OVA_i$, that differentiates between instances of class $c_i$ and instances that belong to all the other classes. In other words, each base classifier specialises in the prediction of one class. The base classifiers, $OVA_1,\ldots, OVA_k$, are combined into one aggregate model using the method of parallel aggregation that was discussed in section 2.2 of chapter 2.

OVA classification was selected as one of the problem decomposition methods to be studied for the following reasons: Firstly, OVA classification enables the creation of base models where each base model is an expert on classification for one specific class. Secondly, since each OVA classifier solves a 2-class problem, the training sample size required to achieve a high level of accuracy is reduced. This is an implication of the Probably Approximately Correct (PAC) learning theory which was discussed in section 2.4 of chapter 2. Equation (2.1) of section 2.4 specifies the theoretical relationship between the samples (complexity) size *n*, classification accuracy $1-\varepsilon$, and hypothesis space size |*H*| . For a fixed level of classification accuracy, reduction of the hypothesis space size |*H*| results in a reduction in the samples size required to achieve a given level of classification accuracy.

### 6.1.2 Problem decomposition for pVn modeling

*Positive-Versus-negative (pVn)* classification is a proposed modification of OVA classification proposed in this thesis. For pVn classification, each base model specializes in the prediction of a subset of classes, instead of just one class, as is the case for OVA classification. For *pVn* classification, a *k*-class prediction problem is

decomposed into a set of sub-problems, $pvn_1$..., $pvn_l$ , (*l*< *k*). For each sub-problem, the task is to create a base classifier $pVn_i$ which specializes in the prediction of *j* classes (*j* < *k*), which are a subset of the *k* classes. The *j* classes are referred to as the *positive* classes. All the other classes whose instances are included in the training dataset for the pVn model are collectively referred to as the *negative* classes. The name *positive-Versus-negative (pVn)* was used to represent the fact that a pVn base model can predict the positive classes in contrast to other classes which are simply treated as negative classes. The *pVn* models are combined into one aggregate model using the method of parallel aggregation.

The initial motivation for *pVn* modeling was as follows: If a multi-class problem has many classes, then many OVA base classifiers must be created. If on the other hand, each of the base models is specialized on more than one class, the number of base models to be created is reduced. After the OVA and pVn modeling experiments reported in chapters 7 and 8 were conducted, it became clear that pVn modeling solves other problems which are discussed in chapter 8.

## 6.2 Methods for improving predictive performance

The methods for improving predictive performance were discussed in detail in chapter 2. A summary of these methods is given in this section, and details are provided for the objectives for the methods that were studied for training dataset selection. Section 6.2.1 provides a discussion of the methods for bias and variance error reduction for small datasets. Section 6.2.2 provides a discussion of the methods for bias and variance error reduction for large datasets. High competence and syntactic diversity are discussed in section 6.2.3.

### 6.2.1 Reduction of bias and variance errors for small datasets

It was stated in chapter 2 that the three major factors that affect the predictive performance of a model are the bias, variance, and intrinsic error components of the prediction error. The bias of a predictive model reflects the error in the estimation process for the model (Giudici, 2003; Friedman, 1997; Geman et al, 1992). The variance reflects the sensitivity of the predictive model to the sample used to create

the model (Giudici, 2003; Friedman, 1997; Geman et al, 1992). The intrinsic error is the irreducible component of the prediction error. Various methods of bias and variance reduction were discussed in detail in section 2.11 of chapter 2. For small datasets, bias and variance reduction have been achieved primarily through two methods. The first method involves the creation of many base models through the re-use of the training data either through bootstrap sampling (Breiman, 1996) or re-using those training instances that are difficult to predict (Freund & Schapire, 1997). The second method involves the use of many base models, each with a different structure, in order to achieve syntactic diversity (Ho, 1998; Ali & Pazzani, 1996; Krogh & Vedelsby, 1995; Kwok & Carter, 1990; Hansen & Salamon, 1990). Additionally, various methods for base model aggregation have been studied (Ooi et al, 2007; Sun & Li, 2008; Ali & Pazzani, 1996).

### 6.2.2 Reduction of bias and variance errors for large datasets

The studies reported in the next two chapters were aimed at the design of aggregate models and the selection of training datasets for the base models, with the objective of reducing the bias and variance components of the prediction error. The training dataset selection methods used for aggregate modeling from small datasets were adapted in this thesis for the selection of training datasets when large amounts of data are available. While the dataset selection methods for small datasets have relied on the re-use of training data, there is generally no need to re-use training data for large datasets, except in those cases where one or more classes are severely under-represented in the dataset. In such cases, bootstrap sampling was employed for the studies of chapter 7 and 8, to increase the number of instances for the severely under-represented classes.

The following methods for bias and variance reduction were incorporated into the base model design and training dataset selection for the studies reported in chapter 7 and 8 of this thesis:

(1) Variance reduction through the use of a different training sample for each of the base models. The objective here was to use as much training data as possible in order to achieve a high level of coverage of the instance space.

(2) Variance reduction through the use of relatively small training samples (relative to the size of the large dataset) for each of the base models. The objective here was to reduce the effects of noise, and chance or phantom structure in the data (Smyth, 2001) as discussed in chapter 2.

(3) Bias and variance reduction through the use of training datasets with a sample composition aimed at providing a high level of coverage of the decision boundary regions of the instance space. The objective here was to provide as much data as possible for the regions where it is difficult to make correct predictions. A second objective was to ensure that the predictive performance does not degrade due to conflicting base model predictions when base models are combined into an aggregate model.

(4) Variance reduction through the selection of good feature subsets. These studies were reported in chapter 5.

(5) Bias reduction through the decomposition of *k*-class problems into 2-class problems as is done for OVA classification, and *j*-class (*j*<*k*) problems which was implemented using the proposed method of pVn classification.

### 6.2.3 High competence and syntactic diversity of base models

Several researchers have argued that syntactic diversity of base models may lead to a higher level of predictive accuracy for the aggregate model (Sun & Li, 2008; Ho, 1998; Ali & Pazzani, 1996; Krogh & Vedelsby, 1995; Kwok & Carter, 1990; Hansen & Salamon, 1990). Several researchers have also argued that a higher level of predictive performance may be achieved by making each member of the aggregate model as competent as possible (Sun & Li, 2008; Ho, 1998; Ali & Pazzani, 1996). Furthermore, Chan and Stolfo (1998) have demonstrated that the use of carefully designed samples from partitions, and creation of aggregate models from the samples, may result in an increased level of predictive performance of 2-class datasets with skewed class distributions.

It is the author's opinion that syntactic diversity and high competence of the base models, both lead to a reduction in the bias and variance components of the prediction error. Syntactic diversity should lead to a reduction in variance errors

(errors due to sampling variations) since the modeling process is conducted using a large amount of information on the data generating process. Syntactic diversity should also lead to a reduction in variance (sensitivity to the sample used for training), since several samples are used in the modeling process. High competence should lead to a reduction in bias since the methods used in the estimation process of a highly competent model will necessarily minimize the errors in the model estimation process.

For the achievement of syntactic diversity and high competence of the base models, the following methods were incorporated in the base model design and training dataset selection:

(1) Syntactic diversity through the use of base models where each base model predicts a different set of classes.

(2) Syntactic diversity through the use of a training sample with a different composition for the training samples of the other base models.

(3) High competence through the use of base models where each model is specialized on a simpler hypothesis space with fewer classes than for the single k-class model.

(4) High competence through the design of training samples to provide a high coverage of those regions of the instance space where correct prediction is difficult.

## 6.3 Design and selection of training and test datasets

This section provides a detailed discussion of the method of training dataset selection that were adopted for the experiments of chapters 7 and 8 for OVA and pVn modeling. The methods were designed to achieve three main objectives. The first objective was to maximise diversification of the base models. The second objective was to maximise individual expertise of the base models. The third objective was to ensure that when base models are combined into one aggregate model, the class confusion (occurrence of conflicting predictions) is minimised. Section 6.3.1 provides a discussion of the strategy that was adopted for base model design, training and test data selection, and model testing. Section 6.3.2 provides a discussion of the

motivation for the sample composition of the training and test datasets. Section 6.3.3 presents the methods employed for large dataset partitioning and sampling. Section 6.3.4 provides a discussion of the sampling process from the dataset partitions.

### 6.3.1 Strategy for dataset selection and model creation

Seven distinct steps were identified for purposes of predictive model design, training dataset selection, model creation, and testing. The steps are shown in figure 6.1. Step 1 involves the design of the base models. Step 2 involves the selection of the relevant feature set for each base model. Step 3 involves making a decision on the partitions that should be created, and then creating the partitions. Steps 2 and 3 could be interchanged. Feature selection is done to ensure that irrelevant features are removed in order to make the individual base models as competent as possible. Step 4 involves the selection of training data and test data. Step 5 involves the creation, validation and testing of each base model. Step 6 involves the combination of the predictions of the base models. Step 7 involves the measurement of the performance gains realized from using an aggregate model versus a single model.

Step 1: Design the base models
Step 2: Select the relevant feature sets for the training datasets
Step 3: Decide on, and create the dataset partitions
Step 4: Select the training datasets and test datasets from the partitions
  a. Create training and test data partions
  b. Create training datasets
  c. Create test datasets
Step 5: Create, validate and test each of the base models
Step 6: Combine the predictions of the base models
Step 7: Measure the performance gain for using an aggregate model versus a single model

*Figure 6.1 Steps for dataset partitioning, model creation and testing*

### 6.3.2 Motivation for the sampling methods

It is important to make a decision on the class distribution of the training set and test set samples when sampling is employed. Two alternatives exist: The first alternative is to select samples which have the same class distribution as the large dataset from which the samples are drawn. The assumption here is that the class distribution of

the large dataset is a true representation of the class distribution of the population. The second alternative, called *oversampling* (Berry & Linoff, 2000) is to use a different class distribution in the samples. One common motivation for employing *oversampling* is to increase the coverage of the minority classes that appear in the large dataset.

Berry and Linoff (2000) have cautioned against the use of *oversampling* and argued that *oversampling* changes the meaning of the scores (class posterior probabilities) that are assigned to the predictions by a probabilistic classifier. Recall from chapter 4 that a classification algorithm outputs a prediction for a test or query instance in the form of a pair (*class, score*). Berry and Linoff (2000) have advised that the *lift factor* which was discussed in chapter 4 should be interpreted with care when *oversampling* is used. Provost and Fawcett (2001) on the other hand, have cautioned against the assumption that the class distribution for the large dataset is always a true representation of the population class distribution. Provost and Fawcett (2001) have argued that, firstly, the true class distribution is rarely ever known precisely for most domains. Secondly, the class distribution for a large dataset is subject to change for many application domains. Provost and Fawcett (2001) have provided the example of fraud detection as a domain where the class distribution for large datasets changes often.

Recall from chapter 2 that boosting is a statistical method for modeling which aims to increase the coverage of those regions of the instance space where correct prediction is more difficult. Boosting will necessarily result in changes to the class distribution of the training dataset to make it different from the class distribution of the large dataset. Given the foregoing discussion of Berry and Linoff (2001), and Provost and Fawcett (2001), the author made a decision to use boosted training samples with a class distribution determined by the base model design. Test samples with an equal class distribution for all the classes were used. The motivation here was that the performance of single and aggregate models should be compared class by class for the same number of test instances of each class. The net result of the adopted approach is *oversampling.* For purposes of measuring model performance, calculation of *lift factors* was avoided and ROC analysis was used instead. Recall from chapter 4 that ROC analysis is not dependent on the class distribution of the training and test data.

### 6.3.3 Partitioning and sampling for dataset selection

The proposed methods of dataset selection involve the use of stratified sampling (Berry & Linoff, 2000; Rao, 2000) in order to obtain the required sample composition for the training and test datasets. Stratification is achieved through the creation of large dataset partitions (strata) with each partition (stratum) consisting of instances of one class. Training datasets are then created by taking random samples from each partition (stratum) with each class having a different level of representation in each of the training datasets. The level of representation of a given class is based on the objectives of model creation. These objectives are further elaborated on later in this chapter. The proposed method of training dataset selection was used to support two different types of aggregate classification models: OVA classification and *positive-Vs-negative (pVn)* classification.

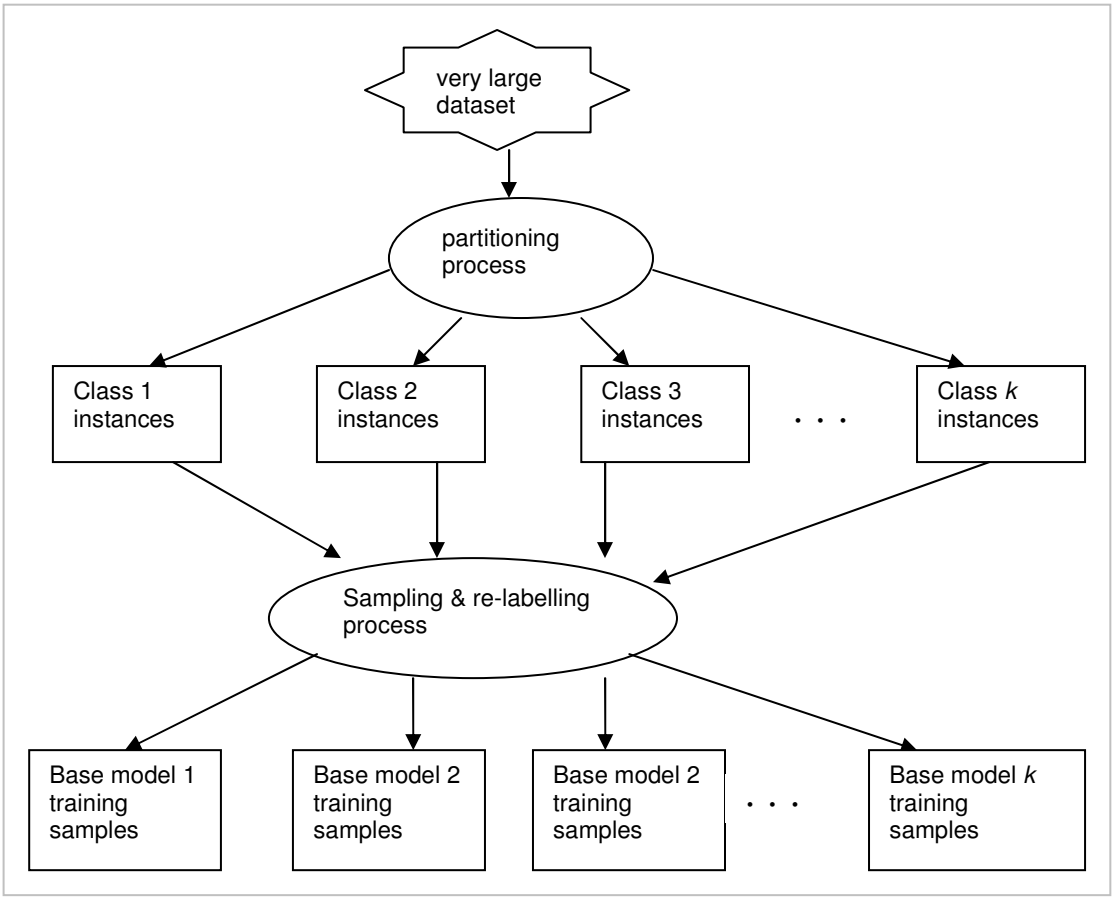


*Figure 6.2: Partitioning and sampling process for base model training dataset selection*

Figure 6.2 shows the approach that was studied for creating the partitions and obtaining samples from the partitions. This corresponds to steps 3 and 4 of figure 6.1. In step 3 the large dataset is partitioned into *k (k > 2)* partitions, where each

partition consists of instances of the same class. Step 4 involves various activities. The first activity is to sub-divide each partition into a training data and test data partition. The second activity is to create training datasets for the base models by selecting instances from the partition determined by the base model design. The third activity is to create test datasets by selecting instances from each partition. Simple random sampling was used for selecting instances from the partitions.

Each of the model training sets for OVA classification consists of instances from the class it is designed to be expert at predicting, as well as instances from some or all the other classes. Re-labeling was done to assign all the negative instances to a single class, so that each OVA training sample consists of instances of two classes. Each of the base model training sets for pVn classification is composed of instances from the *p* (positive) classes in equal proportions, and *n* (negative) classes in equal proportions. The proportions of the positive and negative samples were different. Re-labeling was done to assign all the negative instances to a single class. Details of how sampling was done are given in the next section.

### 6.3.4 Sampling from dataset partitions

Samples were taken from the partitions (strata) for the implementation of step 4 of figure 6.1. It is important to make decisions concerning the proportions of instances (of each class) in each of the base model training samples. When one-class partitions are created there may be great variation in the sizes of the partitions, with the partitions for the majority classes being very large and the partitions for the minority classes being very small. The number of training instances required from each one-class partition was calculated and then simple random sampling was used to obtain the instances from that partition. Details of the calculation of the required number of instances are given in chapters 7 and 8. A situation may arise when the partition size is smaller than the required number of instances for datasets with skewed class distributions. When this is the case, the solution that was used in the experiments of chapters 7 and 8 was to obtain a bootstrap sample from the partition (Rao, 2000). Bootstrap sampling (Breiman, 1996; Cohen, 1995) is a statistical method that is used to generate a large amount of data from a small dataset using simple random sampling with replacement (SRSWR) (Rao, 2000). Test data sets were created using simple random sampling from the test data partitions (strata). Each test dataset was created with an equal (balanced) class distribution.

## 6.4 Methods for creating and testing OVA and pVn models

Steps 5, 6 and 7 of figure 6.1 involve the creation and testing of the base models, aggregation of the base models, and testing of predictive performance of the aggregate models. The methods used to implement steps 5, 6 and 7 are presented in this section. Section 6.4.1 provides a discussion of the implementation of the OVA and pVn base models and a discussion of the outputs generated by the base models. Section 6.4.2 provides a discussion of the methods that were used to implement the aggregate models. The algorithms for model aggregation are presented in section 6.4.3. The experimental procedure for model aggregation is given in section 6.4.4. The methods for measuring performance gains due to aggregate models are presented in section 6.4.5.

### 6.4.1 Design and implementation of OVA and pVn base models

Base models were designed, created and tested for each dataset. The design objectives discussed in section 6.2 were adopted for each set of base models that make up an aggregate model. The details of OVA and pVn base model design and testing are given in chapters 7 and 8 respectively. Test datasets were created to include positive instances for the class(es) that a base model predicts, as well as negative instances from all the other classes. The same test sets were used for testing all the base models, as depicted in figure 6.3.

A predictive classification model may output a prediction *pred*, for a test or query instance in the form

$$pred = (c_i, conf_i) \tag{6.1}$$

where $c_i$ is the predicted class, $conf_i$ is the level of confidence that the test or query instance belongs to the predicted class and is defined as

$$conf_i = P_r(c_i \mid \mathbf{x}_q) \tag{6.2}$$

where $\mathbf{x}_q$ is the test or query instance and $P_r(c_i \mid \mathbf{x}_q)$ is the posterior probability that the instance $\mathbf{x}_q$ belongs to class $c_i$. The value of $conf_i$ is referred to as the score that is assigned by the predictive model for purposes of ROC and lift analysis (Giudici & Figini, 2009; Witten & Frank, 2005; Giudici, 2003; Berry & Linoff, 2000).

Each leaf node of a classification tree stores the posterior probability for prediction of each class at that node. The class with the highest posterior probability is the predicted class for test or query instances that land at that leaf node. The *See5Sam* tool which is part of the *See5* classification software (Quinlan, 2004) that was used for the experiments reported in chapters 5, 7 and 8 outputs a prediction as indicated in equation (6.1).

The 5NN classifier which was used for the experiments of chapters 5, 7 and 8, outputs a prediction *pred*, in the form of a triple

$$pred = (c_i, conf_i, recdist_i) \tag{6.3}$$

where $c_i$ is as defined above, and $conf_i$, the probability that the test or query instance belongs to the predicted class is defined as

$$conf_i = P_r(c_i) = \frac{|U|}{5} \tag{6.4}$$

where the numerator $|U|$ represents the count of nearest neighbours that belong to the predicted class and the denominator is the total number of neighbours used for deciding the predicted class (which is 5 for 5NN). The quantity $recdist_i$ is the sum of reciprocal distances for the neighbours that belong to the predicted class and is defined as

$$recdist_i = \sum_{x \in U} \frac{1}{dist(\boldsymbol{x}_q, \boldsymbol{x})} \tag{6.5}$$

where $dist(\mathbf{x}_q, \mathbf{x})$ is the Euclidean distance between the test or query instance and one of the nearest neighbours. The possible values for $conf_i$ are 0.4, 0.6, 0.8 and 1.0 for the 5NN classifier. These values correspond to the number of nearest

neighbours for the predicted (winning) class. For two nearest neighbours of the predicted class $conf_i$=0.4. For three nearest neighbours of the predicted class $conf_i$ =0.6. For four nearest neighbours of the predicted class $conf_i$=0.8. For five nearest neighbours of the predicted class $conf_i$=1.0.

### 6.4.2 Implementation of OVA and pVn aggregate models

When multiple base models are used, each model will declare a given test or query instance as belonging to a class $c_i$. The purpose of the aggregation step is to examine all the predictions of the individual base models and select that class with the strongest supporting evidence. The parallel method of aggregation, discussed in section 2.2.5, was used in the experiments. Recall that all the base model predictions for parallel aggregation are considered at the same time and the best prediction is selected based on the level of confidence in the prediction. The methods for combining base model predictions when each base model is capable of predicting any of the *k* classes for a prediction task were discussed in section 2.2.5.

Recall that these methods include (1) *majority voting* (2) the *product rule* (3) the *sum rule* (4) the *max rule*, and (5) the *min rule*. The *product rule* and *sum rule* are not directly applicable to OVA and pVn base models for the following reasons: Since each OVA base model can predict only one of the *k* classes and each pVn base model can predict only a subset of the classes, it is not possible to have a meaningful majority vote for any given class. It is also not possible to generate a meaningful mathematically combined probabilistic score for each class when OVA or pVn base models are used. The *max rule* and *min rule* can however be applied to OVA and pVn base model predictions as discussed below.

When OVA or pVn base models assign a single score to each prediction, as is the case for the See5 algorithm, then the output of a parallel aggregation algorithm, based on the max rule, is a pair defined as

$$pred = (c_i^*, conf_i^*) \in \{(c_1, conf_1), ..., (c_j, conf_j)\},\ j \leq k \qquad (6.6)$$

where

$$conf_i^* = \max\{conf_1,...,conf_j\} \quad (6.7)$$

and *k* is the number of classes for the prediction task, and $c_i^*$ is the predicted class which has the largest value $conf_i^*$. Equations (6.6) and (6.7) are sufficient for determining the prediction for the See5 classification tree aggregate models. The domain of possible values for $conf_i$ is small for the 5NN models, having 0.4, 0.6, 0.8 and 1.0 as the possible values. The small domain of values results in a high probability of tied $conf_i$ values for the base model predictions. In order to break ties, equation (6.5) was used to compute *recdist* values and the tied prediction with the highest *recdist* value was selected as the best prediction for the 5NN aggregate model. The interpretation of the *recdist* values is as follows: If base model predictions have a tied $conf_i$ value, then select the model which used the shortest Euclidean distances to determine the predicted class. The output of the 5NN aggregation algorithm is a triple defined as:

$$pred = (c_i^*, conf_i^*, recdist_i^*) \in \{(c_1, conf_1, recdist_1),..., (c_j, conf_j, recdist_j)\} \quad (6.8)$$

where *pred*, *k*, $c_i^*$ and $conf_i^*$ have the same interpretation as before and $recdist_i^*$ is the reciprocal distance for the best tied or untied prediction. It should be noted that Ooi et al (2007) have used the *recdist* values as a measure of the level of confidence in a 5NN prediction. The problem with Ooi et al's (2007) approach is that *recdist* values do not have a straightforward interpretation for ROC analysis.

### 6.4.3 Algorithms for model aggregation

The algorithm of figure 6.3 was used to implement the combination (aggregation) decisions for the See5 OVA and pVn aggregate models using the max rule. A base model may predict class $c_i$ or the class 'other' to indicate that a test instance belongs to one of the other classes. The value $conf_i$ in figure 6.3 is the posterior probability $P_r(c_i \mid \boldsymbol{x}_q)$ for the predicted class as defined in equation (6.2) for See5. The value 'none' indicates that there was no valid prediction. That is, all base models predicted

the class 'other'. A base model predicts 'other' to indicate that the class for the query or test instance is not a class that it is designed to predict. In step 3 of the algorithm in figure 6.3, ties are broken randomly, since there is no other value that can be used to resolve a tie.

1. If only one base model predicts a class $c_i$, and all the other base models predict '**other**', then the prediction is $c_i$
2. If more than one base model predicts a class $c_i$, then select the class $c_i$ which is predicted with the largest value of $conf_i$.
3. If there is a tie on $conf_i$ between winning classes then break the tie randomly
4. If all base models predict the class '**other**', then the prediction is '**none**'

*Figure 6.3: Algorithm for combining See5 base model predictions*

It was stated in section 6.4.2 that the prevalence of tied predictions (tied on the $conf_i$ values) is high for the 5NN base models. The strategy that was used for the implementation of the algorithm that determines all the tied predictions involves the generation of the complete search space of all possible ties. The generation of all possible ties is a combinatorial search problem (Luger & Stubblefield, 1993) requiring the generation of the number of states given by

$$States = \sum_{j=1}^{k} (k-j)^j \tag{6.9}$$

where *k* is the number of classes for the prediction task. For prediction tasks with a small number for classes the combinatorial explosion of equation (6.9) does not pose a major problem. For example, a prediction task with 5 classes will have 22 possible tied predictions. The derivation of equation (6.9) is given in appendix E. Figure 6.4 gives the data structures and algorithms for the functions that were used to combine the 5NN base model predictions.

**Data structures:**

Prediction: a base model prediction stored as a tuple
*(modelname, modelnumber, predictedclass, score, recdist)*

State: a search space state which holds data on tied predictions in the form
*(tiedcount, tiedmodels, tiedscore, bestclass, bestrecdist)*
OPEN,CLOSED, CHILDREN: list used by *BreadthFirstGenerate* algorithm to generate the complete search space

TIED: List used to hold the states for predictions that are actually tied

PREDICTIONS: list to hold the predictions for all the models.

**Algorithm for CheckTies( ):**

1. Call *BreadthFirstGenerate( )* to generate the list of all states for possible tied predictions consisting of *j, j-1, …, 2* tied predictions. Save the states on the CLOSED list.
2. For each state on the CLOSED list:
   a. Check if there is a tie in the $conf_i$ scores for all the predictions in the state.
   b. If there is a tie, record the tied score, largest *recdist* and prediction with largest *recdist*
   c. Copy the state to the TIED list
3. Delete from TIED every state whose nodes are all contained in another state on TIED
4. Select *besttiedstate* as the state on TIED with the highest score. Break ties using *recipdist*
5. Return *besttiedstate*

**Algorithm for CombinePredictions( ):**

1. Assign a unique number to each of the *j* ($j \leq k$) base models
2. Store the predictions for the j base models in the PREDICTIONS list
3. If all base models predict the class 'other', then the prediction is 'none'
4. Check for 2-way tied predictions
5. If there are no 2-way ties
   select prediction with largest $conf_i$ score on the PREDICTIONS list as *bestprediction*
6. else
   a. Call *CheckTies( )* to search for the tied state with the largest number of predictions. Call this *besttiedstate*
   b. *besttiedpred* = prediction in *besttiedstate* with largest *recipdist*
   c. select prediction with largest score on the PREDICTIONS list. Call this *bestuntiedpred*
   d. if (score for *bestuntiedpred* > score for *besttiedpred)*
      *bestprediction* = *bestuntiedpred*
   e. *else*
      *bestprediction* = *besttiedpred*
7. Return *bestprediction*

*Figure 6.4: Algorithm for combining 5NN base model predictions*

The function *CheckTies( )* uses the *BreadthFirstGenerate( )* to generate all the possible ties for base models identified as 1,2,..,*j* (*j* ≤ *k*). The *BreadthFirstGenerate( )* algorithm is based on a breadth-first search strategy (Luger & Stubblefield, 1993) and is given in appendix E. For each possible tied state, if there is an actual tie on the $conf_i$ scores for the predictions, the state is recorded in the TIED list. The tied state with the highest score is then selected as the best tie. The function *CombinePredictions( )* places all predictions on the PREDICTIONS list. If all the base models predict the class 'other' then there is no valid prediction for the aggregate model. The function *CombinePredictions( )* checks if there are any 2-way ties (ties involving two predictions). If there are no 2-way ties then there cannot be any 3-way, 4-way or higher order ties. When there are no tied predictions, the prediction with the highest $conf_i$ score is selected as the prediction for the aggregate model. If 2-way ties exist, *CombinePredictions( )* calls *CheckTies( )* to locate the tied predictions with the highest $conf_i$ score. The $conf_i$ score for the tied predictions is then compared with the highest $conf_i$ score for untied predictions. If the tied predictions have a higher $conf_i$ score, the tied prediction with the highest value of *recdist* is selected as the aggregate model prediction.

### 6.4.4 Experimental procedure for testing aggregate models

The experimental set up for OVA and pVn base model aggregation is shown in figure 6.5. The base models shown in figure 6.5 may be either all OVA models or all pVn models. Ten test sets were used to measure model performance. Each test set was applied to each of the base models and the test (prediction) results were written to a text file. The test results for each test set were combined into a single file and then used as input to the algorithm for combining the predictions of the base model into one prediction for each test instance.

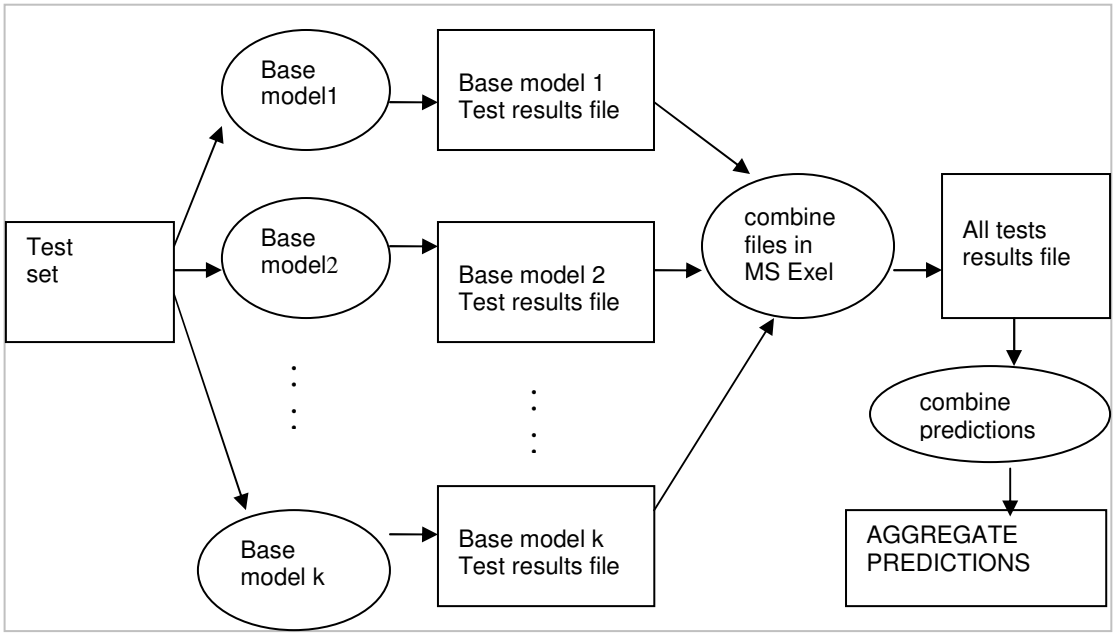


*Figure 6.5: Experimental method for aggregate model implementation for one test set*

## 6.4.5 Measurement of performance gains for OVA and pVn aggregate models

Prediction performance gains for aggregate models are typically established through comparison with single models (Ali & Pazzani, 1996). A detailed discussion of the statistical tests used to compare model performance was given in section 4.7 of chapter 4. Given two predictive models, $M_A$ and $M_B$, Student's paired sample t-test was used to establish whether model $M_A$ provides a higher level of predictive accuracy than model $M_B$. More precisely, if $\mu_A$ and $\mu_B$ are the mean values for predictive accuracy for models $M_A$ and $M_B$ respectively, the following hypotheses were tested: $H_0 : \mu_A - \mu_B = 0$ and $H_a : \mu_A - \mu_B \neq 0$. When the null hypothesis is rejected and the mean difference is positive, this gives an indication that the predictive performance of model $M_A$ is generally higher than the performance of model $M_B$. The mean difference provided by the paired samples t-test, gives an indication of the level of magnitude by which one model is better than the other.

Ali and Pazzani (1996) have conducted studies on different methods of combining the results from various classification models, and have proposed the following measures for computing the error reduction that is realised due to the use of aggregate model:

**Measure 1:** compute the error difference $error_D$ as

$$error_D = error_S - error_A \tag{6.10}$$

**Measure 2:** compute the error ratio $error_R$ as

$$error_R = error_A / error_S \tag{6.11}$$

where $error_S$ is the predictive error of a single model, and $error_A$ is the predictive error of the aggregate model obtained from the base models. The larger the error difference, the greater the error reduction due to the aggregate model. The larger the error ratio the greater the error reduction. Ali and Pazzani (1996) have advised that the error ratio is a better measure as it reflects the fact that it becomes very difficult to achieve error reduction using aggregate models when a single model has a very low prediction error. When the mean values of the errors are used in equation (6.10), the equation has a similar interpretation to the mean difference computed by the paired samples t-test.

For purposes of measuring the performance improvements due to the aggregate models, the Ali and Pazzani (1996) measures were re-interpreted by the author of this thesis as shown in table 6.1.

*Table 6.1 Interpretation of Ali and Pazzani (1996) measures*

| Ali & Pazzani Measure | Name used in thesis | Re-interpretation and computation of the measures used in the thesis based on accuracy and TPRATE: | |
|---|---|---|---|
| | | *accuracy = (1 – error)* | *FNRATE = (1 – TPRATE)* |
| Error difference = $error_S - error_A$ | *Diff(A,S)* | $accuracy_A - accuracy_S$ | $TPRATE_A - TPRATE_S$ |
| Error ratio = $error_A / error_S$ | *Ratio(A,S)* | $\frac{(accuracy_A - accuracy_S)}{(1 - accuracy_S)}$ | $\frac{(TPRATE_A - TPRATE_S)}{(1 - TPRATE_S)}$ |

The measure *Diff(A,S)* represents the performance increase in either the accuracy or TPRATE measures due to the aggregate model. The measure *Ratio(A,S)* represents the fraction (of maximum possible improvement) by which the aggregate model increases the accuracy or TPRATE. A value of *Ratio(A,S)* = 0 indicates that there is no increase in the accuracy or TPRATE. A value of *Ratio(A,S)* = 1 indicates that the

accuracy or TPRATE of the aggregate model is at its maximum value of 1 (or 100%). A negative value for *Ratio(A,S)* indicates deterioration in performance.

Student's paired samples t-test, the *Diff(A,S)* measure, and the *Ratio(A,S)* measure were all used to determine performance improvements due to the aggregate model for the experiments of chapters 7 and 8. Mean values for the accuracy and TPRATE values were used to compute the *Diff(A,S)* and the *Ratio(A,S)* measures.

ROC analysis and lift-factor analysis are commonly used to assess the performance of a predictive classification model and compare different models as discussed in section 4.7.3. It was also noted in section 6.3.2 that lift-factor analysis is difficult to interpret when oversampling is used as was done for this thesis. Multi-class ROC analysis (Fawcett, 2001, 2004, 2006; Provost & Domingos, 2001; Hand & Till, 2001) was used to analyse and compare the performance of the *k*-class single and aggregate models.

## 6.5 Chapter summary

The methods used for base model design and implementation, dataset partitioning and sampling, training dataset selection, base model aggregation, and performance measurement have been presented in this chapter. The next two chapters report the experimental results of the implementation of these methods for OVA and pVn modeling.

# Chapter 7

# Evaluation of Dataset Selection for One-Versus-All Aggregate Modeling

It was stated in chapter 6 that the proposed methods of training dataset selection were aimed at supporting the creation of aggregate models for multi-class prediction tasks. For such models, the two proposed methods for creating the base models are One-Versus-All (OVA) and positive-Versus-negative (pVn) classification. The experiments to study OVA base model design, training dataset selection for OVA base models and the performance of OVA base models and aggregate models are presented in this chapter. Questions 1 and 2 below were posed in chapter 6. The studies reported in this chapter are aimed at providing answers to these questions, in the context of OVA modeling.

*1. How should training datasets be designed in order to create base models that are syntactically diverse and highly expert at prediction for aggregate models?*

*2. How should training datasets for the base models be designed in order to achieve high accuracy for the aggregate model?*

This chapter is organised as follows: Section 7.1 provides a discussion of OVA modeling. Experiments to study 5NN OVA model performance and See5 OVA model performance are respectively discussed in sections 7.2 and 7.3. Sections 7.4 and 7.5 respectively provide a discussion and conclusions for the chapter.

## 7.1 OVA modeling

This section provides the motivation for OVA modeling. The methods for creating OVA aggregate models are also presented. The motivation for OVA modeling is discussed in section 7.1.1. The design of training samples for OVA base models is presented in section 7.1.2. The experimental procedure for this chapter is presented in section 7.1.3.

### 7.1.1 Motivation for OVA modeling

It was stated in chapter 6 that OVA classification was selected as one of the problem decomposition methods to be studied, for several reasons. Firstly, by definition OVA classification enables the creation of base models where each base model is an expert on classification for one specific class. Secondly, since each OVA classifier solves a 2-class problem, the training sample size required to achieve a high level of accuracy is reduced. This is an implication of the Probably Approximately Correct (PAC) learning theory as discussed in section 6.1 of chapter 6.

A third reason for selecting OVA modeling is as follows: It was stated in section 2.8 that increasing the amount of training data for the modeling process results in the reduction of the variance component of the prediction error. However, an excessively large training dataset results in overfitting and modeling of phantom structure. If several moderately sized training datasets are used for the modeling process, then the amount of training data is increased while at the same time overfitting is avoided. The use of OVA base models enables the above approach.

### 7.1.2 Sample composition for OVA base model training datasets

The methods for training sample selection for base models was discussed in section 6.3.3 and illustrated in figure 6.2. It should be highlighted here that each base model was created with a different training set from the other base models. Two options for sample composition for a dataset with *k* classes were studied for OVA base models design. The options cover the use of un-boosted and boosted OVA base models. The boosted OVA base models were designed based on information obtained from a confusion matrix for a single *k*-class model. The confusion matrix was discussed in section 4.7 of chapter 4. The two options that were studied are as follows:

**Option 1**

Use 50% of instances from the class $c_i$ that the OVA classifier specialises in and for each of the other classes use *50/(k-1)%* instances. This option results in the creation of un-boosted OVA base models. This option was used to test whether the increase in the quantity of training data through OVA modeling provides increased predictive performance.

**Option 2**

Use 50% of instances from the class $c_i$ that the OVA classifier specialises in and for each of the *j (j < k)* classes which are predominantly confused with class $c_i$, based on the values in the confusion matrix, use *50 / (j-1)%* instances. Recall that the confusion matrix was discussed in chapter 4. Option 2 results in the creation of boosted OVA base models. This option was used to test whether the use of boosting in addition to increasing the quantity of training data through OVA modeling provides additional increases in predictive performance.

### 7.1.3 Experiment design for the study of OVA modeling

Three categories of experiments on OVA aggregate modeling were conducted as follows:

(1) To compare the performance of *un-boosted* OVA models with single *k*-class models for both 5NN and See5 classification.
(2) To compare the performance of *boosted* OVA models with single *k*-class models for both 5NN and See5 classification.
(3) To compare the performance of *un-boosted* OVA models with *boosted* models for both 5NN and See5 classification.

The methods for OVA base model design and implementation, dataset partitioning and sampling, training dataset selection, model aggregation, and analysis of model performance were presented in chapter 6. These methods were used for the experiment categories listed above. The forest cover type and KDD Cup 1999 datasets were used for the experiments. The 5NN and See5 algorithms were used for the creation of the base models.

## 7.2 Experiments to study OVA models for 5NN classification

The empirical studies of 5NN OVA classification based on the experiment design of section 7.1.3 are discussed in this section. Section 7.2.1 reports the experiments to compare the predictive performance of single models with un-boosted 5NN OVA models. The process that was followed for the design of boosted OVA models is

discussed in section 7.2.2. Section 7.2.3 presents experimental results to compare the predictive performance of single, un-boosted and boosted 5NN OVA models.

### 7.2.1 Predictive performance of un-boosted 5NN OVA models

The predictive performance of un-boosted OVA base models and aggregate models is presented in this section. The training sets for the un-boosted OVA base models were designed using option 1 of section 7.1.3. A training sample size of 4000 was used for OVANORMAL, OVADOS, OVAPROBE, and OVAR2L for the KDD Cup 1999 base models. A training set size of 1000 was used for the OVAU2R model in order to limit the amount of bootstrap sampling for the U2R class. Table 7.1 gives the experimental results for the predictive performance of 5NN un-boosted base models for the forest cover type and KDD Cup 1999 datasets. Columns 3 and 4 respectively show the mean and 95% confidence interval for the TPRATE and TNRATE measures as percentages.

*Table 7.1: Predictive performance of 5NN OVA un-boosted base models*

| Dataset, Training sample size, Test set size | Base model name | Mean performance for base models | |
|---|---|---|---|
| | | Mean TPRATE% | Mean TNRATE% |
| Forest cover type (12000) (350 x 10) | OVA1-unboosted | 91.8 ± 2.5 | 85.0 ± 0.8 |
| | OVA2-unboosted | 83.8 ± 2.6 | 80.5 ± 1.1 |
| | OVA3-unboosted | 90.4 ± 1.1 | 85.3 ± 0.9 |
| | OVA4-unboosted | 95.6 ± 1.5 | 94.3 ± 0.6 |
| | OVA5-unboosted | 99.6 ± 0.5 | 90.8 ± 0.8 |
| | OVA 6-unboosted | 98.4 ± 1.0 | 84.6 ± 0.8 |
| | OVA7-unboosted | 99.2 ± 0.6 | 93.7 ± 0.5 |
| | | | |
| KDD Cup 1999 (4000) (350 x 10) | OVANORMAL-unboosted | 99.3 ± 0.6 | 73.0 ± 1.5 |
| | OVADOS-unboosted | 69.1 ± 4.5 | 97.8 ± 0.7 |
| | OVAPROBE-unboosted | 95.9 ± 1.2 | 88.5 ± 3.4 |
| | OVAR2L-unboosted | 76.7 ± 2.8 | 82.0 ± 1.6 |
| | OVAU2R-unboosted | 54.3 ± 0.0 | 97.7 ± 0.6 |

The results of table 7.1 indicate that the forest cover type base models have very high TPRATE and TNRATE values and are therefore highly competent at predicting the classes they are designed to predict. It remains to be seen if combining these highly competent base models into an aggregate model provides performance gains. The OVANORMAL and OVAPROBE base models for the KDD Cup 1999 dataset have very high TPRATE and TNRATE values while the OVADOS, OVAR2L and OVAU2R have much lower values.

The 5NN OVA base models were combined into aggregate models. The predictions of the individual 5NN OVA models on each test instance were combined into a single prediction using the algorithm of figure 6.4 presented in section 6.4.3. Recall that the algorithm in figure 6.4 uses the probabilistic scores assigned by the base models to determine the best prediction. When there is more than one prediction with the highest probabilistic score (tied scores) the distances to the nearest neighbours are used to break the tie. Single *k*-class models were created and also tested on the same instances as the aggregate models. The single 7-class model for forest cover type was created from a training dataset of 12000 instances with an equal class distribution for all the classes.

The KDD Cup 1999 single 5-class model was created from a training dataset of 4000 instances. The training dataset for the KDD Cup 1999 single model was composed of 500 instances for the class U2R and 3500 instances for the remaining four classes in equal proportions. Table 7.2 shows the results for the 5NN single and un-boosted OVA aggregate models for the forest cover type and KDD Cup 1999 datasets. The details for predictive accuracy and TPRATE values for the single and aggregate models for the forest cover type dataset are given in the appendix tables F.1 and F.2. The details for predictive accuracy and TPRATE values for the single and aggregate models for the KDD Cup 1999 dataset are given in the appendix tables F.9 and F.10.

*Table 7.2: Predictive performance of 5NN single and un-boosted OVA aggregate models*

| Dataset, (training set size), (test set size) | Class | Mean predictive performance of models | |
|---|---|---|---|
| | | Single model | un-boosted OVA aggregate model |
| | | Mean TPRATE% | Mean TPRATE% |
| Forest cover type (12000) (350 x 10) | ALL(accuracy) | 74.7 ± 1.0 | 80.5 ± 0.9 |
| | 1 | 62.8 ± 3.4 | 70.0 ± 4.3 |
| | 2 | 48.8 ± 2.8 | 58.4 ± 2.7 |
| | 3 | 56.8 ± 4.1 | 71.8 ± 1.9 |
| | 4 | 92.4 ± 1.8 | 89.8 ± 1.9 |
| | 5 | 91.2 ± 2.0 | 95.8 ± 3.1 |
| | 6 | 75.0 ± 2.1 | 80.8 ± 4.5 |
| | 7 | 96.0 ± 1.3 | 96.6 ± 0.6 |
| | | | |
| KDD Cup 1999 (4000) (350 x 10) | ALL (accuracy) | 68.5 ± 1.4 | 72.4 ±1.1 |
| | NORMAL | 84.4 ± 3.1 | 92.7 ±2.8 |
| | DOS | 66.3 ± 5.0 | 66.0 ±4.4 |
| | PROBE | 95.7 ± 1.2 | 95.2 ±1.0 |
| | R2L | 64.7 ± 3.6 | 65.4 ±3.6 |
| | U2R | 31.6 ± 0.3 | 42.6 ± 0.4 |

Student's paired samples t-test and the *Diff(A,S)* and *Ratio(A,S)* measures discussed in section 6.4 were used to compare the performance of the single models with that

of the aggregate models. Tables 7.3 and 7.4 respectively give the results of the statistical tests for the forest cover type and KDD Cup 1999 datasets. The paired t-test results of table 7.3 indicate that for the forest cover type dataset, the un-boosted aggregate model provides statistically significant increases in accuracy and the TPRATE values for five out of seven classes. The Diff(A,S) measure indicates an increase in accuracy of 5.8%. The increases in the class TPRATE values range between 4.6% and 15%. The Ratio(A,S) measure indicates a relative improvement of 0.2 for the accuracy and relative improvements that range between 0.2 and 0.5. Recall that the maximum improvement as measured by Ratio(A,S) is 1.0.

*Table 7.3: Statistical tests to compare the performance of 5NN single and un-boosted OVA aggregate models for forest cover type*

| Group names and mean accuracy / TPRATE% for 10 test sets | | Student's paired t-test (9 df) | | | Performance improvement measures | |
|---|---|---|---|---|---|---|
| Group A un-boosted aggregate model | Group S single model | 95% CI of mean difference | P value (2 tail) | Group A better than Group S? | Diff(A,S)% | Ratio(A,S) |
| All classes-A (80.5 ± 0.9) | All classes-S (74.7 ± 1.0) | [ 4.1, 7.5] | 0.000 | yes | 5.8 | 0.2 |
| Class1-A (70.0 ± 4.3) | Class1-S (62.8 ± 3.4) | [0.9, 13.5] | 0.029 | yes | 7.2 | 0.2 |
| Class2-A (58.4 ± 2.7) | Class2-S (48.8 ± 2.8) | [6.3, 12.9] | 0.000 | yes | 9.6 | 0.2 |
| Class3-A (71.8 ± 1.9) | Class3-S (56.8 ± 4.1) | [11.8, 18.3] | 0.000 | yes | 15.0 | 0.3 |
| Class4-A (89.8 ± 1.9) | Class4-S (92.4 ± 1.8) | [-4.6, -0.6] | 0.018 | no | -2.6 | -0.3 |
| Class5-A (95.8 ± 3.1) | Class5-S (91.2 ± 2.0) | [0.8, 8.4] | 0.022 | yes | 4.6 | 0.5 |
| Class6-A (80.8 ± 4.5) | Class6-S (75.0 ± 2.1) | [0.5, 11.1] | 0.036 | yes | 5.8 | 0.2 |
| Class7-A (96.6 ± 0.6) | Class7-S (96.0 ± 1.3) | [-1.2, 2.4] | 0.468 | no | 0.6 | 0.1 |

The paired t-test results of table 7.4 indicate for the KDD Cup 1999 dataset that the un-boosted aggregate model provides statistically significant increases in accuracy and the TPRATE values for two out of five classes. The Diff(A,S) measure indicates an increase in accuracy of 3.9%. The increases in the class TPRATE values are 8.3% for class NORMAL and 11% for class U2R. The Ratio(A,S) measure indicates a relative improvement of 0.1 for the accuracy and relative improvements of 0.2 for the class U2R and 0.5 for the class NORMAL. Overall, the use of OVA base models based on option 1 of section 7.1.2 for training dataset selection, provides significant improvements in predictive performance. Recall that the training set for each un-boosted OVA base model is composed of 50% for the class that the base model predicts and 50% for all the other classes combined.

*Table 7.4: Statistical tests to compare the performance of 5NN single and un-boosted OVA aggregate models for KDD Cup 1999*

| Group names and mean accuracy / TPRATE% for 10 test sets | | Student's paired t-test (9 df) | | | Performance improvement measures | |
|---|---|---|---|---|---|---|
| Group A Aggregate model | Group S Single model | 95% CI of mean difference | P value (2 tail) | Group A better than Group S? | *Diff(A,S)%* | *Ratio(A,S)* |
| All classes-A (72.4±1.1) | All classes-S (68.5±1.4) | [2.6, 5.0] | 0.000 | yes | 3.9 | 0.1 |
| NORMAL-A (92.7±2.8) | NORMAL-S (84.4±3.1) | [5.0, 11.6] | 0.000 | yes | 8.3 | 0.5 |
| DOS-A (66.0±4.4) | DOS-S (66.3±5.0) | [-2.6, 2.0] | 0.790 | no | -0.3 | 0.0 |
| PROBE-A (95.2±1.0) | PROBE-S (95.7±1.2) | [-1.4, 0.3] | 0.164 | no | -0.5 | -0.1 |
| R2L-A (65.4±3.6) | R2L-S (64.7±3.6) | [-1.9, 3.3] | 0.560 | no | 0.7 | 0.0 |
| U2R-A (42.6±0.4) | U2R-S (31.6±0.3) | [10.3, 11.8] | 0.000 | yes | 11.0 | 0.2 |

## 7.2.2 Design of boosted 5NN OVA base models

The results of section 7.2.1 have demonstrated that un-boosted OVA base models result in improvements in predictive performance. Boosting was discussed in sections 2.8.2 and 2.10.2 as a method of bias error reduction. Option 2 of section 7.1.2 involves the use of boosted base models. The author hypothesised that boosting of OVA base models should lead to further improvements in predictive performance compared to un-boosted models. Recall from chapter 2 that boosting is a statistical technique for directing the greatest effort towards those areas of the instance space where prediction is most difficult. It was further hypothesised that examination of the confusion matrix for the single *k*-class model should provide information about those areas of the instance space where correct prediction is most difficult to achieve. Confusion matrices were discussed in section 4.7.

It was further hypothesized that a predictive model makes incorrect decisions in those regions which are class boundary regions in the instance space. The term *confusion regions*, was used by the author to refer to these regions. *Confusion regions* were discussed in section 2.7. Table 7.5 shows the confusion matrix for the single k-class model for the forest cover type dataset based on 5 test sets. For simplicity of presentation only the counts for the off-diagonal cells are shown in the confusion matrix.

*Table 7.5: Confusion matrix for the 5NN single model for the forest cover type dataset*

| Single model confusion matrix, training size =12000, test set size = 250 per class | | | | | | | | Total confusion | |
|---|---|---|---|---|---|---|---|---|---|
| Actual class | Predicted class | | | | | | | | |
| | Class 1 | Class 2 | Class 3 | Class 4 | Class 5 | Class 6 | Class 7 | SUMS | PCNT |
| **Class 1** | | 38 | 2 | | 10 | 1 | 49 | **100** | 40 |
| **Class 2** | 38 | | 16 | 1 | 53 | 12 | 7 | **127** | 50.8 |
| **Class 3** | | 3 | | 41 | 6 | 65 | | **115** | 46 |
| **Class 4** | | | 8 | | | 14 | | **22** | 8.8 |
| **Class 5** | | 2 | 8 | | | 6 | | **16** | 6.4 |
| **Class 6** | | 4 | 37 | 25 | 10 | | | **76** | 30.4 |
| **Class 7** | 4 | 3 | | | | | | **7** | 2.8 |

A confusion matrix can be used to identify the existence or possible absence of a confusion region between the decision regions of any two classes. Identification of the confusion regions was done based on the confusion matrices, using the following simple deductive logic:

(1) Given two classes $c_i$ and $c_j$, if the entry $(c_i, c_j)$ in the confusion matrix is zero, then $c_i$ and $c_j$ do not have a common boundary in the instance space, and so, do not share a confusion region

(2) If the entry $(c_i, c_j)$ for two classes $c_i$ and $c_j$ is non-zero, then the two classes share a common decision boundary in the instance space, and the value in the cell $(c_i, c_j)$ indicates the intensity of the class confusion between the two classes.

The confusion matrix of table 7.5 indicates for the 5NN single model of the forest cover type dataset that class 1 gets predominantly confused with classes 2 and 7. On the other hand class 2, is never confused with classes 3, 4 or 6. Class 7 is predominantly confused with classes 1 and 2, but is never confused with classes 3, 4, 5 or 6. The following strategy could be used to reduce the confusion between class 1 and class 2: Select the training set sample for OVA1 with class 1 as the positive instances and classes 2 and 7 as negative instances. This should provide higher instance space coverage of the confusion regions between classes 1 and 2, and classes 1 and 7. In other words, the training sample for the OVA1 base model is boosted so that there are more instances for the classes that are difficult to separate.

Table 7.6 shows the sample composition that was used for the OVA base models for the forest cover type training datasets for purposes of reducing class confusion. The entries in the second column have the following interpretation. If the counts of

confusion matrix cells $(c_i, c_j)$ and $(c_j, c_i)$ are high then $c_i$ is predominantly confused with $c_j$. The rationale behind the training sample composition for base model $OVA_i$ was to ensure that training instances of the classes appearing in column 2 are included in the training dataset as negative instances. The number of positive and negative instances should be equal as was done for the un-boosted models.

*Table 7.6: 5NN training sample composition to reduce class confusion for forest cover type*

| Class | Predominantly Confused with | Training sample composition for OVA base model | |
|---|---|---|---|
| | | Percentage of positive instances | Percentage of negative instances |
| C1: | C2, C7 | C1: 50 | C2: 25, C7: 25 |
| C2 | C1,C3,C5 | C2: 49 | C1:17, C3:17, C5: 17 |
| C3 | C2,C4,C6 | C3: 49 | C2:17, C4:17, C6: 17 |
| C4 | C3,C6 | C4: 50 | C3: 25, C6: 25 |
| C5 | C2 | C5: 50 | C2: 50 |
| C6 | C3,C4,C5 | C6: 49 | C3:17, C4: 17, C5: 17 |
| C7 | C1,C2 | C7: 50 | C1: 25, C7: 25 |

Table 7.7 shows the confusion matrix for the KDD Cup 1999 dataset for the single model with a training set size of 4000 instances. Based on the information in the confusion matrix, training set samples for OVA base models were designed to provide a higher coverage of the confusion regions. The training set sample design is shown in table 7.8. It should be noted that the sample composition for the OVANORMAL base model is the same as for the un-boosted base model.

*Table 7.7: Confusion matrix for the 5NN single model for the KDD Cup 1999 dataset*

| Single model, training size = 4000, test set size = 350 instances per class | | | | | | | |
|---|---|---|---|---|---|---|---|
| | Predicted class | | | | | Total confusion | |
| Actual class | NORMAL | DOS | PROBE | R2L | U2R | SUM | PCNT |
| NORMAL | | 13 | **29** | 4 | 2 | 48 | 13.7 |
| DOS | 13 | | **91** | 14 | 3 | 121 | **34.6** |
| PROBE | 11 | | | | 4 | 15 | 4.3 |
| R2L | **107** | 1 | | | 6 | 114 | 32.6 |
| U2R | **170** | | | **69** | | 239 | **68.3** |

*Table 7.8: Training sample composition to reduce class confusion for 5NN models for KDD Cup 1999*

| Class | Predominantly Confused for: | Training sample composition for OVA base models | | |
|---|---|---|---|---|
| | | Percentage of positive instances | Percentage of negative instances | Training sample size |
| NORMAL | R2L,U2R, DOS,PROBE | NORMAL: 50 | R2L:12.5, U2R:12.5, DOS:12.5, PROBE:12.5 | 4000 |
| DOS | NORMAL,PROBE, R2L | DOS: 49 | NORMAL:17, PROBE: 17, R2L:17 | 4000 |
| PROBE | NORMAL, DOS, U2R | PROBE: 49 | NORMAL:17, DOS:17, U2R:17 | 4000 |
| R2L | NORMAL | R2L: 50 | NORMAL:50 | 4000 |
| U2R | NORMAL, R2L | U2R: 50 | NORMAL:25, R2L:25 | 1000 |

### 7.2.3 Predictive performance of boosted 5NN OVA models

Boosted 5NN base models were created based on the sample designs shown in tables 7.6 and 7.9 for the forest cover type and KDD Cup 1999 datasets. Implementation of the aggregate models based on the boosted base models as shown in table 7.6 and 7.8 did not result in performance improvements over the single models. However, the approach of using a combination of boosted and un-boosted base models resulted in performance improvements for the forest cover type aggregate model. The base models used for the boosted version of the OVA aggregate models for the forest cover type and KDD Cup 1999 datasets are given in table 7.9. The rationale for choosing boosted base models was as follows: If a boosted base model had a higher TPRATE value than the un-boosted version, the boosted version was selected. This was the case, for example, for the OVA4 forest cover type base model. If a boosted base model had a TPRATE comparable (equal) to that of the un-boosted version then the boosted base model was included in the aggregate model. If a performance improvement was realized, then the boosted base model was retained, otherwise it was replaced with the un-boosted version.

Table 7.10 shows the predictive performance results for the single, un-boosted and boosted OVA aggregate models based on the boosted OVA base models for the forest cover type and KDD Cup 1999 datasets. The details for predictive accuracy and TPRATE measure for the boosted OVA aggregate models for the forest cover type and KDD Cup 1999 datasets are respectively given in appendix tables F.3 and F.11

*Table 7.9: Predictive performance of 5NN OVA boosted base models*

| Dataset, Training sample size, Test set size | Base model name | Mean performance for base models | |
|---|---|---|---|
| | | TPRATE% | TNRATE% |
| Forest cover type (12000) (350 x 10) | OVA1-unboosted | 91.8 ± 2.5 | 85.0 ± 0.8 |
| | OVA2-unboosted | 83.8 ± 2.6 | 80.5 ± 1.1 |
| | OVA3-unboosted | 90.4 ± 1.1 | 85.3 ± 0.9 |
| | OVA 4-boosted | 100.0 ± 0.0 | 96.3 ± 0.6 |
| | OVA 5-boosted | 99.6 ± 0.5 | 89.0 ± 0.9 |
| | OVA 6-boosted | 94.2 ± 0.9 | 87.3 ± 1.3 |
| | OVA 7-unboosted | 99.2 ± 0.6 | 93.7 ± 0.5 |
| | | | |
| KDD Cup 1999 (4000) (350 x 10) | OVANORMAL-unboosted | 99.3 ± 0.6 | 73.0 ± 1.5 |
| | OVADOS-boosted | 68.3 ± 4.8 | 97.3 ± 0.8 |
| | OVAPROBE-unboosted | 95.9 ± 1.2 | 88.5 ± 3.4 |
| | OVAR2L-boosted | 68.2 ± 3.3 | 82.0 ± .2 |
| | OVAU2R-unboosted | 54.3 ±0.0 | 97.7 ±0.6 |

*Table 7.10: Predictive performance of 5NN single, un-boosted and boosted OVA aggregate models*

| Dataset, (training set size), (test set size) | Class | Mean predictive performance of models | | |
|---|---|---|---|---|
| | | single model | un-boosted OVA aggregate model | boosted OVA aggregate model |
| | | Mean TPRATE% | Mean TPRATE% | Mean TPRATE% |
| Forest cover type (12000) (350 x 10) | ALL(accuracy) | 74.7 ± 1.0 | 80.5 ± 0.9 | 82.0 ± 0.6 |
| | 1 | 62.8 ± 3.4 | 70.0 ± 4.3 | 70.0 ± 4.3 |
| | 2 | 48.8 ± 2.8 | 58.4 ± 2.7 | 62.0 ± 3.4 |
| | 3 | 56.8 ± 4.1 | 71.8 ± 1.9 | 71.0 ± 1.3 |
| | 4 | 92.4 ± 1.8 | 89.8 ± 1.9 | 100.0 ± 0.0 |
| | 5 | 91.2 ± 2.0 | 95.8 ± 3.1 | 97.0 ± 0.9 |
| | 6 | 75.0 ± 2.1 | 80.8 ± 4.5 | 77.6 ± 2.0 |
| | 7 | 96.0 ± 1.3 | 96.6 ± 0.6 | 96.6 ± 0.6 |
| | | | | |
| KDD Cup 1999 (4000) (350 x 10) | ALL (accuracy) | 68.5 ± 1.4 | 72.4 ±1.1 | 71.0 ±1.2 |
| | NORMAL | 84.4 ± 3.1 | 92.7 ±2.8 | 92.4 ±3.0 |
| | DOS | 66.3 ± 5.0 | 66.0 ±4.4 | 66.0 ±5.1 |
| | PROBE | 95.7 ± 1.2 | 95.2 ±1.0 | 95.4 ±1.2 |
| | R2L | 64.7 ± 3.6 | 65.4 ±3.6 | 60.9 ±3.8 |
| | U2R | 31.6 ± 0.3 | 42.6 ± 0.4 | 40.5 ±1.4 |

Table 7.11 shows the results of the statistical tests to compare the predictive performance of the single, un-boosted and boosted OVA aggregate models for the forest cover type dataset. The paired t-test results of table 7.11 compare the boosted OVA aggregate model with the single model. The results indicate that the boosted aggregate model provides statistically significant increases in accuracy for the forest cover type dataset. The boosted model also provides increased TPRATE values for six out of seven classes. The Diff(A,S) measure indicates an increase in accuracy of 7.3%. The increases in the class TPRATE values range between 2.6% and 14.2%. The Ratio(A,S) measure indicates a relative improvement of 0.3 for the accuracy and

relative improvements that range between 0.1 and 1.0. Recall that a Ratio(A,S) value of 1.0 indicates maximum improvement.

*Table 7.11: Statistical tests to compare the 5NN single, un-boosted and boosted OVA aggregate models for forest cover type*

| Group names and mean accuracy / TPRATE for 10 test sets | | Student's paired t-test (9 df) | | | Performance improvement measures | |
|---|---|---|---|---|---|---|
| Group A aggregate model | Group B single model | 95% CI of mean difference | P value (2 tail) | Group A better than Group B? | Diff(A,B)% | Ratio(A,B) |
| boosted All classes-A (82.0 ± 0.6) | single All classes-S (74.7 ± 1.0) | [5.8, 8.8] | 0.000 | yes | 7.3 | 0.3 |
| boosted Class1-A (70.0 ± 4.3) | single Class1-S (62.8 ± 3.4) | [0.9, 13.5] | 0.029 | yes | 7.2 | 0.2 |
| boosted Class2-A (62.0 ± 3.4) | single Class2-S (48.8 ± 2.8) | [9.8, 16.6] | 0.000 | yes | 13.2 | 0.3 |
| boosted Class3-A (71.0 ± 1.3) | single Class3-S (56.8 ± 4.1) | [9.8, 18.6] | 0.000 | yes | 14.2 | 0.3 |
| boosted Class4-A (100.0 ± 0.0) | single Class4-S (92.4 ± 1.8) | [5.5, 9.7] | 0.000 | yes | 7.6 | 1.0 |
| boosted Class5-A (97.0 ± 0.9) | single Class5-S (91.2 ± 2.0) | [3.8, 7.8] | 0.000 | yes | 5.8 | 0.7 |
| boosted Class6-A (77.6 ± 2.0) | single Class6-S (75.0 ± 2.1) | [1.2, 4.0] | 0.002 | yes | 2.6 | 0.1 |
| boosted Class7-A (96.6 ± 0.6) | single Class7-S (96.0 ± 1.3) | [-1.2,2.4] | 0.468 | no | 0.6 | 0.1 |
| | | | | | | |
| boosted All classes-A (82.0 ± 0.6) | un-boosted All classes-A (80.5 ± 0.9) | [0.5,2.7 ] | 0.009 | yes | 1.5 | 0.1 |
| boosted Class1-A (70.0 ± 4.3) | un-boosted Class1-A (70.0 ± 4.3) | no variance | no variance | no | 0.0 | 0.0 |
| boosted Class2-A (62.0 ± 3.4) | un-boosted Class2-A (58.4 ± 2.7) | [1.8,5.4] | 0.001 | yes | 3.6 | 0.1 |
| boosted Class3-A (71.0 ± 1.3) | un-boosted Class3-A (71.8 ± 1.9) | [-3.1,1.5] | 0.443 | no | -0.8 | 0.0 |
| boosted Class4-A (100.0 ± 0.0) | Class4-A (89.8 ± 1.9) | [8.0,12.4] | 0.000 | yes | 10.2 | 1.0 |
| boosted Class5-A (97.0 ± 0.9) | un-boosted Class5-A (95.8 ± 3.1) | [-2.9,5.3] | 0.520 | no | 1.2 | 0.3 |
| boosted Class6-A (77.6 ± 2.0) | un-boosted Class6-A (80.8 ± 4.5) | [-7.7,1.2] | 0.137 | no | -3.2 | -0.2 |
| boosted Class7-A (96.6 ± 0.6) | un-boosted Class7-A (96.6 ± 0.6) | no variance | no variance | no | 0.0 | 0.0 |

The paired t-tests to compare the boosted and un-boosted aggregate models indicate for the forest cover type dataset that the boosted aggregate model provides statistically significant increases in accuracy. The boosted model also provides increased TPRATE values for two out of seven classes. The Diff(A,S) measure indicates an additional increase in accuracy of 1.5%, due to boosting. The increases in the class TPRATE values are 3.6% for class 2 and 10.2% for class 4. The Ratio(A,S) measure indicates a relative improvement of 0.1 for the accuracy and relative improvements of 0.1 for class 1 and 1.0 for class 4.

Table 7.12 shows the results of the statistical tests to compare the predictive performance of the boosted and un-boosted OVA aggregate models for the KDD Cup 1999 dataset. A comparison of the test results of tables 7.4 and 7.12 indicates that the use of un-boosted 5NN OVA base models results in performance improvements over the single model for the KDD Cup 1999 dataset. However, there are no performance gains released due to boosting of 5NN OVA base models for the KDD Cup 1999 dataset.

*Table 7.12: Statistical tests to compare the 5NN single, un-boosted and boosted OVA aggregate models for KDD Cup 1999*

| Group names and mean accuracy / TPRATE for 10 test sets | | Student's paired t-test (9 df) | | | Performance improvement measures | |
|---|---|---|---|---|---|---|
| Group A aggregate model | Group B single model | 95% CI of mean difference | P value (2 tail) | Group A better than Group B? | *Diff(A,B)%* | *Ratio(A,B)* |
| boosted All classes-A (71.0±1.2) | un-boosted All classes-A (72.4±1.1) | [-2.1, -0.6] | 0.002 | no | -1.3 | 0.0 |
| boosted NORMAL-A (92.4±3.0) | un-boosted NORMAL-A (92.7±2.8) | [-0.9, 0.4] | 0.343 | no | -0.3 | 0.0 |
| boosted DOS-A (66.0±5.1) | un-boosted DOS-A (66.0±4.4) | [-1.5, 1.5] | 0.988 | no | 0.0 | 0.0 |
| boosted PROBE-A (95.4±1.2) | un-boosted PROBE-A (95.2±1.0) | [-0.2, 0.7] | 0.168 | no | 0.3 | 0.1 |
| boosted R2L-A (60.9±3.8) | un-boosted R2L-A (65.4±3.6) | [-7.4, -1.7] | 0.005 | no | -4.6 | -0.1 |
| boosted U2R-A (40.5±1.4) | un-boosted U2R-A (42.6±0.4) | [-4.1, -0.2] | 0.031 | no | -2.2 | 0.0 |

## 7.3 Experiments to study OVA models for See5 classification

The empirical studies of See5 OVA classification based on the experiment design presented in section 7.1.3 are discussed in this section. Section 7.3.1 reports the experiments to compare predictive performance of single models with un-boosted See5 OVA models. The design of boosted OVA models is discussed in section 7.3.2. Section 7.3.3 presents experimental results to compare predictive performance of single, un-boosted and boosted See5 OVA models.

### 7.3.1 Predictive performance of un-boosted See5 OVA models

The training datasets that were used for the un-boosted 5NN OVA base models were also used for the experiments to compare See5 single and un-boosted OVA aggregate models. Table 7.13 gives the experimental results for the predictive performance of See5 OVA un-boosted base models. Columns 3 and 4 respectively show the mean and 95% confidence interval for the TPRATE and TNRATE measures as percentages.

*Table 7.13: Predictive performance of See5 OVA un-boosted base models*

| Dataset, Training sample size, Test set size | Base model name | Mean performance for base models | |
|---|---|---|---|
| | | Mean TPRATE% | Mean TNRATE% |
| Forest cover type (12000) (350 x 10) | OVA1-unboosted | 92.6 ± 2.3 | 82.7 ± 0.7 |
| | OVA2-unboosted | 85.6 ± 1.7 | 82.0 ± 0.7 |
| | OVA 3-unboosted | 93.2 ± 1.7 | 86.8 ± 0.5 |
| | OVA 4-unboosted | 99.0 ± 0.9 | 95.9 ± 0.6 |
| | OVA 5-unboosted | 98.6 ± 1.3 | 93.7 ± 0.7 |
| | OVA 6-unboosted | 92.2 ± 1.9 | 88.0 ± 0.5 |
| | OVA 7-unboosted | 99.6 ± 0.5 | 96.1 ± 0.5 |
| | | | |
| KDD Cup 1999 (4000) (350 x 10) | OVANORMAL-unboosted | 98.4 ± 0.7 | 82.7 ± 0.9 |
| | OVADOS-unboosted | 53.2 ± 4.6 | 99.6 ± 0.1 |
| | OVAPROBE-unboosted | 88.6 ± 1.3 | 90.3 ± 1.0 |
| | OVAR2L-unboosted | 37.4 ± 3.6 | 88.9 ± 0.8 |
| | OVAU2R-unboosted | 65.7 ± 0.0 | 96.8 ± 0.8 |

The results of table 7.13 indicate that the forest cover type base models have very high TPRATE and TNRATE values and are therefore highly competent at predicting the classes they are designed to predict. It remains to be seen if combining these highly competent base models into an aggregate model provides performance gains.

The OVANORMAL and OVAPROBE base models for KDD Cup 1999 have high TPRATE and TNRATE values. While OVADOS, OVAR2L and OVAU2R have high TNRATE values, the TPRATE values for these base models are low.

The See5 OVA base models were combined into aggregate models. The predictions of the individual See5 OVA base models on each test instance were combined into a single prediction using the combination algorithm of figure 6.3 that was presented in section 6.4.3. Recall that the algorithm in figure 6.3 uses the probabilistic scores assigned by the base models to determine the best prediction. Single *k*-class models were created and also tested on the same instances as the aggregate models. Table 7.14 shows the results for the single and aggregate models for the forest cover type and KDD Cup 1999 datasets. The details for predictive accuracy and TPRATE measure for the forest cover type single and aggregate models are respectively given in appendix tables F.5 and F.6. The details for predictive accuracy and TPRATE measure for the KDD Cup 1999 single and aggregate models are respectively given in appendix tables F.13 and F.14.

*Table 7.14: Predictive performance of See5 single and un-boosted OVA aggregate models*

| Dataset, (training set size), (test set size) | Class | Mean predictive performance of models | |
|---|---|---|---|
| | | Single model | un-boosted OVA aggregate model |
| | | Mean TPRATE% | Mean TPRATE% |
| Forest cover type (12000) (350 x 10) | ALL(accuracy) | 76.9 ± 1.0 | 75.3 ± 0.7 |
| | 1 | 57.4 ± 3.4 | 60.6 ± 2.6 |
| | 2 | 63.8 ± 3.0 | 49.8 ± 3.6 |
| | 3 | 60.8 ± 3.3 | 64.0 ± 1.8 |
| | 4 | 96.8 ± 1.0 | 86.6 ± 1.7 |
| | 5 | 86.2 ± 2.4 | 94.4 ± 1.8 |
| | 6 | 77.8 ± 3.3 | 79.2 ± 2.0 |
| | 7 | 95.6 ± 1.6 | 92.8 ± 2.5 |
| | | | |
| KDD Cup 1999 (4000) (350 x 10) | ALL (accuracy) | 63.8 ± 1.3 | 63.3 ± 1.2 |
| | NORMAL | 86.0 ± 3.1 | 98.3 ± 0.7 |
| | DOS | 82.0 ± 3.8 | 50.1 ± 4.4 |
| | PROBE | 36.8 ± 2.4 | 88.0 ± 1.3 |
| | R2L | 37.7 ± 3.3 | 34.3 ± 3.3 |
| | U2R | 77.1 ± 0.0 | 45.7 ± 0.0 |

Student's paired samples t-test and the *Diff(A,S)* and *Ratio(A,S)* measures were used to compare the performance of the single models with that of the aggregate models. Tables 7.15 and 7.16 respectively give the results of the statistical tests for the forest cover type and KDD Cup 1999 datasets. The results of the paired samples t-tests for the forest cover type models indicate that there is a general degradation in performance due to the use of the un-boosted aggregate model. The accuracy and TPRATE values for 6 out of 7 classes are lower for the un-boosted OVA aggregate

model compared to the single model. The statistical tests of table 7.16 indicate that there is no overall improvement in accuracy due the un-boosted OVA aggregate model. However, there is a significant improvement in the TPRATE values for the NORMAL and PROBE classes.

*Table 7.15: Statistical tests to compare the performance of See5 single and un-boosted OVA aggregate models for forest cover type*

| Group names and mean accuracy / TPRATE for 10 test sets | | Student's paired t-test (9 df) | | | Performance improvement measures | |
|---|---|---|---|---|---|---|
| Group A un-boosted aggregate model | Group S single model | 95% CI of mean difference | P value (2 tail) | Group A better than Group S? | Diff(A,S) % | Ratio(A,S) |
| All classes-A (75.3±0.7) | All classes-S (76.9 ± 1.0) | [-2.5, -0.8] | 0.002 | no | -1.6 | -0.1 |
| Class1-A (60.6±2.6) | Class1-S (57.4 ± 3.4) | [-1.1, 7.5] | 0.125 | no | 3.2 | 0.1 |
| Class2-A (49.8±3.6) | Class2-S (63.8 ± 3.0) | [-17.6, -10.4] | 0.000 | no | -14 | -0.4 |
| Class3-A (64.0±1.8) | Class3-S (60.8 ± 3.3) | [-1.3, 7.7] | 0.141 | no | 3.2 | 0.1 |
| Class4-A (86.6±1.7) | Class4-S (96.8 ± 1.0) | [-12.5, -7.0] | 0.000 | no | -10.2 | -3.2 |
| Class5-A (94.4±1.8) | Class5-S (86.2 ± 2.4) | [5.8, 10.6] | 0.000 | yes | 8.2 | 0.6 |
| Class6-A (79.2±2.0) | Class6-S (77.8 ± 3.3) | [-2.1,4.9] | 0.390 | no | 1.4 | 0.1 |
| Class7-A (92.8±2.5) | Class7-S (95.6 ± 1.6) | [-5.3, -0.4] | 0.029 | no | -2.8 | -0.6 |

*Table 7.16: Statistical tests to compare the performance of See5 single and un-boosted OVA aggregate models for KDD Cup 1999*

| Group names and mean accuracy / TPRATE% for 10 test sets | | Student's paired t-test (9 df) | | | Performance improvement measures | |
|---|---|---|---|---|---|---|
| Group A un-boosted aggregate model | Group S single model | 95% CI of mean difference | P value (2 tail) | Group A better than Group S? | *Diff(A,S)%* | *Ratio(A,S)* |
| All classes-A (63.3 ± 1.2) | All classes-S (63.8 ± 1.3) | [-2.0, 0.9] | 0.430 | no | -0.5 | 0.01 |
| NORMAL-A (98.3 ± 0.7) | NORMAL-S (86.0 ± 3.1) | [9.0, 15.6] | 0.000 | yes | 12.3 | 0.9 |
| DOS-A (50.1 ± 4.4) | DOS-S (82.0 ± 3.8) | [-38.2, -25.5] | 0.000 | no | -31.9 | -1.8 |
| PROBE-A (88.0 ± 1.3) | PROBE-S (36.4 ± 2.4) | [48.2, 54.9] | 0.000 | yes | 52.6 | 0.8 |
| R2L-A (34.3 ± 3.3) | R2L-S (37.7 ± 3.3) | [-7.5, 0.5] | 0.082 | no | -3.4 | -0.1 |
| U2R-A (45.7 ± 0.0) | U2R-S (77.1 ± 0.0) | no variance | no variance | no | -31.4 | -1.4 |

## 7.3.2 Design of See5 boosted OVA base models

Table 7.17 and 7.18 show the confusion matrices for the single models created from the training samples with an equal class distribution for the forest cover type and KDD Cup 1999 datasets. For simplicity of presentation, only the off-diagonal counts are given. A comparison of the confusion matrices for forest cover type for the 5NN and See5 models reveals that the nature of the class confusion is fairly similar for both models. However, there a significant change in the level of confusion between the PROBE and U2R classes of the KDD Cup 1999 dataset. The 5NN OVA training sample designs given in table 7.6 for forest cover type were also used for the implementation of the See5 OVA base models. The sample design for KDD Cup 1999 See5 OVA base models is shown in table 7.19. It should be noted that the sample composition for the OVANORMAL, OVAPROBE and OVAR2L base models is the same as that for the un-boosted base models.

*Table 7.17: Confusion matrix for See5 classification tree single 7-class model for forest cover type*

| **See5 single model, training set size = 12000, test set = 250 per class** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Actual class** | **Predicted class** | | | | | | | **Total confusion** | |
| | **Class 1** | **Class 2** | **Class 3** | **Class 4** | **Class 5** | **Class 6** | **Class 7** | **SUMS** | **PCNT** |
| **Class 1** | | **60** | | | 3 | 2 | **38** | **103** | 41.2 |
| **Class 2** | **43** | | 5 | | **32** | 8 | 8 | **96** | 38.4 |
| **Class 3** | | | | **26** | **11** | **50** | | **87** | 34.8 |
| **Class 4** | | | 6 | | | | | **6** | 2.4 |
| **Class 5** | | **17** | 6 | | | 6 | | **29** | 11.6 |
| **Class 6** | | 4 | **30** | **23** | 3 | | | **60** | 24 |
| **Class 7** | **16** | | | | | | | **16** | 6.4 |

*Table 7.18: Confusion matrix for See5 classification tree single 5-class model for KDD Cup 1999*

| **See5 single model, training set = 4000, test set size = 350 instances per class** | | | | | | | |
|---|---|---|---|---|---|---|---|
| | Predicted class | | | | | Total confusion | |
| Actual class | NORMAL | DOS | PROBE | R2L | U2R | SUM | PCNT |
| NORMAL | | 1 | **30** | 11 | 1 | 43 | 12.3 |
| DOS | **32** | | 15 | 10 | | 57 | 16.3 |
| PROBE | 4 | 17 | | | **198** | 219 | 62.6 |
| R2L | **185** | | 8 | | 20 | 213 | 60.9 |
| U2R | **70** | 10 | | | | 80 | 22.9 |

*Table 7.19: See 5 Training sample composition to reduce class confusion for KDD Cup 1999*

| Class | Predominantly Confused for: | Training sample composition for OVA base models | | |
|---|---|---|---|---|
| | | Percentage of positive instances | Percentage of negative instances | Training sample size |
| NORMAL | R2L,U2R, DOS,PROBE | NORMAL: 50 | R2L:12.5, U2R:12.5, DOS:12.5, PROBE:12.5 | 4000 |
| DOS | NORMAL,PROBE, R2L | DOS: 49 | NORMAL:17. PROBE: 17, R2L:17 | 4000 |
| PROBE | NORMAL, DOS, R2L,U2R | PROBE: 50 | NORMAL:12.5, DOS:12.5, R2L:12.5, U2R:12.5 | 4000 |
| R2L | NORMAL, DOS,U2R | R2L: 49 | NORMAL:17, DOS:17, U2R:17 | 4000 |
| U2R | NORMAL, DOS, PROBE, R2L | U2R: 50 | NORMAL:12.5, DOS:12.5, PROBE: 12.5, R2L:12.5 | 1000 |

The performance of the boosted base models and aggregate models is discussed in the next section.

## 7.3.3 Predictive performance of boosted See5 OVA models

Boosted See5 base models were created based on the training set designs of table 7.6 for forest cover type and table 7.19 for the KDD Cup 1999 dataset. The TPRATE and TNRATE values for the base models are given in table 7.20. A comparison of the un-boosted base models of table 7.9 and the boosted base models of table 7.20 reveals that the boosted base models generally have lower mean TPRATE values.

*Table 7.20: Predictive performance of See5 OVA boosted base models*

| Dataset, Training sample size, Test set size | Base model name | Mean performance for base models | |
|---|---|---|---|
| | | Mean TPRATE% | Mean TNRATE% |
| Forest cover type (12000) (350 x 10) | OVA1-boosted | 75.0 ± 2.9 | 92.7 ± 0.7 |
| | OVA2-boosted | 81.4 ± 1.8 | 83.3 ± 0.9 |
| | OVA3-boosted | 85.8 ± 2.4 | 91.8 ± 0.7 |
| | OVA4-boosted | 99.0 ± 0.7 | 97.5 ± 0.4 |
| | OVA5-boosted | 96.4 ± 1.4 | 90.4 ± 0.7 |
| | OVA6-boosted | 93.2 ± 1.2 | 91.3 ± 0.8 |
| | OVA7-boosted | 97.6 ± 1.1 | 98.3 ± 0.3 |
| | | | |
| KDD Cup 1999 (4000) (350 x 10) | OVANORMAL-unboosted | 99.3 ± 0.6 | 73.0 ± 1.5 |
| | OVADOS-boosted | 56.3 ± 4.3 | 88.5 ± 0.2 |
| | OVAPROBE -unboosted | 95.9 ± 1.2 | 88.5 ± 3.4 |
| | OVAR2L-boosted | 51.0 ± 4.4 | 88.2 ± 1.4 |
| | OVAU2R-unboosted | 54.3 ± 0.0 | 97.7 ± 0.6 |

Boosted aggregate models were created using the base models of table 7.20. Table 7.21 shows the predictive performance results for the See5 single, un-boosted and boosted OVA aggregate models for the forest cover type and KDD Cup 1999 datasets. The details for predictive accuracy and TPRATE measures for the forest cover type boosted aggregate model are given in appendix tables table F.7. The

details for predictive accuracy and TPRATE measures for the KDD Cup 1999 boosted aggregate model are given in appendix table F.15.Table 7.22 shows the results of the statistical tests to compare the predictive performance of the forest cover type single, un-boosted and boosted aggregate models.

Comparison of the test results of tables 7.15 and 7.22 indicates that there is degradation in performance when un-boosted OVA base models are combined into an aggregate model. However, comparison of the forest cover type single and boosted OVA aggregate models indicates that there are significant performance improvements in the accuracy and TPRATE values for 3 out of 7 classes. The Diff(A,S) measure indicates an increase of 2.5% in accuracy and increases of TPRATE values of 2.2% (class 7), 6.0% (class 2), and 7.6% (class 1).

*Table 7.21: Predictive performance of See5 single, un-boosted and boosted OVA aggregate models*

| Dataset, (training set size), (test set size) | Class | Mean predictive performance of models | | |
|---|---|---|---|---|
| | | single model | un-boosted OVA aggregate model | boosted OVA aggregate model |
| | | Mean TPRATE% | Mean TPRATE% | Mean TPRATE% |
| Forest cover type (12000) (350 x 10) | ALL(accuracy) | 76.9 ± 1.0 | 75.3 ± 0.7 | 79.4 ± 0.6 |
| | 1 | 57.4 ± 3.4 | 60.6 ± 2.6 | 65.0 ± 2.9 |
| | 2 | 63.8 ± 3.0 | 49.8 ± 3.6 | 69.8 ± 2.4 |
| | 3 | 60.8 ± 3.3 | 64.0 ± 1.8 | 63.2 ± 3.3 |
| | 4 | 96.8 ± 1.0 | 86.6 ± 1.7 | 95.4 ± 1.3 |
| | 5 | 86.2 ± 2.4 | 94.4 ± 1.8 | 88.4 ± 2.3 |
| | 6 | 77.8 ± 3.3 | 79.2 ± 2.0 | 76.0 ± 1.9 |
| | 7 | 95.6 ± 1.6 | 92.8 ± 2.5 | 97.8 ± 1.1 |
| | | | | |
| KDD Cup 1999 (4000) (350 x 10) | ALL (accuracy) | 63.8 ± 1.3 | 63.3 ± 1.2 | 61.7 ± 0.9 |
| | NORMAL | 86.0 ± 3.1 | 98.3 ± 0.7 | 99.2 ± 0.6 |
| | DOS | 82.0 ± 3.8 | 50.1 ± 4.4 | 56.3 ± 4.3 |
| | PROBE | 36.8 ± 2.4 | 88.0 ± 1.3 | 89.3 ± 1.4 |
| | R2L | 37.7 ± 3.3 | 34.3 ± 3.3 | 23.6 ± 3.4 |
| | U2R | 77.1 ± 0.0 | 45.7 ± 0.0 | 40.0 ± 0.0 |

Table 7.23 shows the results of the statistical tests to compare the predictive performance of the single and boosted aggregate models for KDD Cup 1999. The results show that the predictive accuracy of the aggregate model on all the classes combined is not better than that of the single model. Secondly, the TPRATE values of the aggregate model on the classes DOS, PROBE and R2L are lower than the TPRATE values of the single model on the same classes. However, the aggregate model provides significant improvements on the TPRATE values for the classes NORMAL and U2R. Overall, both the Student's paired t-test results and the *Diff(A,S)* and *Ratio(A,S)* measures demonstrate that there are no impressive gains to be

realized by using the aggregate model. This is in contrast to the forest cover type dataset where the aggregate model provides significant gains over the single model.

*Table 7.22: Statistical tests to compare the See5 single, un-boosted and boosted OVA aggregate models for forest cover type*

| Group names and mean accuracy / TPRATE for 10 test sets | | Student's paired t-test (9 df) | | | Performance improvement measures | |
|---|---|---|---|---|---|---|
| Group A model | Group B model | 95% CI of mean difference | P value (2 tail) | Group A better than Group B? | Diff(A,B)% | Ratio(A,B) |
| boosted All classes-A (79.4 ± 0.6) | single All classes-S (76.9 ± 1.0) | [1.6, 3.4] | 0.000 | yes | 2.5 | 0.1 |
| boosted Class1-A (65.0 ± 2.9) | single Class1-S (57.4 ± 3.4) | [3.1, 12.1] | 0.004 | yes | 7.6 | 0.2 |
| boosted Class2-A (69.8 ± 2.4) | single Class2-S (63.8 ± 3.0) | [2.4, 9.6] | 0.004 | yes | 6.0 | 0.2 |
| boosted Class3-A (63.2 ± 3.3) | single Class3-S (60.8 ± 3.3) | [-0.9, 5.7] | 0.132 | no | 2.4 | 0.1 |
| boosted Class4-A (95.4 ± 1.3) | single Class4-S (96.8 ± 1.0) | [-3.1, 0.3] | 0.088 | no | -1.4 | -0.4 |
| boosted Class5-A (88.4 ± 2.3) | single Class5-S (86.2 ± 2.4) | [-1.9, 6.3] | 0.258 | no | 2.2 | 0.2 |
| boosted Class6-A (76.0 ± 1.9 | single Class6-S (77.8 ± 3.3) | [-4.0, 0.4] | 0.096 | no | -1.8 | -0.1 |
| boosted Class7-A (97.8 ± 1.1) | single Class7-S (95.6 ± 1.6) | [0.6, 3.8] | 0.012 | yes | 2.2 | 0.5 |
| | | | | | | |
| boosted All classes-A (79.4 ± 0.6) | un-boosted All classes-A (75.3±0.7) | [ 3.7, 4.5] | 0.000 | yes | 4.1 | 0.2 |
| boosted Class1-A (65.0 ± 2.9) | un-boosted Class1-A (60.6±2.6) | [1.7, 7.1] | 0.005 | yes | 4.4 | 0.1 |
| boosted Class2-A (69.8 ± 2.4) | un-boosted Class2-A (49.8±3.6) | [16.5, 23.5] | 0.000 | yes | 20 | 0.4 |
| boosted Class3-A (63.2 ± 3.3) | un-boosted Class3-A (64.0±1.8) | [-4.3, 2.7] | 0.619 | no | -0.8 | 0.0 |
| boosted Class4-A (95.4 ± 1.3) | un-boosted Class4-A (86.6±1.7) | [7.4. 10.2] | 0.000 | yes | 8.8 | 0.7 |
| boosted Class5-A (88.4 ± 2.3) | un-boosted Class5-A (94.4±1.8) | [-9.0, -3.0] | 0.001 | no | -6.0 | -1.1 |
| boosted Class6-A (76.0 ± 1.9 | un-boosted Class6-A (79.2±2.0) | [-5.7, -0.8] | 0.016 | no | -3.2 | -0.2 |
| boosted Class7-A (97.8 ± 1.1) | un-boosted Class7-A (92.8±2.5) | [2.3, 7.7] | 0.002 | yes | 5.0 | 0.7 |

It was stated in sections 2.2.4 and 6.2.3 that syntactic diversity and high competence of base models should lead to performance improvements for an aggregate model. The statistical test results of table 7.16 indicate that the See5 un-boosted OVA aggregate models for the KDD Cup 1999 dataset did not provide a statistically significant increase in predictive accuracy. The statistical test results of table 7.23 indicate that the See5 boosted OVA aggregate model resulted in a statistically significant reduction in predictive accuracy. Two problems were observed for the See5 OVA aggregate models for the KDD Cup 1999 dataset. The first problem was that only two base models (OVANORMAL and OVAPROBE) had a high level of competence, based on the results of tables 7.13 and 7.20.

The second problem was that the prevalence of 'no prediction' was high for both the un-boosted and boosted aggregate models. Recall from section 6.4.3 that it is possible for all OVA base models to predict the class '**other**'. When this happens, then the aggregate model prediction is '**none**' to indicate that there is no valid prediction. The prevalence of '**none**' predictions for the un-boosted OVA aggregate model ranged between 11.4% and 13.4% on the ten test samples. Boosting had the desirable effect of reducing the '**none**' prediction to between 5.4% and 7.7%. However, the rate of incorrect predictions also increased in the boosted version of the model.

Both the See5 un-boosted and boosted base models for the forest cover type dataset had a high level of competence, based on the results of tables 7.13 and 7.20. The occurrence of '**none**' predictions was very low for the forest cover type aggregate models, varying from 0.3% to 1.4% for the un-boosted model and 0.6% to 1.7% for the boosted model. The reduction in predictive performance for the See5 un-boosted OVA aggregate model is due to the occurrence of '**none**' predictions and tied predictions which could not be resolved. The problem of unresolved tied predictions is further discussed in section 7.4.

*Table 7.23: Statistical tests to compare the See5 single and boosted OVA aggregate models for KDD Cup 1999*

| Group names and mean accuracy / TPRATE for 10 test sets | | Student's paired t-test (9 df) | | | Performance improvement measures | |
|---|---|---|---|---|---|---|
| Group A boosted aggregate model | Group S single model | 95% CI of mean difference | P value (2 tail) | Group A better than Group S? | *Diff(A,S)%* | *Ratio(A,S)* |
| All classes-A (61.7 ± 0.9) | All classes-S (63.8 ± 1.3) | [-3.6,0.8] | 0.008 | no | -2.1 | -0.01 |
| NORMAL-A (99.2 ± 0.6) | NORMAL-S (86.0 ± 3.1) | [9.9,16.4] | 0.000 | yes | 13.2 | 0.9 |
| DOS-A (56.3 ± 4.3) | DOS-S (82.0 ± 3.8) | [-32.6,-18.6] | 0.000 | no | -25.7 | -1.4 |
| PROBE-A (89.3 ± 1.4) | PROBE-S (36.4 ± 2.4) | [49.5,56.3] | 0.000 | yes | 52.9 | 0.8 |
| R2L-A (23.6 ± 3.4) | R2L-S (37.7 ± 3.3) | -18.3,-10.0] | 0.000 | no | -14.1 | -0.2 |
| U2R-A (40.0 ± 0.0) | U2R-S (77.1 ± 0.0) | no variance | no variance | no | -37.1 | -1.6 |

## 7.4 Discussion

OVA modeling was studied as a method of problem decomposition with a potential to reduce the bias variance components of the prediction error. It has been demonstrated through the experimental results of this chapter that highly competent and syntactically diverse base models can be obtained through OVA modeling. Recall from chapter 2 and section 6.2 that several researchers (e.g. Sun & Li, 2008; Ho, 1998; Ali & Pazzani, 1996; Krogh & Vedelsby, 1995; Kwok & Carter; 1990; Hansen & Salamon, 1990) have argued that high competence and syntactic diversity of base models lead to aggregate models with improved predictive performance. The experiments reported in this chapter were conducted in order to establish:

(1) Whether the use of OVA base models, each with a different training set, results in increased performance for an aggregate model.
(2) Whether the use of boosting in addition to OVA base models results in additional increased performance for the aggregate model.

Table 7.24 provides a summary of the conclusions from the OVA modeling experiments. The use of OVA modeling alone resulted in increased performance for the 5NN algorithm. The use of OVA modeling alone did not result in increased performance for the See5 algorithm. However, for the forest cover type dataset, the use of boosting in addition to OVA modeling resulted in increased performance for the See5 algorithm.

*Table 7.24: Summary of the conclusions from the OVA modeling experiments*

| Dataset | Algorithm | Is performance improvement realized for the aggregate model when the base models used are: | |
|---|---|---|---|
| | | un-boosted OVA? | boosted OVA? |
| Forest cover type | 5NN | yes | yes |
| | See5 | no | yes |
| KDD Cup 1999 | 5NN | yes | no |
| | See5 | no | no |

Recall that the combination algorithm for the 5NN aggregate models uses probabilistic scores as well as distances to the nearest neighbour in order to resolve tied predictions. On the other hand, the combination algorithm for See5 does not have a second measure available for resolving tied predictions, except to break ties randomly. It was observed by the author that even though the occurrence of tied predictions is rare for the See5 aggregate models, ties do occur. A sample of the output of the See5 combination algorithm is given in table 7.25 for the forest cover type un-boosted OVA aggregate model. Recall that an OVA base model predicts the one class it is designed to predict or it predicts the value 10 to represent 'other'. The instances in the first two rows are correctly predicted since there are no tied predictions with the highest score values. The third instance is incorrectly predicted as the tie between the class 1 and class 2 predictions cannot be correctly resolved.

*Table 7.25: Sample of the output for the See5 combination algorithm*

| OVA1 | score1 | OVA2 | score2 | OVA3 | score3 | OVA4 | score4 | OVA5 | score5 | OVA6 | score6 | OVA7 | score7 | predicted | actual | score |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.91 | 2 | 0.85 | 10 | 0.99 | 10 | 1 | 10 | 1 | 10 | 1 | 10 | 0.91 | 1 | **1** | 0.91 |
| 1 | 0.91 | 10 | 0.85 | 10 | 0.99 | 10 | 1 | 10 | 1 | 10 | 1 | 10 | 1 | 1 | **1** | 0.91 |
| 1 | 0.91 | 2 | 0.91 | 10 | 0.99 | 10 | 1 | 10 | 1 | 10 | 1 | 10 | 1 | 1 | **2** | 0.91 |

## 7.5 Conclusions

The first question posed in this chapter was: *How should training datasets be designed in order to create base models that are syntactically diverse and highly expert at prediction for aggregate models?* The experimental results have demonstrated that the use of OVA modeling results in base models that are highly expert in predicting instances in specific regions of the instance space. However, the experimental results also demonstrated that expertise of base models, as measured

in terms of the predictive accuracy of the individual models, is not always enough to guarantee high predictive performance of the aggregate model.

The second question was: *How should training datasets for the base models be designed in order to achieve high accuracy for the aggregate model?* The experimental results demonstrated that one limiting factor for the predictive performance of aggregate models, created through parallel combination of the base model predictions, is the level of conflicting predictions for the base models. The experimental results for the 5NN algorithm demonstrated that the use of un-boosted OVA aggregate models results in performance improvements. Recall that the algorithm that was used for the combination of predictions for the 5NN base models has the ability to resolve conflicting predictions which are tied on the scores.

The experiments also demonstrated that when training datasets for base models are selected with the objective of minimising conflicting predictions, a high level of predictive performance may be realised. This was the case for the forest cover type 5NN and See5 boosted OVA aggregate models. For the experiments reported in this chapter, the minimisation of class confusion was realised through boosting which was achieved through the selection of training datasets that provide a high coverage of the confusion regions for the classes. It was demonstrated that boosting can result in improvements to predictive performance when OVA base models have conflicting predictions.

Further studies of the proposed training dataset selection method are reported in the context of pVn modeling in the next chapter.

# Chapter 8

# Evaluation of Dataset Selection for Positive-Versus-Negative Aggregate Modeling

It was stated in chapter 6 that the proposed methods of training dataset selection were aimed at supporting the creation of aggregate models for multi-class prediction tasks. The last chapter presented an evaluation of OVA modeling. This chapter presents the experiments to study training dataset selection for positive-Versus-negative (pVn) models, a discussion of pVn model performance, and a comparison of predictive performance of single, OVA and pVn aggregate models. Recall that each pVn base model specialises in the prediction of a subset of the classes (the p-classes). Also recall that the following two questions were posed in chapter 6, and answers to these questions were provided in chapter 7 for OVA classification:

*1. How should training datasets be designed in order to create base models that are syntactically diverse and highly expert at prediction for aggregate models?*

*2. How should training datasets for the base models be designed in order to achieve high accuracy for the aggregate model?*

This chapter presents further studies for the purpose of answering the above questions in the context of pVn modeling. Section 8.1 provides a discussion of pVn modeling. Experiments to study 5NN pVn model performance and See5 pVn model performance are respectively discussed in sections 8.2 and 8.3. Section 8.4 provides a discussion of the statistical tests used to compare the predictive coherence of single, OVA and pVn models. Sections 8.5 and 8.6 respectively provide discussions and conclusions for the chapter.

## 8.1 pVn modeling

The motivation for pVn modeling is presented in this section. The methods used for the design of pVn base models, and the creation and testing of pVn base models and pVn aggregate models are also discussed. Section 8.1.1 provides a discussion of the motivation for pVn modeling. The methods used to design the base models are discussed in section 8.1.2. The experiment design for the study of pVn modeling is presented in section 8.1.3.

### 8.1.1 Motivation for pVn modeling

pVn classification is a proposed modification of OVA classification. The initial motivation for using pVn base classifiers was given in chapter 6. Briefly stated, pVn modeling results in a reduction of the number of base models in comparison to OVA modeling. A further motivation for pVn modeling is as follows: The experimental results of chapter 7 demonstrated that there are datasets for which OVA base models do not result in aggregate models that provide a higher level of predictive accuracy. This is the case, for example, for the KDD Cup 1999 dataset where only the un-boosted 5NN model showed a small improvement in performance. It is useful to compare aggregate models based on OVA classification and with aggregate models based on pVn classification in order to establish whether pVn base models can result in predictive performance which is better than that of a single model which can predict any one of $k$ ($k > 2$) classes.

### 8.1.2 Design of pVn base models

The following three questions need to be answered for pVn classification:

(1) What pVn models should be created?
(2) Which classes should be the positive classes, and which classes should be the negative classes for each pVn base model?
(3) What should be the class distribution for the positive and negative classes for the training dataset of each pVn base model?

An algorithm was designed by the author to provide answers to questions 1 and 2 above. The algorithm uses the information in the confusion matrix for the single *k*-class model to determine the number of models and the class composition of each pVn base model. This algorithm is presented in the next section. The methods used to answer question 3 above are also discussed in the next section. After the decisions have been made on the composition of the pVn base models, the training datasets must be selected. The selection process that was presented in chapter 6, and depicted in figure 6.2, was followed. The feature subset used for all pVn base models was the same as that for the single model, for both the 5NN and classification tree models.

### 8.1.3 Experiment design for the study of pVn modeling

Experiments were conducted to study the effectiveness of the proposed pVn base model design. The forest cover type and KDD Cup 1999 datasets were used for the experiments. The 5NN and See5 algorithms were used for the creation of the base models. The base models were combined into aggregate models using the combination algorithm in figure 6.3 (for See5 base models) and figure 6.4 (for 5NN base models). The analysis of pVn model performance was conducted as follows:

(1) To compare the predictive performance of the single and pVn aggregate models for both 5NN and See5 classification.

(2) To compare the predictive coherence of single, OVA, and pVn aggregate models for both 5NN and See5 classification.

Models were compared on predictive performance using the accuracy and class TPRATE measures as discussed in section 6.4.5. Student's paired t-test and the *Diff(A,S)* and *Ratio(A,S)* measures discussed in section 6.4.5 were used to establish whether the aggregate models provide significant improvements in predictive performance.

## 8.2 Experiments to study pVn models for 5NN classification

This section provides a discussion of the experiments on pVn classification for the 5NN algorithm. Section 8.2.1 presents the methods for base model design and training dataset selection. Sections 8.2.2 and 8.2.3 respectively provide a discussion of the experimental results for base model and aggregate model performance.

### 8.2.1 Design of training datasets for 5NN pVn base models

Several interesting observations arose out of the experiments on OVA modeling. The following observations can be made for the forest cover type 5NN OVA aggregate models, based on table 7.6. A training sample of 50% class 1, 25% class 2 and 25% class 7 was used for the boosted OVA1 base model. A training sample of 25% class 1, 25% class 2, and 50% class 7 was used for the boosted OVA7 base model. For both models, the main reason behind this decision was due to the fact that there is significant class confusion between classes 1, 2 and 7. A question that comes to mind is: *Would the performance of one base model, based on a sample with an equal class distribution for classes 1, 2, and 7 provided better performance than that of the two OVA base models, OVA1 and OVA7?* In fact, the other OVA base models could be similarly combined based on the observations made from the confusion matrix of table 7.5.

A structure that was referred to as a *confusion graph* was designed by the author for purposes of graphically representing the information in a confusion matrix. Figures 8.1 and 8.2 respectively show the confusion graphs for the forest cover type and KDD Cup 1999 5NN single *k*-class models that were presented in section 7.2. The nodes in a confusion graph represent the classes for the prediction task. The arc $(c_i, c_j)$ means that class $c_i$ is predicted as class $c_j$. That is, classes $c_i$ and $c_j$ share a confusion region. The number in brackets in a node indicates the number of arcs connected to the node. The value labelling an arc represents the level of confusion between classes $c_i$ and $c_j$. This value comes from cell $(c_i, c_j)$ of the confusion matrix. For simplicity of presentation, the arcs of the confusion graphs with values of 5 or less are shown as dashed lines and are not labeled.

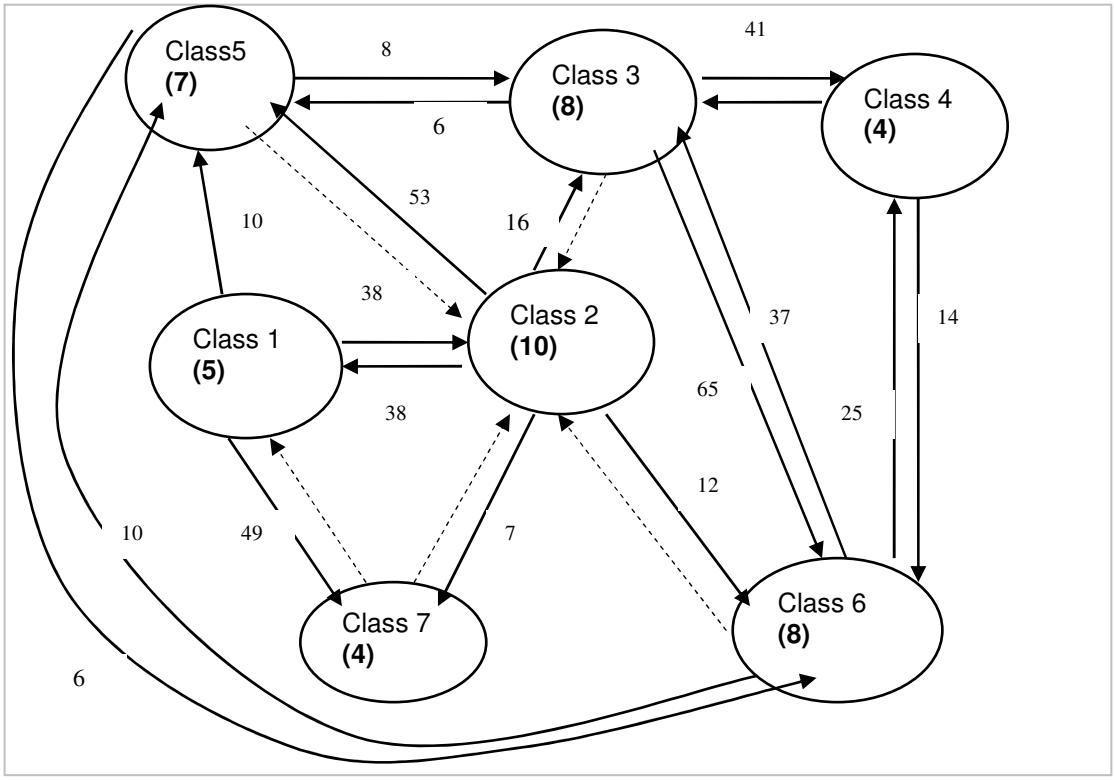


*Figure 8.1: Confusion graph for the 5NN single 7-class model for Forest cover type for training set size of 12000 instances*

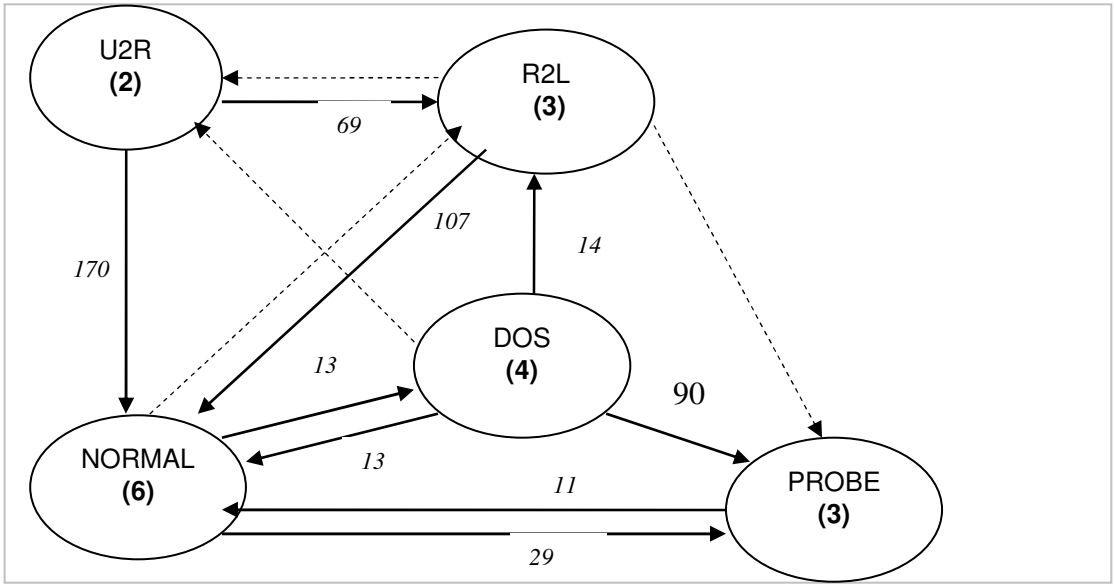


*Figure 8.2: Confusion graph for the 5NN single 5-class model for KDD Cup 1999 for training set size of 4000 instances*

The algorithm shown in figure 8.3 was designed by the author for selecting classes to include in each of the pVn base models. The objectives of the algorithm are as follows: When selecting the positive (p) classes for each base model, include those classes that share confusion regions. Exclude those classes that do not share

confusion regions with all the selected classes. The motivation here is to identify groups of classes which should be modelled together. Each model based on a subset of classes needs negative instances. The negative instances should be drawn from those classes that have a confusion region with at least one of the positive classes included in the model.

Table 8.1 provides a demonstration of the execution of the algorithm on the confusion graph of figure 8.1 for the forest cover type dataset. The last row of table 8.1 indicates that four pVn base models are identified by the algorithm. These models are: M346 for the positive classes 3, 4 and 6, M127 for the positive classes 1, 2 and 7, M125 for the positive classes 1, 2 and 5, and M2356 for the positive classes 2, 3, 5 and 6. The algorithm was also applied to the confusion graph for the KDD Cup 1999 single model shown in figure 8.2. The pVn base models that were identified are MNRU for the positive classes NORMAL, R2L and U2R, MNDR for the positive classes NORMAL, DOS and R2L, and MNDP for the positive classes NORMAL, DOS and PROBE.

A. Make a copy of the confusion graph

B. Repeat

Step 1: Select the node with the lowest connectivity and call it $c_{i,\min}$ (break ties randomly)

Step 2: Create a set whose members are $c_{i,\min}$ plus all the nodes connected to the node $c_{i,\min}$

Step 3: Delete all the arcs connected to $c_{i,\min}$ and delete $c_{i,\min}$ from the graph

Until there are no arcs left in the graph

C. For each set created in step 2 of B above, if the set is a subset of another set, delete the set (since its nodes are already included in another set)

D. Assign each of the remaining sets as the positive classes for one model

E. For each model, determine the negative classes as follows: For each positive class in the model, add all classes that have an arc to the class node, but are not one of the positive classes of the model.

*Figure 8.3: Algorithm for class selection for the pVn base models*

Table 8.2 shows the training set composition that was used to study the predictive performance of the pVn base models identified by the algorithm in figure 8.3 for the forest cover type and KDD Cup 1999 datasets. Each base model was composed of

instances from the indicated classes and sample percentages for each class. The rationale behind the samples composition was to ensure that each of the positive classes has nearly the same number of instances as the other positive classes, and nearly the same number of instances as all the negative classes combined. The training sample size for the MNRU model was reduced to 1900 instances to avoid excessive bootstrapping of the U2R instances.

*Table 8.1: Trace of the class selection algorithm for the5NN forest cover type graph*

| Iterations for steps B1, B2 and B3 | | | |
|---|---|---|---|
| Iteration | B1: selected node | B2: created set | B3: deleted arcs and node |
| 1 | $c_{i,\min} = 4$ | class set: { 3, 4, 6 } | arcs: { 3→4, 4→3, 6→4, 4→6 }<br>node: 4 |
| 2 | $c_{i,\min} = 7$ | class set: {1, 2, 7 } | arcs: { 1→7, 7→1, 2→7, 7→2 }<br>node: 7 |
| 3 | $c_{i,\min} = 1$ | class set: {1, 2, 5 } | arcs: { 1→5, 1→2, 2→1}<br>node: 1 |
| 4 | $c_{i,\min} = 5$ | class set: { 2, 3, 5, 6 } | arcs: { 2→5, 5→2, 3→5, 5→3, 6→5, 5→6 }<br>node:5 |
| 5 | $c_{i,\min} = 6$ | class set: { 2,3,6 } | arcs: { 2→6, 6→2, 3→6, 6→3 }<br>node: 6 |
| 6 | $c_{i,\min} = 3$ | class set: { 2, 3 } | arcs: { 2→3, 3→2 }<br>node: 3 |
| Final results of iterations of steps B1, B2, B3: | | { { 3, 4, 6 }, {1, 2, 7 }, {1, 2, 5 }, { 2, 3, 5, 6 }, { 2,3,6 }, { 2, 3 } } | |
| Steps C, D and E | | | |
| Step | Action | Results | |
| C | Delete subsets of other sets | deleted sets: {2,3} and {2,3,6}<br>remaining sets: { { 3, 4, 6 }, {1, 2, 7 }, {1, 2, 5 }, { 2, 3, 5, 6 } } | |
| D | Assign positive classes | M346: positive classes = { 3, 4, 6 }<br>M127: positive classes = { 1, 2, 7 }<br>M125: positive classes = { 1, 2, 5 }<br>M2356: positive classes = { 2, 3, 5, 6 } | |
| E | Determine negative classes | M346: negative classes = { 2, 5 }<br>2 borders with 3, 5 borders with 3 and 6 | |
| | | M127: negative classes: = { 3, 5, 6 }<br>3 borders with 2, 5 borders with 1 and 2, 6 borders with 2 | |
| | | M125: negative classes = {3, 6, 7}<br>3 borders with 2, 6 borders with 5, 7 borders with 1 and 2 | |
| | | M2356: negative classes = {1, 4, 7}<br>1 borders with 2, 4 borders with 3 and 6, 7 borders with 2<br>(but confusion level is very low) | |
| Algorithm output | Model definitions | M346: positive classes = { 3, 4, 6 }; negative classes = { 2, 5 }<br>M127: positive classes = {1, 2, 7 }; negative classes= { 3, 5, 6 }<br>M125: positive classes = {1, 2, 5 }; negative classes = {3, 6, 7 }<br>M2356: positive classes = { 2, 3, 5, 6 }; negative classes= { 1, 4}<br>class 7 ignored | |

*Table 8.2: 5NN training set composition for the pVn base models for forest cover type and KDD Cup 1999*

| Dataset | Model ID | p (positive) classes | | n (negative classes) | | Training sample size |
|---|---|---|---|---|---|---|
| | | Classes used | sample percentage | classes used | sample percentage | |
| Forest cover type | M125 | C1,C2,C5 | 80: (27,27,26) | C3,C6,C7 | 20: (7,7,6) | 12000 |
| | M127 | C1,C2,C7 | 80: (27,27,26) | C3,C5,C6 | 20: (7,7,6) | |
| | M2356 | C2,C3,C5,C6 | 80: (20,20,20,20) | C1,C4 | 20: (10,10) | |
| | M346 | C2,C3,C6 | 80: (27,27,26) | C2,C5 | 20: (1,10) | |
| | | | | | | |
| KDD Cup 1999 | MNRU | NORMAL, R2L,U2R | 80: (27,27,26) | DOS, PROBE | 20: (10,10) | 1900 |
| | MNDR | NORMAL,DOS, R2L | 80: (27,27,26) | PROBE, U2R | 20: (10,10) | 4000 |
| | MNDP | NORMAL,DOS, PROBE | 80: (27,27,26) | R2L, U2R | 20: (10,10) | 4000 |

## 8.2.2 Predictive performance of the 5NN pVn base models

The performance of the 5NN pVn base models for the forest cover type and KDD Cup 1999 dataset is shown in table 8.3. Columns 3 and 4 of table 8.3 show the mean TPRATE and mean TNRATE values for the base models. The TPRATE in this context is the predictive accuracy on the test instances for the p-classes while the TNRATE is the predictive accuracy on the test instances for the n-classes.

*Table 8.3: Predictive performance of 5NN pVn base models*

| Dataset (Training size) (test size) | Base model ID | Base model performance | | single model performance |
|---|---|---|---|---|
| | | Mean TPRATE% (p instances) | Mean TNRATE% (n instances) | Mean TPRATE% for single model on p instances |
| Forest cover type (12000) (350 x 10) | M125 | 75.3 ± 2.3 | 85.1 ± 1.2 | 67.3 ± 7.3 |
| | M127 | 74.4 ± 1.4 | 91.6 ± 0.7 | 66.9 ± 7.2 |
| | M2356 | 57.9 ± 0.5 | 70.8 ± 3.5 | 67.1 ± 6.7 |
| | M346 | 81.3 ± 1.5 | 94.1 ± 0.7 | 72.3 ± 5.6 |
| | | | | |
| KDD Cup 1999 (4000) (350 x 10) | MNRU | 76.3 ±0.8 | 97.4 ±1.0 | 60.2 ± 8.1 |
| | MNDR | 88.8 ±1.7 | 71.1 ±1.1 | 71.8 ± 3.9 |
| | MNDP | 74.4 ±5.5 | 68.9 ±7.4 | 82.1 ± 4.9 |

Column 5 of table 8.3 shows the mean TPRATE values for the single 7-class model for forest cover type and the single 5-class model for KDD Cup 1999. The results of table 8.3 indicate that three out of four pVn models for forest cover type have a higher TPRATE value on the p-classes than for the single model. Two out of three pVn models for the KDD Cup 1999 dataset have a higher TPRATE value than the single 5-class model. It remains to be seen whether the aggregate model based on

these base models provide higher predictive performance compared to the single models.

### 8.2.3 Predictive performance of the 5NN pVn aggregate models

The pVn base models for forest cover type and KDD Cup 1999 were combined into aggregate models using the combination algorithm that was given in figure 6.4 of section 6.4.3. The experimental procedure that was used for aggregation was presented in section 6.4. Table 8.4 shows the results of the predictive performance of the 5NN aggregate model for the forest cover type pVn models. The details of predictive performance are given in table F.4. Table 8.4 also shows the results of the predictive performance of the 5NN single 7-class and OVA aggregate models of chapter 7 for the forest cover type dataset. Table 8.5 shows the results of the statistical tests used to compare the performance of the single 7-class model and the pVn aggregate model.

*Table 8.4: Mean Predictive performance of the 5NN single, OVA and pVn aggregate models for forest cover type*

| Class name | 5NN Mean accuracy / TPRATE% (10 test sets of size 350) | | | |
|---|---|---|---|---|
| | Single model | un-boosted OVA aggregate model | boosted OVA aggregate model | pVn aggregate model |
| All classes | 74.7 ± 1.0 | 80.5 ± 0.9 | 82.0 ± 0.6 | 78.6 ± 1.2 |
| 1 | 62.8 ± 3.4 | 70.0 ± 4.3 | 70.0 ± 4.3 | 67.8 ± 5.1 |
| 2 | 48.8 ± 2.8 | 58.4 ± 2.7 | 62.0 ± 3.4 | 57.8 ± 2.1 |
| 3 | 56.8 ± 4.1 | 71.8 ± 1.9 | 71.0 ± 1.3 | 65.0 ± 2.3 |
| 4 | 92.4 ± 1.8 | 89.8 ± 1.9 | 100.0 ± 0.0 | 97.0 ± 1.2 |
| 5 | 91.2 ± 2.0 | 95.8 ± 3.1 | 97.0 ± 0.9 | 94.2 ± 2.1 |
| 6 | 75.0 ± 2.1 | 80.8 ± 4.5 | 77.6 ± 2.0 | 75.0 ± 2.9 |
| 7 | 96.0 ± 1.3 | 96.6 ± 0.6 | 96.6 ± 0.6 | 93.2 ± 2.4 |

The results of Student's paired t-test and the *Diff(A,S)* and *Ratio(A,S)* performance improvement measures provide the following evidence: The pVn aggregate model has a higher level of performance compared to the single model. The aggregate model results in an accuracy increase of 3.9% for all classes combined. The *Diff(A,S)* measure indicates that the pVn aggregate model provides significant increases of 3.0% to 9 % on the TPRATE for four out of seven classes, namely classes 2, 3, 4 and 5. The *Ratio(A,S)* measure indicates increases between 0.2 and 0.6 for classes 2, 3, 4 and 5. However, for classes 1, 6 and 7 there are no statistically significant improvements in the TPRATE. The best 5NN OVA aggregate model for forest cover

type reported in chapter 7 provided a mean accuracy of 82.0±0.6 as shown in table 8.4. The mean accuracy of the pVn aggregate model was 78.6±1.2. This leads to the conclusion that both the OVA and pVn aggregate models can provide improvements in predictive performance for the forest cover type dataset.

*Table 8.5: Statistical tests to compare the performance for 5NN single and pVn aggregate models for forest cover type*

| Group names and mean accuracy / TPRATE% for 10 test samples | | Student's paired t-test (9 df) | | | Performance improvement measures | |
|---|---|---|---|---|---|---|
| Group A Aggregate model | Group S Single model | 95% CI of mean difference | P value (2 tail) | Group A better than Group S? | *Diff(A,S)%* | *Ratio(A,S)* |
| All classes-A (78.6±1.2) | All classes-S (74.7 ± 1.0) | [2.5, 5.2] | 0.000 | yes | 3.9 | 0.2 |
| Class1-A (67.8±5.1) | Class1-S (62.8 ± 3.4) | [-2.1, 12.1] | 0.146 | no | 5.0 | 0.1 |
| Class2-A (57.8±2.1) | Class2-S (48.8 ± 2.8) | [4.9, 13.1] | 0.001 | yes | 9.0 | 0.2 |
| Class3-A (65.0±2.3) | Class3-S (56.8 ± 4.1) | [3.7, 12.8] | 0.003 | yes | 8.2 | 0.2 |
| Class4-A (97.0±1.2) | Class4-S (92.4 ± 1.8) | [3.1, 6.1] | 0.000 | yes | 4.6 | 0.6 |
| Class5-A (94.2±2.1) | Class5-S (91.2 ± 2.0) | [1.5, 4.6] | 0.002 | yes | 3.0 | 0.3 |
| Class6-A (75.0±2.9) | Class6-S (75.0 ± 2.1) | [-1.8, 1.8] | 1.000 | no | 0.0 | 0.0 |
| Class7-A (93.2±2.4) | Class7-S (96.0 ± 1.3) | [-5.3, -0.4] | 0.029 | no | -2.8 | -0.7 |

Table 8.6 shows the results of the Predictive performance of the 5NN pVn aggregate model for the KDD Cup 1999 dataset. The detailed results are given in the appendix table F.12. The results for the single 5-class and OVA aggregate models of chapter 7 are also shown in table 8.6. Table 8.7 shows the results of the statistical tests used to compare the predictive performance of the single 5-class model and the pVn aggregate model. The results of Student's paired samples t-test clearly indicate that the pVn aggregate model performance is much higher than that of the single 5-class model. The pVn model provided an increase of 11.8% in the mean accuracy for all the classes. The *Ratio(A,S)* measure indicates an increase of 0.4. The *Diff(A,S)* measure indicates an increase in the TPRATE ranging between 2.7% and 31% for four out of five classes. The *Ratio(A,S)* measure indicates high increases of between 0.5 and 0.9.

*Table 8.6: Mean Predictive performance of single, OVA and pVn aggregate 5NN models for KDD Cup 1999*

| Class name | 5NN Mean accuracy / TPRATE% for 10 test sets of size 350 | | | |
|---|---|---|---|---|
| | Single model | un-boosted OVA aggregate model | boosted OVA aggregate model | pVn aggregate model |
| All classes | 68.5 ± 1.4 | 72.4 ± 1.1 | 71.0 ± 1.2 | 80.3 ± 1.1 |
| NORMAL | 84.4 ± 3.1 | 92.7 ± 2.8 | 92.4 ± 3.0 | 98.7 ± 0.9 |
| DOS | 66.3 ± 5.0 | 66.0 ± 4.4 | 66.0 ± 5.1 | 97.3 ± 1.7 |
| PROBE | 95.7 ± 1.2 | 95.2 ± 1.0 | 95.4 ± 1.2 | 98.4 ± 0.9 |
| R2L | 64.7 ± 3.6 | 65.4 ± 3.6 | 60.9 ± 3.8 | 81.4 ± 4.1 |
| U2R | 31.6 ± 0.3 | 42.6 ± 0.4 | 40.5 ± 1.4 | 25.7 ± 2.2 |

*Table 8.7 Statistical tests to compare the 5NN single and pVn aggregate models for KDD Cup 1999*

| Group name and mean accuracy / TPRATE% for 10 test sets | | Student's paired t-test (9 df) | | | Performance improvement measures | |
|---|---|---|---|---|---|---|
| Group A Aggregate model | Group S Single model | 95% CI of mean difference | p value (2 tail) | Group A better than Group S? | *Diff(A,S)%* | *Ratio(A,S)* |
| All classes-A (80.3 ±1.1) | All classes-S (68.5 ±1.4) | [10.1,13.4] | 0.000 | yes | 11.8 | 0.4 |
| NORMAL-A (98.7 ±0.9) | NORMAL-S (84.4 ±3.1) | [10.9,17.8] | 0.000 | yes | 14.3 | 0.9 |
| DOS-A (97.3 ± 1.7) | DOS-S (66.3 ±5.0) | [25.5,36.6] | 0.000 | yes | 31.0 | 0.9 |
| PROBE-A (98.4 ±0.9) | PROBE-S (95.7 ±1.2) | [1.2,4.2] | 0.002 | yes | 2.7 | 0.6 |
| R2L-A (81.4 ±4.1) | R2L-S (64.7 ±3.6) | [13.1,20.3] | 0.000 | yes | 16.7 | 0.5 |
| U2R-A (25.7 ±2.2) | U2R-S (31.6 ±0.3) | [-8.4,-3.3] | 0.001 | no | -5.9 | -0.1 |

In comparison to the KDD Cup 1999 OVA aggregate models of chapter 7, the best OVA aggregate model had a mean predictive accuracy of 72.4±1.1 as shown in table 8.6, while the pVn aggregate model has a mean predictive accuracy of 80.3±1.1. This comparison indicates that the pVn aggregate model has a much higher level of predictive performance. The foregoing observations provide evidence that pVn aggregate modeling can provide much higher performance improvements than OVA modeling.

## 8.3 Experiments to study pVn models for See5 classification

pVn aggregate modeling was also tested using the See5 classification tree algorithm. A discussion of the experiments and the predictive performance of the See5 base models and aggregate models for the forest cover type and KDD Cup 19999

datasets are provided in this section. The training dataset design for the base models is presented in section 8.3.1. The predictive performance results for the base models and aggregate models are respectively presented in sections 8.3.2 and 8.3.3.

### 8.3.1 Design of training datasets for pVn base models

The confusion graphs for the forest cover type and KDD Cup 1999 See5 single models are shown in figures 8.4 and 8.5 respectively. The algorithm in figure 8.3 was used to determine the class composition of the pVn classification tree models for both the forest cover type and the KDD Cup 1999 datasets. It became evident that the algorithm in figure 8.3 is not suitable for determining the class composition for the KDD Cup 1999 dataset because the confusion graph for the KDD Cup 1999 See5 single model is a maximally connected (fully interconnected) graph. When a maximally connected confusion graph is used as input to the algorithm of figure 8.3, the first iteration of step B will create a set of nodes which includes all the nodes in the graph. When step C is executed, all the sets of nodes created after the first iteration of step B will be deleted, since they will be subsets of the first set of nodes. A modification of the algorithm in figure 8.3 is given in figure 8.4. The motivation for the modification was to reduce the level of connectivity in the graph while at the same time retaining all the information about the regions with the highest levels of class confusion. The rationale behind step I of the algorithm in figure 8.4 is to ignore those regions that have low levels of confusion and favour those regions which have higher levels of class confusion.

The application of step I of the algorithm in figure 8.4 to the confusion graph for the KDD Cup 1999 dataset resulted in the confusion graph of figure 8.5. The algorithm in figure 8.3 was applied to the confusion graph for the forest cover type dataset. The modified algorithm in figure 8.6 was applied to the confusion graph for the KDD Cup 1999 dataset. The resulting pVn base model designs are shown in table 8.8. Column 2 of table 8.8 shows the names of the pVn models. Each model is identified by the positive classes it is designed to predict. The training sample composition for each pVn base model is also shown in table 8.8. The training sample sizes for the MNRU and MNPU base models were reduced to 1900 instances to avoid excessive bootstrapping of the U2R instances.

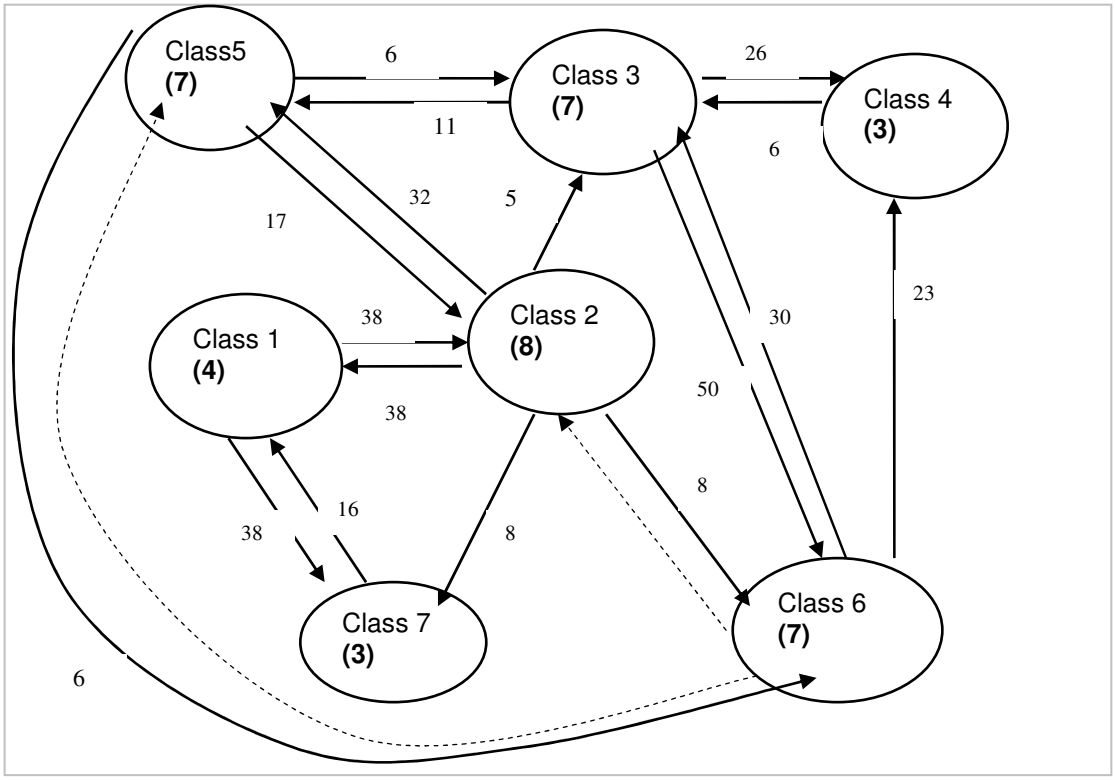


*Figure 8.4 Confusion graph for theSee5 single 7-class model for forest cover type for training set size of 12000 instances*

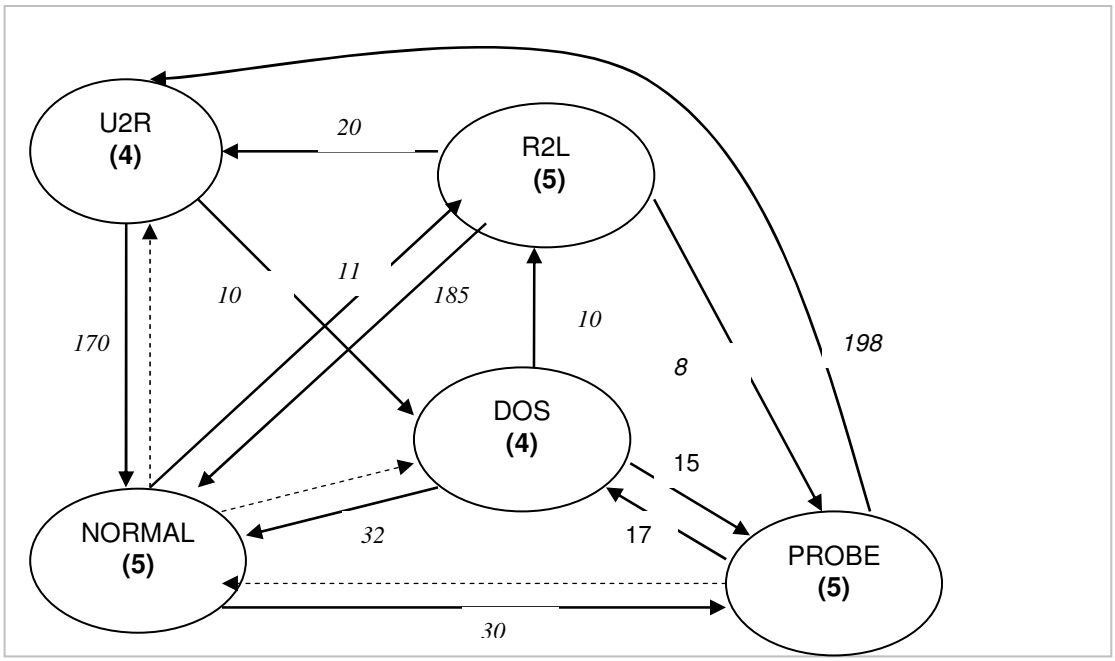


*Figure 8.5: Confusion graph for the See5 single 5-class model for KDD Cup 1999 for training set size of 4000 instances*

**I. Analyse the confusion graph as follows:**

If each node is fully connected to all the other nodes then

for each node

delete the weakest outgoing link (the outgoing arc with the smallest weight)

end-for

**II. Process the confusion graph as follows:**

A. Make a copy of the confusion graph

B. Repeat

Step 1: Select node with the lowest connectivity and call it $c_{i,\min}$ (break ties randomly)

Step 2: Create a set whose member are $c_{i,\min}$ plus all the nodes connected to the node $c_{i,\min}$

Step 3: Delete all the arcs connected to $c_{i,\min}$ and delete $c_{i,\min}$ from the graph

Until there are no arcs left in the graph

C. For each set of nodes created in step 2 of B above, if the set is a proper subset of another set, delete the set.

D. Assign each of the remaining sets as the positive classes for one model.

E. For each model, determine the negative classes. For each positive class in the model, add all classes that have an arc to the class node, but are not one of the positive classes for the model.

*Figure 8.6: Modified algorithm for class selection for the pVn base models*

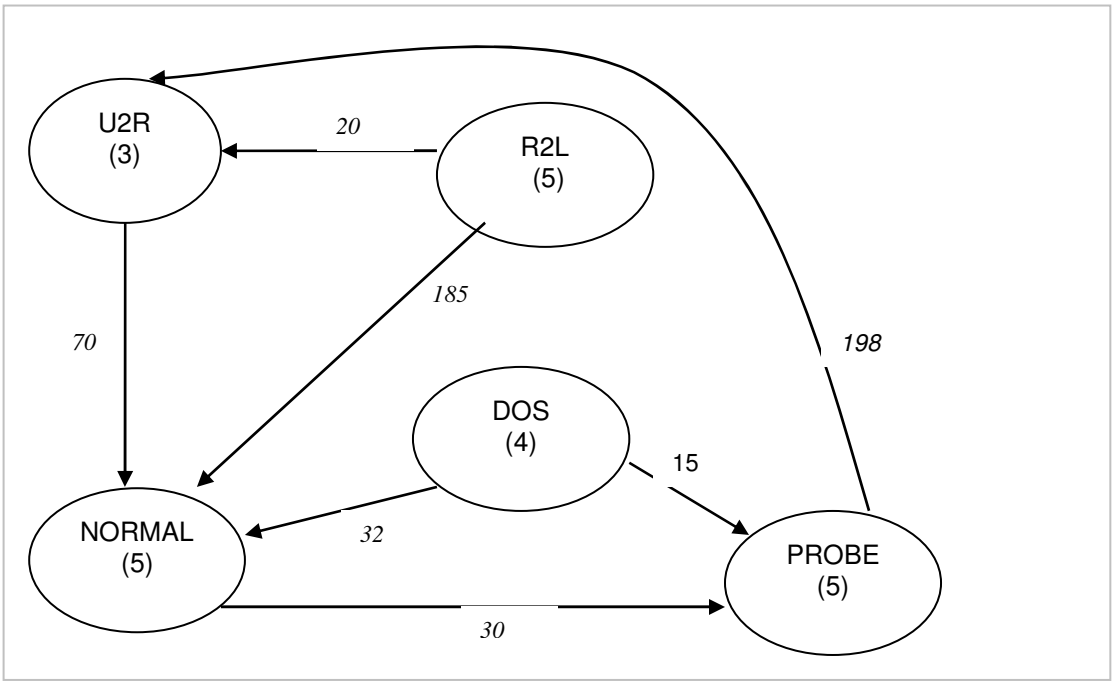


*Figure 8.7: Simplified confusion graph for the See5 single 5-class model for KDD Cup 1999*

*Table 8.8: Training set composition for theSee5 pVn base models*

| Dataset | Model ID | p (positive) classes | | n (negative classed) | | Training sample size |
|---|---|---|---|---|---|---|
| | | Classes used | sample percentage | classes used | sample percentage | |
| Forest cover type | M127 | C1,C2,C7 | 80: (27,27,26) | C3,C5,C6 | 20: (7,7,6) | 12000 |
| | M2356 | C2,C3,C5,C6 | 80: (20,20,20,20) | C1,C4 | 20: (10,10) | |
| | M346 | C2,C3,C6 | 80: (27,27,26) | C2,C5 | 20: (7,7,6) | |
| | | | | | | |
| KDD Cup 1999 | MNRU | NORMAL, R2L,U2R | 80: (27,27,26) | DOS, PROBE | 20: (10,10) | 1900 |
| | MNDP | NORMAL, DOS, PROBE | 80: (27,27,26) | R2L, U2R | 20: (10,10) | 4000 |
| | MNPU | NORMAL, PROBE, U2R | 80: (27,27,26) | DOS,R2L | 20: (10,10) | 1900 |

## 8.3.2 Predictive performance of the See5 pVn base models

The performance of the See5 pVn base models for the forest cover type and KDD Cup 1999 dataset are shown in table 8.9. Columns 3 and 4 of table 8.9 show the mean TPRATE and mean TNRATE values for the base models. The TPRATE in this context is the predictive accuracy on the p-classes while the TNRATE is the predictive accuracy on the n-classes.

*Table 8.9: Predictive performance of See5 pVn base models*

| Dataset (Training sample size) | Base model ID | Base model performance | | single model performance |
|---|---|---|---|---|
| | | Mean TPRATE% (p instances) | Mean TNRATE% (n instances) | Mean TPRATE% (p instances) |
| Forest cover type (12000) | M127 | 76.7 ± 1.5 | 89.9 ± 0.9 | 72.3 ± 1.4 |
| | M2356 | 76.8 ± 1.3 | 81.5 ± 2.0 | 72.2 ± 1.7 |
| | M346 | 82.3 ± 0.9 | 96.9 ± 0.6 | 78.5 ± 1.8 |
| | | | | |
| KDD Cup 1999 (4000) | MNRU | 77.4 ± 2.6 | 84.7 ± 3.2 | 67.0 ± 1.6 |
| | MNDP | 91.1 ± 1.9 | 63.9 ± 1.3 | 68.1 ± 1.7 |
| | MNPU | 74.8 ± 0.4 | 77.3 ± 1.4 | 66.5 ± 1.3 |

Column 5 of table 8.9 shows the mean TPRATE values for the single 7-class model on the p-classes for forest cover type, and the single 5-class model for KDD Cup 1999. The results in table 8.9 indicate that the pVn base models M127, M2356 and M346 for forest cover type each have higher TPRATE values on their p-classes compared to the single 7-class model on the same classes. The pVn models MNRU, MNDP and MNPU for the KDD Cup 1999 dataset also have significantly higher TPRATE values on their p-classes compared to the single 5-class model.

### 8.3.3 Predictive performance of the See5 pVn aggregate models

The pVn base models for the forest cover type and KDD Cup 1999 datasets were combined into aggregate models using the algorithm in figure 6.3. Table 8.10 shows the results of the predictive performance of the See5 pVn aggregate model for the forest cover type dataset. The detailed performance results are given in the appendix table F.8. Table 8.10 also gives the performance results for the single 7-class and OVA aggregate models of chapter 7. Table 8.11 shows the results of the statistical tests used to compare the performance of the single 7-class model and the pVn aggregate model.

*Table 8.10: Predictive performance of the See5 single, OVA and pVn models for forest cover type*

| Class name | See5 Mean accuracy / TPRATE% | | | |
|---|---|---|---|---|
| | Single model | un-boosted OVA aggregate model | boosted OVA aggregate model | pVn aggregate model |
| All classes | 76.9 ± 1.0 | 75.3 ± 0.7 | 79.4 ± 0.6 | 79.9 ± 1.0 |
| 1 | 57.4 ± 3.4 | 60.6 ± 2.6 | 65.0 ± 2.9 | 64.6 ± 2.9 |
| 2 | 63.8 ± 3.0 | 49.8 ± 3.6 | 69.8 ± 2.4 | 65.5 ± 4.2 |
| 3 | 60.8 ± 3.3 | 64.0 ± 1.8 | 63.2 ± 3.3 | 71.8 ± 3.3 |
| 4 | 96.8 ± 1.0 | 86.6 ± 1.7 | 95.4 ± 1.3 | 94.6 ± 1.7 |
| 5 | 86.2 ± 2.4 | 94.4 ± 1.8 | 88.4 ± 2.3 | 88.6 ± 1.8 |
| 6 | 77.8 ± 3.3 | 79.2 ± 2.0 | 76.0 ± 1.9 | 82.2 ± 2.6 |
| 7 | 95.6 ± 1.6 | 92.8 ± 2.5 | 97.8 ± 1.1 | 92.0 ± 2.8 |

The results of Student's paired sample t-test and the *Diff(A,S)* and *Ratio(A,S)* performance improvement measures provide the following evidence: The pVn aggregate model has a significantly higher level of performance compared to the single model. The aggregate model results in an increase of 3% in accuracy for all classes combined. For the TPRATE values of the individual classes, the aggregate model provides a significantly higher level of performance with an increase in the TPRATE of 11% on class 3 and 7.2% on class 1. The aggregate model provided a performance improvement of 4.4% in theTPRATE for class 6. However, there is no statistically significant improvement in the TPRATE values for the remaining four classes. In fact, the single model provided higher TPRATE values on two of these classes. The best See5 OVA aggregate model for the forest cover type dataset that was reported in chapter 7 provided a mean accuracy of 79.4 ± 0.6 as shown in table 8.10. The mean accuracy of the pVn aggregate model was 79.9 ± 1.0. This leads to the conclusion that both the OVA and pVn aggregate models can provide a comparable improvement in Predictive performance for the forest cover type dataset.

*Table 8.11: Statistical tests to compare the performance for See5 classification tree single and pVn aggregate models for forest cover type*

| Group mean accuracy / TPRATE% for 10 test sets | | Student's paired t-test (9 df) | | | Performance improvement measures | |
|---|---|---|---|---|---|---|
| Group A Aggregate model | Group S Single model | 95% CI of mean difference | P value (2 tail) | Group A better than Group S? | *Diff(A,S) %* | *Ratio(A,S)* |
| All classes-A (79.9 ± 1.0) | All classes-S (76.9 ± 1.0) | [1.9, 4.0] | 0.000 | yes | 3.0 | 0.1 |
| Class1-A (64.6 ± 2.9) | Class1-S (57.4 ± 3.4) | [2.9,11.5] | 0.004 | yes | 7.2 | 0.2 |
| Class2-A (65.2 ± 4.2) | Class2-S (63.8 ± 3.0) | [-1.7, 4.5] | 0.334 | no | 1.4 | 0.0 |
| Class3-A (71.8 ± 3.3) | Class3-S (60.8 ± 3.3) | [6.8, 15.2] | 0.000 | yes | 11.0 | 0.3 |
| Class4-A (94.6 ± 1.7) | Class4-S (96.8 ± 1.0) | [-4.6, 0.2] | 0.066 | no | -2.2 | -0.7 |
| Class5-A (88.6 ± 1.8) | Class5-S (86.2 ± 2.4) | [-1.4, 6.2] | 0.188 | no | 2.4 | 0.2 |
| Class6-A (82.2 ± 2.6) | Class6-S (77.8 ± 3.3) | [1.2, 7.6] | 0.014 | yes | 4.4 | 0.2 |
| Class7-A (92.0 ± 2.8) | Class7-S (95.6 ± 1.6) | [-5.5, -1.7] | 0.002 | no | -3.6 | -0.8 |

Table 8.12 shows the results of the Predictive performance of the pVn aggregate model for the KDD Cup 1999 dataset. The performance details are given in the appendix table F.16. The results for the single 5-class and OVA aggregate models are also shown in table 8.12. Table 8.13 shows the results of the statistical tests to compare the predictive performance of the single 5-class model and the pVn aggregate model. The results of Student's paired t-test clearly indicate that the aggregate model performance is much higher than that of the single 5-class model. The *Diff(A,S)* and *Ratio(A,S)* measures indicate that the increase in the TPRATE for three of the classes is between 12.1% and 60.6%. The TPRATE increase for the PROBE class is 60.6%, which is remarkably high. Overall, the accuracy increase over all the classes is 15.2%.

In comparison to the KDD Cup 1999 OVA aggregate model of chapter 7, the best See5 OVA aggregate model had a mean predictive accuracy of 61.7±0.9 as shown in table 8.12, while the pVn aggregate model has a mean predictive accuracy of 79.0±2.1. This comparison clearly indicates that the pVn aggregate model has a much higher level of predictive performance. Again, this provides evidence that pVn aggregate modeling can provide much higher performance gains compared to OVA modeling.

*Table 8.12: Predictive performance of See5 single, OVA and pVn aggregate models for KDD Cup 1999*

| Class name | See5 Mean accuracy / TPRATE (10 test sets of size 350) | | | |
|---|---|---|---|---|
| | Single model | un-boosted OVA aggregate model | boosted OVA aggregate model | pVn aggregate model |
| All classes | 63.8 ± 1.3 | 63.3 ± 1.2 | 61.7 ± 0.9 | 79.0 ± 2.1 |
| NORMAL | 86.0 ± 3.1 | 98.3 ± 0.7 | 99.2 ± 0.6 | 98.1 ± 0.6 |
| DOS | 82.0 ± 3.8 | 50.1 ± 4.4 | 56.3 ± 4.3 | 68.4 ± 6.5 |
| PROBE | 36.8 ± 2.4 | 88.0 ± 1.3 | 89.3 ± 1.4 | 97.0 ± 1.0 |
| R2L | 37.7 ± 3.3 | 34.3 ± 3.3 | 23.6 ± 3.4 | 54.1 ± 6.9 |
| U2R | 77.1 ± 0.0 | 45.7 ± 0.0 | 40.0 ± 0.0 | 77.1 ± 0.0 |

*Table 8.13 Statistical tests to compareSee5 single and pVn aggregate models for KDD Cup 1999*

| Group name and mean TPRAE% for 10 test samples | | Student's paired t-test (9 df) | | | Performance improvement measures | |
|---|---|---|---|---|---|---|
| Group A Aggregate model | Group S Single model | 95% CI of mean difference | p value (2 tail) | Group A better than Group S? | *Diff(A,S)%* | *Ratio(A,S)* |
| All classes-A (79.0 ± 2.1) | All classes-S (63.8 ± 1.3) | [12.8,17.5] | 0.000 | yes | 15.2 | 0.4 |
| NORMAL-A (98.1 ± 0.6) | NORMAL-S (86.0 ± 3.1) | [8.7,15.6] | 0.000 | yes | 12.1 | 0.9 |
| DOS-A (68.4 ± 6.5) | DOS-S (82.0 ± 3.8) | [-18.6,8.6] | 0.000 | no | -13.6 | -0.8 |
| PROBE-A (97.0 ± 1.0) | PROBE-S (36.4 ± 2.4) | [60.0,63.5] | 0.000 | yes | 60.6 | 0.97 |
| R2L-A (54.1 ± 6.9) | R2L-S 37.7 ± 3.3) | [9.5,23.2] | 0.000 | yes | 16.4 | 0.3 |
| U2R-A (77.1 ± 0.0) | U2R-S (77.1 ± 0.0) | no variance | no variance | same | 0.0 | 0.0 |

## 8.4 Comparison of performance variability for single and aggregate models

Given two systems or methods, the system or method with more predictable behaviour should be preferred (Cohen, 1995:pg 205). In the context of predictive modeling, the method with predictive performance which has lower variability should be preferred to one which exhibits erratic performance. A model with low performance variability has more predictable behaviour. The F-test for variances which was discussed in chapter 4, was used to test the null hypothesis that the variance of predictive accuracy for a single *k*-class model is the same as that for the OVA or pVn aggregate model. There are two available rules for the rejection of the null hypothesis for the 2-tail F-test. The first rule states that the null hypothesis should be rejected if the p-value for the test is less than the critical p-value. The second rule states that the null hypothesis should be rejected if the value of the F-statistic is greater or equal to the critical value of the F-statistic. The second rule was

used for the F-test inference given in table 8.14. The results of the F-tests indicate that, in general, there is no significant difference in performance variability between the single *k*-class models and OVA aggregate models, and between single *k*-class models and pVn aggregate models. This leads to the conclusion that both the single and aggregate models exhibit equal predictive coherence.

*Table 8.14: F- tests for comparison of performance variability for single and aggregate models*

| Dataset | Algorithm | Variance of predictive accuracy | | F-test for variance of accuracy on 10 test sets (9 x 9 df, $F_{critical} = 3.18$) | | |
|---|---|---|---|---|---|---|
| | | Single model (S) | Aggregate model (A) | F value = *Max{VarA ,VarS} / Min{VarA,VarS}* | p-value (F ≤ f) 1-tail | A has same coherence as S? |
| forest cover type | 5NN | single (2.9) | un-boosted OVA (2.3) | 1.26 | 0.37 | yes |
| | | | boosted OVA (0.9) | 3.13 | 0.05 | yes |
| | | | pVn (3.8) | 1.34 | 0.33 | yes |
| | See5 | single (2.5) | un-boosted OVA (1.1) | 2.17 | 0.13 | yes |
| | | | boosted OVA (0.8) | 2.95 | 0.06 | yes |
| | | | pVn (2.4) | 1.01 | 0.49 | yes |
| KDD Cup 1999 | 5NN | single (4.9) | un-boosted OVA (3.2) | 1.53 | 0.27 | yes |
| | | | boosted OVA (3.9) | 1.28 | 0.36 | yes |
| | | | pVn (3.4) | 1.47 | 0.29 | yes |
| | See5 | single (4.7) | un-boosted OVA (3.8) | 1.24 | 0.38 | yes |
| | | | boosted OVA (2.0) | 2.42 | 0.10 | yes |
| | | | pVn (11.9) | 2.52 | 0.09 | yes |

The following general conclusions can be made from the statistical tests of chapter 7 for the comparison means and the statistical tests of this chapter for the comparison of means and comparison of variances: Both OVA and pVn aggregate models provided a higher level of predictive performance compared to a single 7-class model for the forest cover type dataset. The single and aggregate models exhibited similar levels of predictive coherence, so that overall the aggregate models should be preferred to the single 7-class model. The pVn aggregate model provided a higher level of predictive performance compared to a single 5-class model for the KDD Cup 1999 dataset. The level of predictive coherence is similar for the single and pVn aggregate model, so that the aggregate model should be preferred.

It should be emphasized that the variance shown in table 8.14 is not the same as the variance component of the prediction error. Recall from section 2.8 that variance error is defined as variability in prediction of an instance ***x*** from one training sample to the next. For a given algorithm and modeling method, the measurement of variance error requires the creation of many models each based on a different training sample. The variance error is then estimated using the same test set for the different models (Kohavi & Wolpert, 1996).

## 8.5 Discussion

The benefits of pVn modeling are summarised in this section. The performance of OVA and pVn models is compared to the performance of single models. The limitations of the proposed methods for training dataset selection are discussed. Section 8.5.1 provides a summary of the benefits of pVn modeling. Section 8.5.2 presents a comparison of OVA and pVn modeling. Section 8.5.3 discusses the limitations of the proposed dataset selection methods.

### 8.5.1 Dataset selection for pVn modeling

pVn modeling was proposed as a method of problem decomposition with a potential to reduce the bias (errors in the model estimation process) and variance (sensitivity to the training sample) components of the prediction error. Secondly, the initial motivation for proposing pVn modeling was to reduce the number of base models as required for OVA modeling. The experimental results demonstrated that pVn modeling enables the creation of syntactically diverse and highly competent base models. The pVn models were designed based on the lessons learned from OVA modeling. Confusion graphs derived from confusion matrices were used as input to the proposed algorithm for determining the class composition for the pVn base models. The experimental results reported in this chapter have demonstrated that the design of the base models based on the proposed algorithms results in pVn base models that provide a high level of predictive performance when combined into an aggregate model. The pVn aggregate models provided a much higher level of predictive performance compared to a single *k*-class model for the two datasets and two algorithms used for the experiments.

## 8.5.2 Comparison of OVA and pVn modeling

Table 8.15 provides a summary of the predictive performance of the OVA and pVn models for the datasets and algorithms used in the experiments. One small dataset, namely Wine quality (white) (Cortez et al, 2009) was also used to test performance of OVA and pVn dataset selection and modeling. The experimental results for the forest cover type and KDD Cup 1999 datasets were discussed in detail in chapter 7 and in this chapter. The details of the test results for the wine quality dataset are provided in appendix tables F.17 through F.26.

*Table 8.15: Summary of performance improvements for OVA and pVn models*

| Dataset (size) | Algorithm | Is there a performance improvement compared to single model for the: | | |
|---|---|---|---|---|
| | | un-boosted OVA aggregate model? | boosted OVA aggregate model? | pVn aggregate model? |
| Forest cover type (large) | 5NN | yes | yes | yes |
| | See5 | no | yes | yes |
| KDD Cup 1999 (large) | 5NN | yes | no | yes |
| | See5 | no | no | yes |
| Wine quality - white (small) | 5NN | no | no | yes |
| | See5 | no | no | yes |

OVA modeling provided performance gains for the forest cover type dataset for both the 5NN and the See5 algorithms. The un-boosted version of OVA modeling provided a small performance improvement for KDD Cup 1999 for the 5NN algorithm. The boosted version of OVA modeling did not provide any performance gains for the KDD Cup 1999 dataset for the 5NN and See5 algorithms. OVA modelling did not provide any performance gains for the wine quality dataset. pVn modeling provided performance gains for the forest cover type, KDD Cup 1999 and wine quality datasets for both algorithms. The performance improvements for the pVn aggregate models were far more impressive for the KDD Cup 1999 dataset compared to the forest cover type and wine quality datasets. An examination of the confusion graphs of figures 8.1, 8.2, 8.4, 8.5 and 8.7 reveals that one main difference between the prediction tasks for forest cover type and KDD Cup 1999 is that there is one class (NORMAL) for the KDD Cup 1999 whose node is connected to all the other nodes (classes) in the graph. This is not the case for the forest cover type confusion graphs. This observation could help to explain why, for a dataset such as KDD Cup 1999, OVA modeling as proposed in chapter 7 does not provide significant performance

gains, while pVn modeling provides significant gains. Further studies are required before firm conclusions can be made.

The F-tests for variance indicated that, in general, both OVA and pVn aggregate models exhibit the same level of predictive coherence. This leads to the conclusion that the OVA or pVn aggregate model should be preferred if such a model provides a higher level of mean predictive performance compared to a single *k*-class model.

It was observed from the experiments on OVA and pVn modeling, that OVA and pVn modeling can be used to reduce the problems associated with creating predictive models from datasets with skewed class distributions, especially when one or more classes are severely under-represented in the dataset. This is the case, for example, for the U2R class in the KDD Cup 1999 dataset. For the 52 instance of the U2R class, a combination of bootstrap sampling, training sample design to include only the necessary classes in the OVA and pVn models, and reduction of the training sample size were implemented for the OVAU2R, MNPU and MNRU base models. This scheme resulted in performance improvements on the TPRATE for the U2R class for the OVA aggregate models using the 5NN algorithm. The U2R TPRATE for the single 5-class model was 31.6±0.3, for the un-boosted OVA aggregate model the TPRATE was 42.6±0.4, and for the boosted OVA aggregate model the TPRATE was 40.5±1.4. However, for the See5 algorithm, the OVA and pVn aggregate models did not provide an increase in the TPRATE for the U2R class.

### 8.5.3 Classification problems where proposed boosting methods are not appropriate

Two-class problems are very common in data mining especially in business applications (Giudici & Figini, 2009; Witten & Frank, 2005; Giudici, 2003; Berry & Linoff, 2000). It was stated in chapter 2 that the OVA and pVn base model design and dataset selection methods proposed in this thesis are not appropriate for 2-class problems, but rather to *k*-class problems where $k > 2$. However if each of the classes for a 2-class problem is located in more than one contiguous region of the instance space, then it should be possible to apply the proposed methods to that dataset. For example, suppose that a 2-class dataset has classes $c_1$ and $c_2$ with the instances of class $c_1$ located in regions $g_1$ and $g_2$ while the instances of class $c_2$ are located in

regions $g_3$ and $g_4$. Classes $c_1$ and $c_2$ can be re-labelled as $c_1g_1, c_1g_2, c_2g_3, c_2g_4$ so that the classification task becomes a 4-class prediction problem to which the proposed methods can be applied. Liu and Motoda (1998) have observed that cluster analysis is commonly used as a pre-processing step in data mining. Samoilenko and Osei-Bryson (2008), and Osei-Bryson (2010) have observed that clustering is commonly used as a step prior to predictive modeling for purposes of improving the performance of predictive models. The author of this thesis hypothesised that identification of 1-class contiguous regions in the instance space of a 2-class problem can be achieved through cluster analysis. Experiments to test this hypothesis are left for future work.

The datasets used for the empirical studies on boosting have the desirable property that their confusion matrices have off-diagonal entries $CM(c_i, c_j)$ with $i = 1,.., k,\ j = 1,.., k$ and $i \neq j$ which do not have an equal (or nearly equal) distribution of instances. In fact, some of the entries in the off-diagonal confusion matrix cells are zero. The proposed OVA and pVn base model design and training dataset selection for boosted OVA and pVn base models were based on this property. The training samples for each $OVA_i$ boosted base model or $pVn_i$ base model were designed as follows: Each training sample included only instances of the classes where the off-diagonal entries $CM(c_i, c_j)$ and $CM(c_j, c_i)$ for $i \neq j$ in the matrix cells have large values, and to exclude instances of the classes with small or zero counts. There are *k*-class datasets for which the above property does not hold as shown in tables 8.16 and 8.17.

*Table 8.16: See5 single 3-class model confusion matrix for abalone3C*

| Single model confusion matrix, training size = 3000, 10-fold cross validation | | | |
|---|---|---|---|
| Actual class | Predicted class | | |
| | young | middle | old |
| young | | 206 | 51 |
| middle | 183 | | 316 |
| old | 74 | 272 | |

For such datasets the (off-diagonal) entries in the class confusion cells all have nearly the same instance counts. The 3-class abalone3C dataset is a case in point. The 3-class waveform dataset (Blake & Merz, 1998; Breiman et al, 1984) was also identified as fitting this category. The confusion matrices for these two datasets for the See5 classification algorithm are given in tables 8.16 and 8.17.

*Table 8.17: See5 single 3-class model confusion matrix for waveform*

| Single model confusion matrix, training size = 5000, 10-fold cross validation | | | |
|---|---|---|---|
| Actual class | Predicted class | | |
| | Class 0 | Class 1 | Class 2 |
| Class 0 | | 269 | 217 |
| Class 1 | 160 | | 151 |
| Class 2 | 179 | 140 | |

The foregoing observations led the author to formulate the following property for *k*-class confusion matrices:

**Sparse confusion matrix property**:
A *k x k* confusion matrix with exactly one off-diagonal cell having a zero count is minimally sparse. A *k x k* confusion matrix with all *k(k-1)* off-diagonal cells having zero counts is maximally sparse. A *k x k* confusion matrix with *j* off-diagonal cells, $1 \leq j \leq k(k-1)$ having zero counts is a sparse confusion matrix.

The implication of the above property is that there are classes in the dataset that do not share a common region of class confusion. The two large datasets that were used in the OVA and pVn studies for boosting training datasets both have the sparse confusion matrix property for the single *k*-class models. For this reason, it was possible to design boosted training datasets for OVA and pVn base models which resulted in increased predictive performance. It should be noted that it is possible that a non-sparse confusion matrix has off-diagonal cells with counts that are much smaller than the counts of all the other off-diagonal cells. Such a matrix can be converted into a sparse confusion matrix by setting the off-diagonal cell counts with small values to zero.

## 8.6 Conclusions

The first question that was posed for the studies on aggregate modeling and training dataset selection was: *How should training datasets be designed in order to create base models that are syntactically diverse and highly expert at prediction for aggregate models?* The experimental results reported in this chapter have demonstrated that the design of pVn models based on the information in the confusion matrix and confusion graph for a single k-class model and the new pVn model design algorithm presented in this chapter, results in the design of pVn base

models that are syntactically diverse and highly expert at prediction. The discussion of section 8.5.3 has however made it clear that the pVn and boosted OVA base model designs that are proposed are only applicable to datasets for which the single k-class predictive model has a sparse confusion matrix.

The second question that was posed was: *How should training datasets for the base models be designed in order to achieve high accuracy for the aggregate model?* The experimental results reported in this chapter have demonstrated that when pVn base models are designed as described above, the aggregation of such base models results in increased predictive performance. This was shown to be the case for the datasets and the algorithms that were used in the experiments. The experimental results also demonstrated that the predictive performance increases achieved through the proposed OVA and pVn aggregate modeling methods do not come at the cost of reduced coherence in the predictions.

The models discussed in chapter 7 and this chapter were assessed for performance using mean values for accuracy and TPRATE values as well as the variance in accuracy. Evaluation of model performance using Receiver Operating Characteristic (ROC) analysis is presented in the next chapter.

# Chapter 9

# ROC Analysis for Single and Aggregate Models

Recall from section 4.7 that a discrete classifier simply assigns a class label to a test instance (Fawcett, 2001, 2004, 2006).The single, OVA aggregate and pVn aggregate models were treated as discrete classifiers for the predictive performance analysis reported in chapters 7 and 8. Even though the single, OVA aggregate and pVn aggregate models assign probabilistic scores to the test instances as discussed in section 6.4, the scores could not be used in the statistical tests used in chapters 7 and 8 to compare model performance. Student's paired samples t-test, the Diff(A,S) measure, and the Ratio(A,S) measures that were used to compare model performance do not provide the capability for the analysis of the probabilistic scores assigned to the model predictions.

Receiver Operating Characteristic (ROC) curves and ROC analysis were discussed in section 4.7. ROC analysis enables the analysis of classifiers based on the scores that are assigned to the test instances. The classification models of chapters 7 and 8 were treated as probabilistic classifiers for the ROC analysis reported in this chapter. The purpose of the ROC analysis was to answer the questions below in order to establish whether the aggregate models provide a better level of performance compared to the single models for different operating conditions:

*1. Do OVA aggregate models provide a higher level of predictive performance compared to single models for different operating conditions?*

*2. Do pVn aggregate models provide a higher level of predictive performance compared to single models for different operating conditions?*

This chapter is organised as follows: Sections 9.1 and 9.2 respectively provide a discussion of 2-class and multi-class ROC analysis. Section 9.3 provides a discussion of ROC analysis for the 5NN single and aggregate models. Section 9.4 provides a discussion of ROC analysis for the See5 single and aggregate models. Section 9.5 concludes the chapter.

## 9.1 ROC analysis for 2-class predictive models

Recall that ROC curves provided a graphic representation of predictive model performance for 2-class prediction tasks (Giudici & Figini, 2009; Witten & Frank, 2005; Giudici, 2003; Berry & Linoff, 2000). A probabilistic classification model typically assigns a class and a score for the class. Most commonly, the score is the probability that a test instance belongs to the predicted class (Giudici & Figini, 2009; Witten & Frank, 2005; Giudici, 2003; Berry & Linoff, 2000). ROC analysis is concerned with the selection of the model with the optimal performance based on the cut-off point (threshold) $\lambda$ that is used to decide when an instance should be declared positive or negative. A cut-off point (threshold) is the score value $conf(\mathbf{x})$ for which $conf(\mathbf{x}) \geq \lambda$ implies that the predicted class for instance $\mathbf{x}$ is the positive class. ROC analysis may also be used to determine which of two models provides a higher level of predictive performance as discussed in section 4.7. ROC analysis produces a statistic called the Area Under ROC curve (AUC). Recall from section 4.7.3 that when the predictive performance of a probabilistic classifier is better than random guessing then $AUC = AUC_{below} + AUC_{above}$. $AUC_{below}$ and $AUC_{above}$ are respectively the area below and the area above the 45 degree line which represents random guessing in the 2-dimensional ROC plane. The AUC is also the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance (Fawcett, 2006). Given two probabilistic classifiers, the classifier with the larger AUC value has a higher level of predictive performance under different operating conditions.

## 9.2 ROC analysis for multi-class predictive models

Computation of the AUC for 2-class classification models is a straight forward task. ROC analysis for *k*-class ($k > 2$) prediction tasks is concerned with the Volume Under the ROC Surface (VUS). Computation and visualisation of the VUS is a non-trivial task. Fawcett (2004, 2006) has discussed two approximations of the VUS measure that have been proposed by Hand and Till (2001) and Provost and Domingos (2001). The Hand and Till (2001) measure is defined as

$$AUC_{total} = \frac{2}{k.(k-1)} \sum_{i,j \le k, i \ne j}^{k} AUC(c_i, c_j) \tag{9.1}$$

where $AUC(c_i, c_j)$ is the AUC value computed for the two class ROC curve for the classes $c_i$ and $c_j$ and *k* is the number of classes for the predictive model.

The Provost and Domingos (2001: cited by Fawcett, 2004, 2006) measure is defined as

$$AUC_{total} = \sum_{i=1}^{k} P_r(c_i).AUC(c_i, rest) \tag{9.2}$$

where $AUC(c_i, rest)$ is the *AUC* for class $c_i$ compared to all the other *k*-1 classes and $P_r(c_i)$ is the prevalence (prior probability) of class $c_i$ in the training dataset(s). The Provost and Domingos (2001) measure is commonly called the *one-versus-rest* approximation of the VUS (Fawcett, 2001, 2004, 2006). The Provost and Domingos (2001) measure is easier to visualise and faster to compute. However, determining the prevalence (prior probability) $P_r(c_i)$ of a class is a simple matter for a single predictive model. When base models are based on boosted training datasets and then combined into one aggregate model, the determination of $P_r(c_i)$ is not straight forward any more. A modified version of the Provost and Domingos (2001) measure that was designed by the author of this thesis and used for the ROC analysis of this chapter is a simple mean value for the AUC and is defined as

$$AUC_{total} = \frac{1}{k} \sum_{i=1}^{k} AUC(c_i, rest) \tag{9.3}$$

where $AUC(c_i, rest)$ has the same meaning as before and *k* is the number of classes for the multi-class (*k*-class) prediction task. The justification for computing the mean value of $AUC(c_i, rest)$ in equation (9.3) is as follows: The VUS estimates of equations (9.1) and (9.2) are based on the arithmetic combination of the AUC values for many 2-dimensional planes in multi-class ROC space. Equation (9.1) computes a mean value for $k(k-1)/2$ such planes. Equation (9.2) computes a simple sum of weighted values of the AUC for *k* 2-dimensional planes. Given the foregoing

observations, computation of the mean AUC in equation (9.3) gives a useful estimate of the VUS, especially for purposes of comparing the performance of two multi-class probabilistic classifiers.

Several values need to be computed in order to derive the approximation of the VUS. The values that were computed and the methods for computation of these values are given in table 9.1. Since 10 test sets were used to measure model performance, it was necessary to combine the test results for the TPRATE and FPRATE into summary measures for the 10 test sets. The mean TPRATE and mean FPRATE were computed for each threshold value for the probabilistic classifier. Fawcett (2004, 2005) calls this approach *threshold averaging*.

*Table 9.1: Computations for the estimation of the VUS*

| Value | Description | Computation |
|---|---|---|
| Mean $TPRATE(c_i, rest, \lambda)$ | Mean TPRATE for probabilistic classifier $PC(c_i, rest)$ for threshold value $\lambda$ | Mean values computed using 10 test sets |
| Mean $FPRATE(c_i, rest, \lambda)$ | Mean FPRATE for probabilistic classifier $PC(c_i, rest)$ for threshold value $\lambda$ | Mean values computed using 10 test sets |
| $AUC(c_i, rest)$ | AUC computed for the curve defined by the mean TPRATE and FPRATE values for probabilistic classifier $PC(c_i, rest)$ for different $\lambda$ values. | Integration of the area between the curve and the $45^0$ line in the 2-dimensional ROC space. The $\lambda$ values for the 5NN probabilistic classifiers were: 0.6, 0.8 and 1.0. The values for See5 were: 0.5, 0.75 and 1.0. |
| Mean $AUC_{total}$ | Estimation of VUS | Computed using equation (9.3) |

## 9.3 ROC analysis for 5NN models

The ROC analysis results for the 5NN single and aggregate models for the forest cover type, KDD Cup 1999 and wine quality datasets are given in table 9.2. The details of the ROC analysis are given in the appendix tables G.2, G.3 and G.4. The $AUC_{above}$ values and *Gini* concentration coefficients for the probabilistic classifiers are given in table 9.2 columns 3 to 10 for each class. The mean $AUC_{above}$ and mean *Gini* values for the single *k*-class model and aggregate *k*-class models are also given in the table. Recall from sections 4.7.3 and 9.1 that $AUC_{above}$ is the area between the ROC curve and the 45 degree line and $Gini = 2\text{x}AUC_{above}$. When the 2-

dimensional ROC space is visualised as a grid of 100 cells with each cell having a width of 0.1 and a height of 0.1, then an increment of 0.01 in the AUC corresponds to an AUC increase of one such cell. This corresponds to a 2% increase in the area $AUC_{above}$ whose maximum value is 0.5, and an increase of 4% in the *Gini* concentration coefficient whose maximum value is 1.0.

*Table 9.2: ROC analysis results for the 5NN single and aggregate models*

| Dataset, algorithm | Probabilistic classifier *PC(c$_i$,rest)* | *AUC$_{above}$* and *Gini* concentration coefficient for model: | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | single | | un-boosted OVA | | boosted OVA | | pVn | |
| | | *AUC$_{above}$* | *Gini* | *AUC$_{above}$* | *Gini* | *AUC$_{above}$* | *Gini* | *AUC$_{above}$* | *Gini* |
| Forest cover type, 5NN | PC(1,rest) | 0.29 | 0.58 | 0.33 | 0.66 | 0.33 | 0.66 | 0.32 | 0.64 |
| | PC(2,rest) | 0.23 | 0.46 | 0.28 | 0.56 | 0.30 | 0.60 | 0.27 | 0.54 |
| | PC(3,rest) | 0.25 | 0.50 | 0.35 | 0.70 | 0.34 | 0.68 | 0.31 | 0.62 |
| | PC(4,rest) | 0.45 | 0.90 | 0.44 | 0.88 | 0.49 | 0.98 | 0.48 | 0.96 |
| | PC(5,rest) | 0.43 | 0.86 | 0.46 | 0.92 | 0.47 | 0.94 | 0.46 | 0.92 |
| | PC(6,rest) | 0.33 | 0.66 | 0.38 | 0.76 | 0.37 | 0.74 | 0.36 | 0.72 |
| | PC(7,rest) | 0.47 | 0.94 | 0.47 | 0.94 | 0.47 | 0.94 | 0.46 | 0.92 |
| | **Mean** | **0.35** | **0.70** | **0.39** | **0.78** | **0.40** | **0.80** | **0.38** | **0.76** |
| | | | | | | | | | |
| KDD Cup 1999, 5NN | PC(NORMAL,rest) | 0.36 | 0.72 | 0.41 | 0.82 | 0.41 | 0.82 | 0.43 | 0.86 |
| | PC(DOS,rest) | 0.33 | 0.66 | 0.33 | 0.66 | 0.33 | 0.66 | 0.48 | 0.96 |
| | PC(PROBE,rest) | 0.44 | 0.88 | 0.44 | 0.88 | 0.44 | 0.88 | 0.49 | 0.98 |
| | PC(R2L,rest) | 0.30 | 0.60 | 0.29 | 0.58 | 0.27 | 0.54 | 0.38 | 0.76 |
| | PC(U2R,rest) | 0.15 | 0.30 | 0.21 | 0.42 | 0.20 | 0.40 | 0.13 | 0.26 |
| | **Mean** | **0.32** | **0.64** | **0.33** | **0.66** | **0.33** | **0.66** | **0.38** | **0.76** |
| | | | | | | | | | |
| | PC(4,rest) | 0.04 | 0.08 | 0.04 | 0.08 | 0.04 | 0.08 | 0.03 | 0.06 |
| | PC(5,rest) | 0.15 | 0.30 | 0.17 | 0.34 | 0.18 | 0.36 | 0.16 | 0.32 |
| | PC(6,rest) | 0.03 | 0.06 | 0.04 | 0.08 | 0.06 | 0.12 | 0.12 | 0.24 |
| | PC(7,rest) | 0.09 | 0.18 | 0.11 | 0.22 | 0.12 | 0.24 | 0.10 | 0.20 |
| | PC(8,rest) | 0.04 | 0.08 | 0.04 | 0.08 | 0.04 | 0.08 | 0.05 | 0.10 |
| | **Mean** | **0.07** | **0.14** | **0.08** | **0.16** | **0.09** | **0.18** | **0.09** | **0.18** |

The mean $AUC_{above}$ values for the forest cover type models range between 0.35 and 0.40. The boosted OVA aggregate model provided the best performance (0.40), followed by the un-boosted OVA aggregate model (0.39) followed by the pVn model (0.38). Since the single model has a mean $AUC_{above}$ of 0.35, all forest cover type aggregate models provided an increased level of predictive performance over the single model. An examination of the performance on the individual classes reveals that the aggregate models provided increased performance levels on six out of the seven classes. There were no improvements on class 7.

The mean $AUC_{above}$ values for the KDD Cup 1999 models range between 0.32 and 0.38. The pVn aggregate model provided the best performance (0.38), followed by the un-boosted and boosted OVA aggregate models (0.33). Since the single model has a mean $AUC_{above}$ of 0.32, the OVA models provided a very slight improvement in predictive performance. The pVn model provided a much higher performance improvement over the single model. The $AUC_{above}$ values for the individual classes indicate that the KDD Cup 1999 pVn aggregate model provided increased performance levels on four out of the five classes. The un-boosted and boosted OVA aggregate models each provided increased performance levels on two out of five classes.

The mean $AUC_{above}$ values for the wine quality models are very small. The values range between 0.07 and 0.09. The boosted OVA and pVn aggregate models provided the best performance (0.09), followed by the un-boosted OVA aggregate models (0.08). Since the single model has a mean $AUC_{above}$ of 0.07, the OVA and pVn models provided a slight improvement in predictive performance. The $AUC_{above}$ values for the individual classes indicate that the wine quality pVn aggregate model provided increased performance levels on four out of the five classes. The un-boosted and boosted OVA aggregate models each provided increased performance levels on three out of five classes.

## 9.4 ROC analysis for See5 models

The ROC analysis results for the See5 single and aggregate models for the forest cover type, KDD Cup 1999, and wine quality datasets are given in table 9.3. The details of ROC analysis are given in the appendix tables G.5, G.6 and G.7. The $AUC_{above}$ values for the probabilistic classifiers are given in table 9.2 columns 3 to 10 for each class. The mean $AUC_{above}$ and mean *Gini* values for the single *k*-class model and aggregate *k*-class models are also given in the table.

The mean $AUC_{above}$ values for the See5 forest cover type models range between 0.36 and 0.38. The boosted OVA and pVn aggregate models provided the best performance (0.38), followed by the single model (0.37) followed by the un-boosted

OVA aggregate model (0.36). Since the single model has a mean $AUC_{above}$ of 0.37, the boosted OVA and pVn aggregate models for forest cover type provided an increased level of predictive performance over the single model. The un-boosted OVA aggregate model did not provide any performance gains. An examination of the performance on the individual classes reveals that the boosted OVA aggregate model provided increased performance on five out of the seven classes. The pVn aggregate model provided increased performance on six out of the seven classes.

*Table 9.3: ROC analysis results for the See5 single and aggregate models*

| Dataset, algorithm | Probabilistic classifier *PC(ci,rest)* | *AUCabove* and *Gini* concentration coefficient for model: | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | single | | un-boosted OVA | | boosted OVA | | pVn | |
| | | *AUCabove* | *Gini* | *AUCabove* | *Gini* | *AUCabove* | *Gini* | *AUCabove* | *Gini* |
| Forest cover type, See5 | PC(1,rest) | 0.27 | 0.54 | 0.28 | 0.56 | 0.30 | 0.60 | 0.31 | 0.62 |
| | PC(2,rest) | 0.29 | 0.58 | 0.22 | 0.44 | 0.30 | 0.60 | 0.30 | 0.60 |
| | PC(3,rest) | 0.29 | 0.58 | 0.30 | 0.60 | 0.30 | 0.60 | 0.34 | 0.68 |
| | PC(4,rest) | 0.46 | 0.92 | 0.43 | 0.86 | 0.47 | 0.94 | 0.47 | 0.94 |
| | PC(5,rest) | 0.42 | 0.84 | 0.45 | 0.90 | 0.42 | 0.84 | 0.43 | 0.86 |
| | PC(6,rest) | 0.37 | 0.74 | 0.36 | 0.72 | 0.36 | 0.72 | 0.39 | 0.78 |
| | PC(7,rest) | 0.47 | 0.94 | 0.45 | 0.90 | 0.48 | 0.96 | 0.45 | 0.90 |
| | **Mean** | **0.37** | **0.74** | **0.36** | **0.72** | **0.38** | **0.76** | **0.38** | **0.76** |
| | | | | | | | | | |
| KDD Cup 1999, See5 | PC(NORMAL,rest) | 0.38 | 0.76 | 0.44 | 0.88 | 0.41 | 0.82 | 0.40 | 0.80 |
| | PC(DOS,rest) | 0.40 | 0.80 | 0.25 | 0.50 | 0.27 | 0.54 | 0.34 | 0.68 |
| | PC(PROBE,rest) | 0.17 | 0.34 | 0.39 | 0.78 | 0.41 | 0.82 | 0.48 | 0.96 |
| | PC(R2L,rest) | 0.18 | 0.36 | 0.12 | 0.24 | 0.11 | 0.22 | 0.26 | 0.52 |
| | PC(U2R,rest) | 0.31 | 0.62 | 0.23 | 0.46 | 0.19 | 0.38 | 0.38 | 0.76 |
| | **Mean** | **0.29** | **0.58** | **0.29** | **0.58** | **0.28** | **0.56** | **0.37** | **0.74** |
| | | | | | | | | | |
| Wine quality white, See5 | PC(4,rest) | 0.11 | 0.22 | 0.16 | 0.32 | 0.16 | 0.32 | 0.14 | 0.28 |
| | PC(5,rest) | 0.18 | 0.36 | 0.17 | 0.34 | 0.18 | 0.36 | 0.19 | 0.38 |
| | PC(6,rest) | 0.05 | 0.10 | 0.01 | 0.02 | 0.01 | 0.02 | 0.11 | 0.22 |
| | PC(7,rest) | 0.14 | 0.28 | 0.09 | 0.18 | 0.10 | 0.20 | 0.16 | 0.32 |
| | PC(8,rest) | 0.04 | 0.08 | 0.05 | 0.10 | 0.06 | 0.12 | 0.06 | 0.12 |
| | **Mean** | **0.10** | **0.20** | **0.10** | **0.20** | **0.10** | **0.20** | **0.13** | **0.26** |

The mean $AUC_{above}$ values for the See5 KDD Cup 1999 models range between 0.29 and 0.37. The pVn aggregate models provided the best performance (0.37), followed by the single model and un-boosted OVA aggregate model (0.29) followed by the boosted OVA aggregate model (0.28). Since the single model has a mean $AUC_{above}$ of 0.29, the pVn aggregate models for KDD Cup 1999 provided an increased level of

predictive performance over the single model. The OVA aggregate models did not provide any performance gains. An examination of the performance on the individual classes reveals that the pVn aggregate model provided increased performance on four out of the five classes.

The mean $AUC_{above}$ values for the See5 wine quality models are very small. The single, un-boosted OVA, and boosted OVA models have values of 0.10 for the mean $AUC_{above}$. These results indicate that the OVA aggregate models did not provide any performance gains. The pVn aggregate model provided the best performance with a mean $AUC_{above}$ value of 0.13 which indicates an increased level of predictive performance over the single model. An examination of the performance on the individual classes reveals that the pVn aggregate model provided increased performance on all five classes.

## 9.5 Conclusions

The single and aggregate models of chapters 7 and 8 were treated as probabilistic classifiers for the ROC analysis discussed in this chapter. The first question that was posed for this chapter was: *Do OVA aggregate models provide a higher level of predictive performance compared to single models for different operating conditions?* Performance improvements were realised for the 5NN OVA aggregate models and the See5 boosted aggregate model for the forest cover type dataset. No performance gains were realised for the See5 un-boosted OVA aggregate model. No performance gains were realised from the OVA aggregate models for the 5NN and See5 algorithms for the KDD Cup 1999 and wine quality datasets.

The conclusion from the foregoing observations is that OVA aggregate modeling as proposed in this thesis may or may not result in improved performance. Schaffer (1994) has observed that no single strategy for machine learning is better at generalisation (prediction) than all other strategies for all problem domains. The above conclusion should therefore be viewed in the context of Schaffer's (1994) observation. The single model confusion matrices of the forest cover type 5NN and See5 models had higher levels of sparsity compared to the KDD Cup and wine quality single models. It can be concluded that OVA modeling, as proposed in this

thesis, provides performance improvements for a dataset whose confusion matrix has a high level of sparsity.

The second question that was posed for this chapter was: *Do pVn aggregate models provide a higher level of predictive performance compared to single models for different operating conditions?* The pVn aggregate models provided performance improvements for the forest cover type, KDD Cup1999, and wine quality datasets for both the 5NN and See5 algorithms. It can be concluded that pVn modeling provides performance improvements as long as the single model for a dataset has the sparse confusion matrix property.

In conclusion, the observations based on the ROC analysis of this chapter support the conclusions of chapter 7 and 8. The ROC analysis results have additionally demonstrated that OVA and pVn aggregate models can provide better predictive performance under different operating conditions compared to single models. Based on the conclusions of chapters 5, 7, 8 and this chapter, recommendations are given in the next chapter for dataset selection and aggregate modeling from large datasets.

# Chapter 10
# Recommendations for Dataset Selection

> '…the problems in science … on a deeper level ... are directed towards a consensus, or rational agreement, between the parties concerned …' (Toulmin et al, 1979)

The studies conducted on feature selection, training dataset selection, and aggregate modeling, the experimental results and analysis of the results were presented in chapters 5 to 9. This chapter provides an integrated discussion of the experimental results by giving a summary of the results. The chapter also provides theoretical models that were derived from the results and suggestions on how to conduct feature and training dataset selection for aggregate modeling from large datasets. Recall from section 4.3.5 that several researchers have argued for the need for empirically derived theories for computer systems (Simon, 1996), machine learning (Dietterich, 1997) and artificial intelligence systems (Cohen, 1995). It is the author's opinion that empirically derived theoretical models for data mining should provide value for researchers and practitioners in data mining. Recall that the main research question for the thesis was:

*What methods of dataset selection can be used to obtain as much information as possible from large datasets while at the same time using training datasets of small sizes to create predictive models that have a high level of predictive performance?*

The following sections provide several concise answers to this question. A summary of the methods that were used for the reduction of prediction error is given in section 10.1. Theoretical models and recommendations for feature selection and training dataset selection are provided in sections 10.2 and 10.3 respectively. Section 10.4 provides a summary of the chapter.

## 10.1 Reduction of prediction error

It was argued in chapter 2 that a high level of predictive performance should be achieved when training datasets are selected with the main objective of reducing prediction error. Chapter 2 provided a discussion of the components that make up

the predictive error, namely bias, variance and intrinsic error. The methods that have the potential to reduce the bias and variance error components were discussed in chapter 2. The use of simple models (Dietterich & Bakiri, 1995) and boosting (Freund & Schapire, 1997) are known to reduce the bias component of the prediction error. The use of aggregate modeling (Breiman, 1996) and the use of simple models (e.g. OVA base models) are known to reduce variance error. The use of good feature subsets for prediction (Dietterich & Kong (2005) and reduction of noise through sampling (Smyth, 2001) are known to reduce the variance error.

The main objective of the experiments reported in chapters 5, 7 and 8 was to reduce the bias and variance components of prediction error using the methods stated above. This was achieved through:
(1) The use of many (relatively) small samples for correlation measurement and base model construction.
(2) The design of simple base models, each of which specialises in the prediction of a subset of the $k$ classes ($k > 2$) for the prediction task and uses a different training dataset from the other base models.
(3) The design of training datasets for base models, with the objective of increasing the coverage of those regions of the instance space where correct prediction is more difficult.

## 10.2 Recommendations for feature selection

This section provides a summary of the discussion of the studies that were conducted for feature selection as reported in chapter 5. A theoretical model of the factors that affect the quality of selected features is proposed and guidelines are provided on how to proceed with feature selection in the presence of large datasets. Section 10.2.1 provides a summary of the feature selection studies. Section 10.2.2 provides guidelines for feature selection based on the reported experimental results.

### 10.2.1 Summary of the feature selection experimental results

The factors that affect the quality of selected features for single models were discussed in sections 5.6 and 5.7. In the context of this discussion, quality refers to the extent to which as many relevant features as possible are included, and as many

irrelevant and redundant features as possible are excluded from the selected subset of predictive features. The point was made in chapter 3 that existing literature in computational data mining indicates that most commonly a single sample of (all available) data is used to measure class-feature and feature-feature correlations for feature ranking. Probes (fake variables) have been used for the validation of class-feature correlations. The experiments of chapter 5 demonstrated that class-feature correlations measured from samples of a large dataset can vary widely from sample to sample. The point was also made in chapter 3 that in computational data mining, mathematical functions are commonly used as heuristic measures by feature subset search algorithms. The experimental results of chapter 5 revealed that the use of mathematical functions as heuristic measures does not always result in the best decisions for the features to be included in the subset of the best predictive features.

Based on the experimental results and conclusions of chapter 5, the following are research contributions of this thesis to the problem of feature selection:
(1) Reliable methods of measuring class-feature and feature-feature correlations through the use of many samples.
(2) Reliable feature ranking through the use of mean class-feature correlations values.
(3) Reliable class-feature and feature-feature correlation validation through the use of mean values for the class-probe correlations to eliminate non-relevant features.
(4) Usage of decision rules for heuristics evaluation of the best feature to select at a given decision point for a feature subset search algorithm.

Arising from the discussions of chapter 3 and the experimental results of chapter 5, the theoretical model shown in figure 10.1 was developed for purposes of representing the relationships between the factors that have an effect on the quality of selected features for predictive classification modeling. The theoretical model of figure 10.1 offers a predictive theory of the outcome of feature selection as depicted in figure 4.2 of section 4.3.3. However, the model of figure 10.1 does not provide causal explanations as depicted in figure 4.2. Proper causation experiments, with experiment controls are needed in order to conclude beyond reasonable doubt that the relationships shown in figure 10.1 are due to the indicated factors and not fully or partially due to other factors (Cohen, 1995: ch.9).

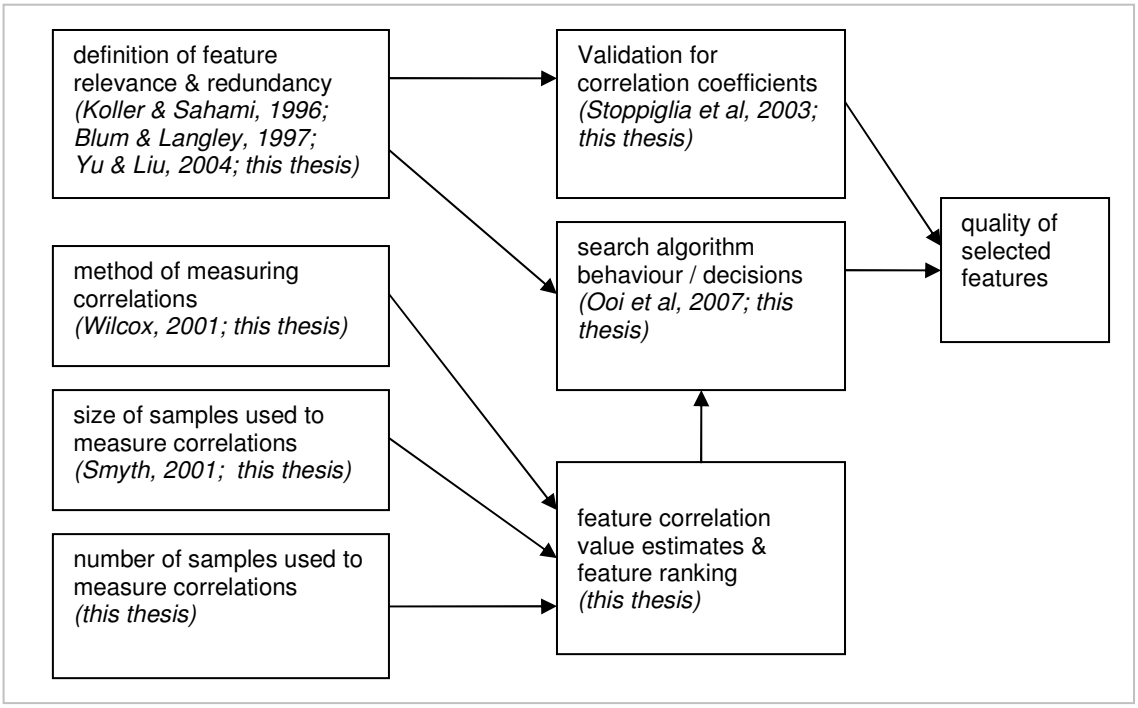


*Figure 10.1: Theoretical predictive model for feature selection using filtering methods*

The theoretical model of figure 10.1 predicts that the definitions of feature relevance and redundancy that are used in the procedures for the validation of the class-feature correlation coefficient estimate, will affect the outcome of the validation procedure. When feature ranking is all that is required, then the quality of the selected features will be affected by the outcome of the validation procedure. The definitions of feature relevance and redundancy will affect the behaviour of a feature subset search algorithm in terms of the final subset of selected features. The method used to measure the correlation coefficients, the sample sizes used, and the number of samples used, will all affect the estimate of the correlation coefficients, and in turn affect the feature ranking that is generated for input to the search algorithm. Finally, the quality of the feature subset selected by a search algorithm is influenced by the quality of the decisions made by the search algorithm.

### 10.2.2 Guidelines for feature selection

The steps shown in figure 10.2 are recommended for feature selection from large datasets of moderate dimensionality.

**Step 1:** Obtain information on the meaning of *low, medium* and *high* correlation for the domain from where the data originates. If this information is not available, use Cohen's (1988) guidelines.

**Step 2:** Take many small random samples and add one or more probes to each sample. Ten test samples of 1000 instances and at least one probe (Gaussian for quantitative continuous data, Uniform for quantitative discrete and qualitative data) provided useful information for the chapter 5 experiments.

**Step 3:** Measure the class-probe, class-feature and feature-feature correlations using a robust measure of association, eg. Kendall's tau or Pearson's r with the outliers removed.

**Step 4:** Compute the mean class-probe, class-feature and feature-feature correlations. If the confidence intervals of the means for the correlation values are large, go back to step 1 and increase the sample size.

**Step 5:** Conduct feature ranking based on the mean values of the class-probe and class-feature correlations.

**Step 6:** Use the probe method discussed in chapter 5 to eliminate all features whose ranking is below that of any of the probes from further consideration, as discussed in chapter 5.

**Step 7:** If the feature selection task is to select a pre-defined number of features (*w*), then select the top *w* features that have a correlation coefficient of practical significance for the problem domain and stop. Alternatively, a user-specified threshold for correlation values can be used to determine which features to select.

**Step 8:** If the feature selection task is to identify the best subset of features then construct decision rules for the meanings of relevance and redundancy for the problem domain where the dataset originates. If this information is not available, use Cohen's (1988) guidelines.

**Step 9**: Conduct the feature subset search using the decision rules of step 8 to obtain the best feature subset.

*Figure 10.2: Recommended procedure for feature selection from large datasets*

If the feature selection task is to select a pre-specified number of features, then steps 1 to 7 of figure 10.2 are recommended. If on the other hand, the task is to select the best subset of features, then steps 1 to 7 should be followed by steps 8 and 9. Step 9 involves the search for the best feature subset. Suggestions on how to conduct steps 1 to 7 using commonly available software (SPSS and MS Excel) are given in the appendix table H.1. The decision-rule base feature selection algorithm that was presented in chapter 5 is a good candidate for performing step 9. One alternative to

the above approach is to conduct steps 1 to 7 of figure 10.2 followed by Yu and Liu's (2004) method of redundancy analysis that was discussed in chapter 3.

## 10.3 Recommendations for training dataset selection for aggregate modeling

This section provides a summarised discussion of the studies that were conducted for OVA and pVn base model design and training dataset selection as well as the implications of the experimental results. A theoretical model that was developed for the factors that affect the quality of selected training datasets, based on existing literature is presented. An extension of the theoretical model based on the studies conducted for this thesis is proposed, and guidelines are provided on how to proceed with training dataset selection for aggregate model implementation in the presence of large datasets. Section 10.3.1 provides a summary of the training dataset selection experiments and the research contributions arising from the experiments. Section 10.3.2 presents the theoretical model for training dataset selection. Parallel and serial aggregation methods are discussed in section 10.3.3. Guidelines for training dataset selection are provided in section 10.3.4.

### 10.3.1 Summary of the training dataset selection experimental results

Sections 7.4 and 7.5 provided the discussion and conclusions for the OVA model dataset selection experiments. Sections 8.5 and 8.6 provided the discussion and conclusions for the pVn model dataset selection experiments. Chapter 9 provided the results for ROC analysis to compare single models, OVA and pVn aggregate models. The main conclusions from chapters 7, 8 and 9 were that the proposed dataset selection methods for OVA and pVn aggregate modeling generally provided improvements in predictive performance. In summary, the main research contributions arising from the reported experiments are as follows:

(1) The use of OVA modeling to increase the amount of training data used for modeling from large datasets, and to increase the level of predictive performance.

(2) The use of pVn modeling to increase the amount of training data used for modeling from large datasets, and to increase the level of predictive performance. pVn modeling reduces the number of base models compared to OVA modeling.

(3) The use of a confusion matrix to provide information for the design of boosted OVA and pVn base models.
(4) The use of a confusion graph as a graphical and mathematical representation of the information in a confusion matrix to be used as input to the algorithm for determining the positive and negative classes of pVn base models.
(5) A definition of the sparse confusion matrix property which can be used to determine whether boosted OVA and pVn base models will provide performance improvements for a given dataset.
(6) A base model combination algorithm for KNN OVA and pVn base model predictions. The algorithm resolves tied predictions.

### 10.3.2 Theoretical model for training dataset selection

A theoretical model to summarise the work on aggregate modeling as reported in chapter 2 was developed by the author. The theoretical model is shown in figure 10.3. One major factor that affects the performance of aggregate models is syntactic diversity. Recall from chapter 2 that the term syntactic diversity refers to the level of dis-similarity between the base models that make up an aggregate model. Syntactic diversity has been achieved by researchers (as indicated in figure 10.3) either through variation of the learning task, or variation of the base model structure, or variation of the training datasets for base models. A second major factor that affects aggregate model performance is the predictive accuracy of the base models. Several researchers (as indicated in figure 10.3) have achieved a high level of base model predictive accuracy (Chan & Stolfo, 1998) or single model accuracy (Kubat & Matwin, 1997) through sampling methods that balance the level of class representation for datasets with skewed class distributions. A second approach has been to vary the learning task and/or the base model structure.

Syntactic diversity, predictive accuracy of the base models and the method of determining the winning class lead to a reduction of the bias and variance components of the prediction error of an aggregate model as depicted in figure 10.3. The level to which the bias and variance components of the prediction error are reduced affects the predictive performance of the aggregate model.

The research for this thesis concentrated on the selection of training datasets from large amounts of data, with the objective of constructing aggregate models which

provide a high level of predictive performance. The methods of training dataset selection that were studied were aimed at achieving variation in the base model structures, variation in the training datasets for the base models, and balancing of the class representation in the base models. The theoretical model shown in figure 10.4 is an extension of the model of figure 10.3, based on the studies conducted for this thesis.

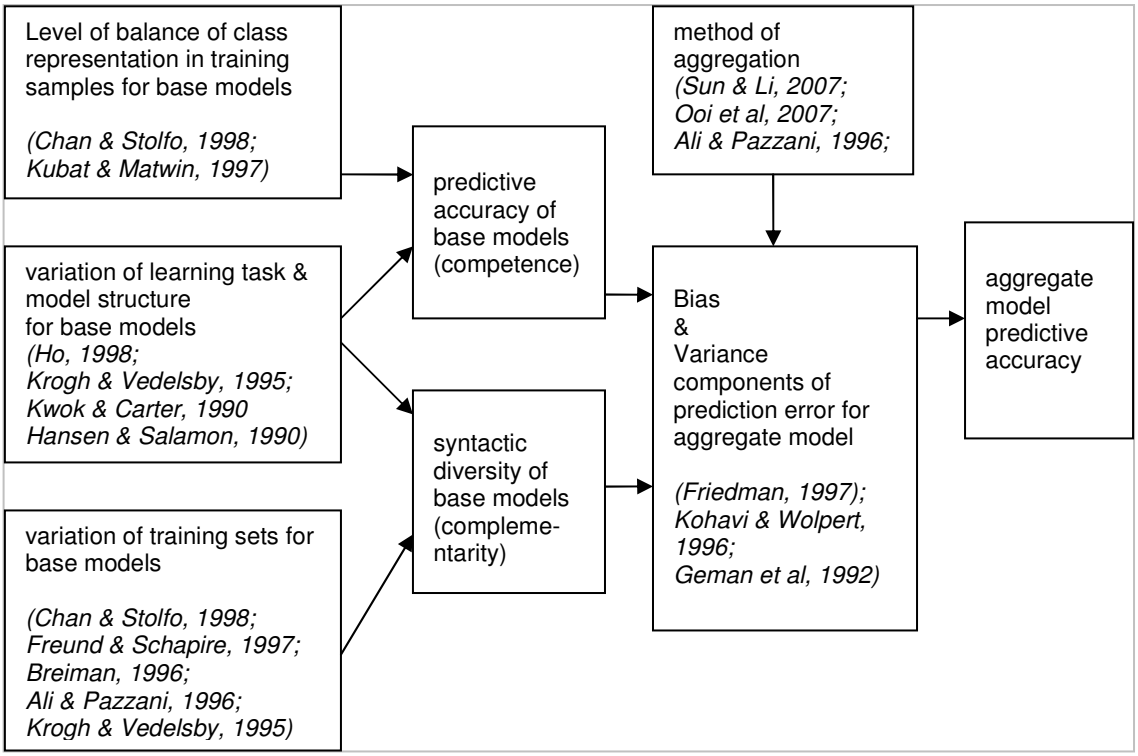


*Figure 10.3: Theoretical predictive model for aggregate model performance based on existing literature*

The model of figure 10.4 predicts that the use of information about the structure of the instance space combined with information on the aggregation method should result in the design of a set of base models whose performance should ultimately result in high predictive performance. The design of the base models should influence the methods used to select the training sample for each base model from the large dataset. The methods of training dataset selection, based on the designed base models, should influence the level of balance of the classes in the training datasets, the level of variation in the training datasets for the base models, and the level of variation in the learning tasks and structures of the base models. This should in turn influence the predictive performance of each base model, and the syntactic diversity in the set of base models. The algorithm used for combining the base model predictions will affect the bias error of the aggregate model.

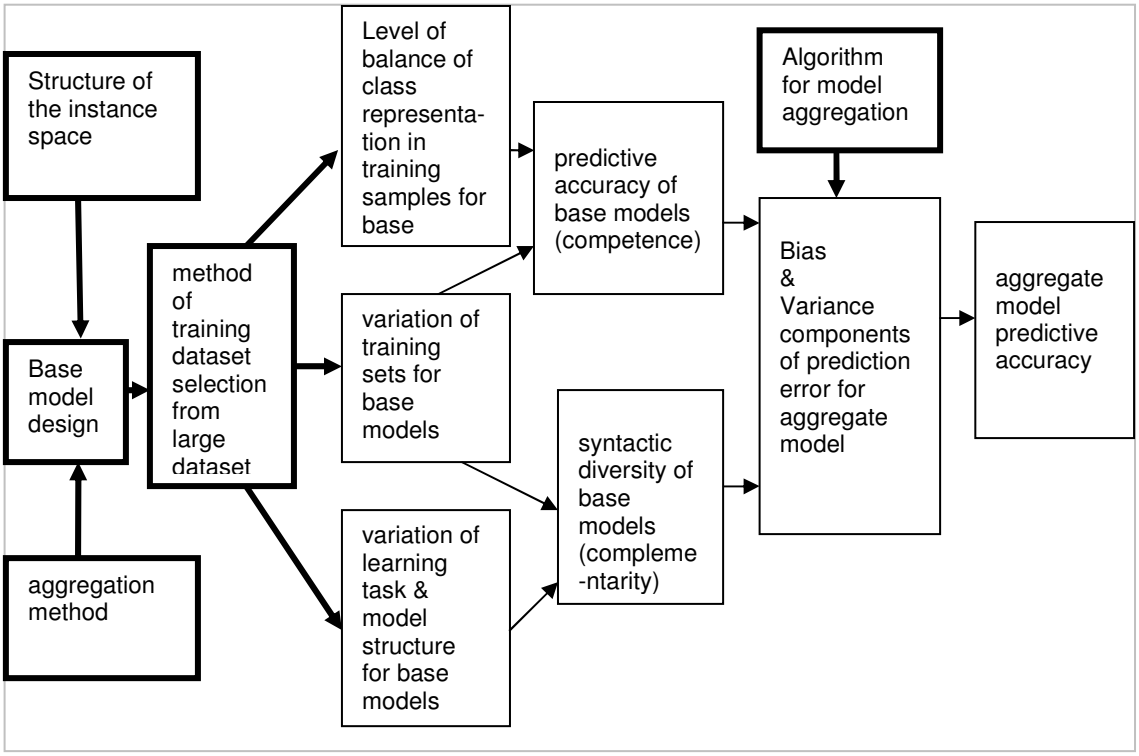


*Figure 10.4: Extensions to the theoretical predictive model for aggregate model performance based on studies for this thesis*

## 10.3.3 Parallel versus serial aggregation of base models

The method of parallel combination of base models was used to create the aggregate models for the experiments reported in this thesis. Serial combination (Sun & Li, 2008; Neagu et al, 2006; Kim et al, 2002) is the second method of base model aggregation which was discussed in chapter 2. Recall that s*erial combination* is a multi-step process. In the first step the base models are arranged in a series. In order to classify a new instance, the instance is passed to the first base model in the series. If the base model makes a *credible prediction,* then the process stops, otherwise the instance is passed to the next base model in the series. In general, if a base model makes a *credible prediction* the process stops otherwise the instance is passed to the next base model in the series (Sun & Li, 2008). The base model which provides the highest predictive accuracy on a given class is considered to be the base model that makes a *credible prediction* for that class (Sun & Li, 2008).

Sun and Li (2008) have conducted studies on the serial combination of base models, where each base model can make predictions for any of the classes and is constructed with a different classification algorithm. Two useful aspects of the base models were noted in the design and testing process for the pVn base models. First

of all, in general, a pVn base model was found to provide a higher level of accuracy on the positive classes that it predicts, compared to the single *k*-class model. Secondly, several of the classes for the prediction task can be predicted by more than one base model. Based on the foregoing observations the author hypothesized that the use of pVn base models in a serial combination scheme would provide performance improvements, especially for decision tree algorithms where no measures are available for resolving tied predictions. Studies to confirm this hypothesis were left for future work.

### 10.3.4 Guidelines for OVA and pVn model design, training dataset selection and testing

The steps given in figures 10.5 to 10.8 are recommended for the design of OVA and pVn aggregate models, training dataset design and selection for the models, and aggregate model creation and testing.

**Phase I: Steps to establish class confusion**

1. Partition the large dataset according to class, so that *k* partitions are created, one for each class.
2. From each partition obtained in step 1, set aside the data for model testing.
3. Decide on the sample size, *n*, for the creation of a single *k*-class model.
4. To obtain the training dataset for the single class model, proceed as follows. For each class $c_i$ in the data, obtain a random sample of size $n/k$ from the corresponding partition. If the partition has a size less than $n/k$, use bootstrap sampling to obtain the required sample size.
5. Combine all the samples obtained in step 4 to create the training dataset for the single *k*-class model. This training dataset will have an equal class distribution.
6. Create several test sets with an equal class distribution from the test data partitions.
7. Create the single *k*-class model and test it with the test sets created in step 6 in order to generate a confusion matrix for the classes.
8. Compute the predictive accuracy, TPRATE and TNRATE for each class in the single *k*-class model on the test sets.
9. If the confusion matrix is sparse, create a confusion graph from the confusion matrix.

*Figure 10.5: Steps for the creation of a confusion matrix and confusion graph*

The steps are based on the observations from the experiments of chapters 7 and 8. The first phase involves the establishment of the class confusion in a single *k*-class

model. Figure 10.5 shows the recommended steps to be followed for the identification of the class confusion.

**Phase IIa: Un-boosted OVA model design, training dataset selection and testing**

To create an un-boosted OVA aggregate model, proceed as follows:

1. Design the class and training sample composition for each $OVA_i$ model so that class $c_i$ has 50% of the instances and all the other classes combined have 50% of the instances.
2. Obtain the training samples for the OVA base models based on the design of step 1 by sampling from the partitions created in phase I. Use bootstrap sampling if the partition size is smaller than the required number of instances.
3. Create the OVA base models and OVA aggregate model, and test the aggregate model using the test samples created in phase I.
4. Compare the performance of the un-boosted OVA aggregate model with that of the single *k*-class model on the test samples.

*Figure 10.6: Steps for the design, creation and testing of un-boosted OVA aggregate models*

**Phase IIb: Boosted OVA model design, training dataset selection and testing**

If the single *k*-class model has a sparse confusion matrix, proceed as follows to create a boosted OVA aggregate model:

1. For each class $c_i$, determine from the confusion matrix or confusion graph which other classes are predominantly confused with $c_i$.
2. Design the class and training sample composition for each $OVA_i$ model so that class $c_i$ has 50% of the instances and the classes identified in step 1 have 50% of the instances. Consider apportioning the class representation based on the level of confusion, as discussed in chapter 7.
3. Obtain the training samples for the OVA base models based on the design of step 2 by sampling from the partitions created in phase I. Use bootstrap sampling it the partition size is smaller than the required number of instances.
4. Create the OVA base models and OVA aggregate model, and test the aggregate model using the test samples created in phase I.
5. Compare the performance of the OVA aggregate model with that of the single k-class model on the test samples.

*Figure 10.7: Steps for the design, creation and testing of boosted OVA aggregate models*

The steps given in figures 10.6 and 10.7 are recommended for purposes of creating OVA aggregate models. These steps should be conducted after the steps given in figure 10.5. For purposes of creating pVn aggregate models, the steps given in figure 10.8 are recommended. These steps should be conducted after the steps given in figure 10.5. Suggestions on how to conduct the steps of figures 10.6, 10.7 and 10.8 using commonly available software (SPSS and MS Excel) are given in the appendix table H.2.

**Phase III: pVn model design, training dataset selection, and testing**

If the single *k*-class model has a sparse confusion matrix, proceed as follows to create a pVn model:

1. Use the algorithms of figures 8.4 and 8.5 to establish the p-classes and n-classes for the base models, based on the confusion graphs created in phase I.
2. Design the training samples so that the p-classes combined have a high instance representation (eg. 80%) and the n-classes combined have a low instance representation (eg. 20%).
3. Obtain the training datasets designed in step 2 through random sampling from the partitions created in phase I.
4. Create the pVn base models and aggregate model and test the performance of the aggregate model using the test sets created in phase I.

*Figure 10.8: Steps for the design, creation and testing of pVn aggregate models*

## 10.5 Chapter summary

A summary of the research contributions for feature selection, base model design and training dataset selection have been given in this chapter. Recommendations for feature selection and training dataset selection for OVA and pVn modeling from large datasets have also been presented. A detailed discussion of the research contributions in terms of the expectations for design science research is provided in the next chapter.

# Chapter 11
# Discussion of Research Contributions

> 'In theory every individual scientist is capable of being his/her most severe critic, and his/her own writings are expected to discuss with real care and seriousness the objections against his/her own novel ideas..' (Toulmin et al, 1979)

The research contributions for feature selection, base model design and dataset selection for aggregate modeling were summarised in sections 10.2.1 and 10.3.1. It was stated in chapter 4 that the design science research paradigm was used to guide the activities of the research, and the design science research process was discussed in detail in that chapter. A brief discussion of the expected design science research outputs is provided in this chapter followed by the author's self-assessment of how the research meets the expectations of design science research. Sections 11.1 and 11.2 respectively provide a discussion of the outputs of design science research and the recommendations for design science research evaluation. Section 11.3 provides a discussion of the limitations of the methods proposed in this thesis. Section 11.4 provides a summary of this chapter.

## 11.1 Outputs of design science research

Hevner et al (2004) have stated that design science research for Information Systems must produce one or more artifacts. Recall from chapter 4 that Hevner et al (2004) have defined an artifact as:

> '..innovations that define ideas, practices, technical capabilities, and products, through which the analysis, design, implementation, and use of Information Systems can be effectively accomplished.'

Hevner et al (2004) and March and Smith (1995) have further stated that the artifacts for design science research are *constructs, models, methods*, and *instantiations*. Vaishnavi and Kuechler (2004/5) have observed that in addition to the production of artifacts, design science research should produce *better theories* for the field of research. *Constructs* form the conceptual vocabulary of the field of study. *Constructs*

make up the language used to define and communicate the problems and solutions in the field of study. For design science research, the term '*model*' is used to refer to the set of propositions that specify relationships between the *constructs*. *Methods* are definitions of the processes that need to be achieved. A method may be stated as a set of steps to perform a given task, or a method may be specified as a formal computational algorithm. *Instantiations* are the actual implementations of the models and methods in order to demonstrate that they actually work. '*Better theories'* provide an increased understanding arising from the study of the created artifacts.

## 11.2 Evaluation of design science research

The criteria provided by Hevner et al (2004) for the evaluation of design science research are discussed in this chapter together with the author's self assessment of how the these criteria were met. The criteria for design science research evaluation are presented in section 11.2.1. Sections 11.2.2 through 11.2.6 provide a discussion of the author's self-assessment based on Hevner et al's (2004) assessment criteria.

### 11.2.1 Criteria for design science research evaluation

Manson (2006) has argued that criteria for the evaluation of research help researchers, reviewers, editors, and readers to understand the requirements for effective research. Hevner et al (2004) have provided seven guidelines for evaluating design science research as shown in table 11.1. Even though Hevner et al (2004) have advised against mandatory use of these guidelines, the author is of the opinion that in the absence of alternative guidelines at her disposal, these guidelines are suitable for stating the research contributions and conducting a self-assessment of the work done. The extent to which requirement number 2 (problem relevance) was met, was discussed in chapters 1, 2 and 3 of this thesis. The research that was conducted is assessed in the following sections.

*Table 11.1: Criteria for the evaluation of design science research: adopted from Hevner et al (2004/5)*

| **Criterion / Requirement** | **Description** |
|---|---|
| 1. Design of an artifact | Design science research for Information Systems must produce a useful artifact in the form of a construct, model, method or an instantiation |
| 2. Problem relevance | The problem that the design science research is aimed at solving, must be technology-based, important, and relevant to some business function. |
| 3. Design evaluation | The utility, quality, and effectiveness of the designed artifact must be rigorously demonstrated using well executed methods of evaluation. |
| 4. Research contributions | Design science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies |
| 5. Research rigor | The researcher must demonstrate that rigorous methods were applied in both the construction and evaluation of the designed artifact. |
| 6. Design as a search process | The researcher must demonstrate that the available means were utilized well, in order to reach the desired ends while satisfying the laws in the problem environment (Satisficing). |
| 7. Communication of the research | Design science research must be presented effectively to the intended audience. |

### 11.2.2 Constructs, models and better theories

Requirement number 1 in table 11.1 refers to the design of artifacts. Constructs and models are two of the artifacts that design science research must produce (Hevner et al, 2004). The author claims in this thesis that one construct that arose from this research is the concept of decision rule-based search for feature subset selection. The author further claims that two constructs that arose out of this research are: pVn modeling, confusion graphs and the associated sparse confusion matrix property for aggregate modeling. Theoretical models are propositions expressing the relationships between the constructs / concepts of the research domain. Theoretical models were developed to express the relationships between the factors that affect the quality of selected features, and the factors that have an influence on the outcome of training dataset selection for aggregate modeling.

With reference to *better theories*, the experimental results were used to demonstrate the relationships between the various factors that affect predictive model performance. It will be necessary in future to conduct causation experiments (Cohen, 1995) to provide proof of these relationships.

### 11.2.3 Methods and instantiations

Methods and instantiations are two of the artifacts that design science research must produce (Hevner et al, 2004). A summary of the research contributions and proposed methods for feature selection, base model design, training dataset selection, and model aggregation for large datasets was given in sections 10.2 and 10.3. In this thesis, the author claims that the proposed methods and algorithms result in the selection of good feature ranking, good feature subset selection, design of highly competent base models, and selection of good training datasets for the base models. Furthermore, the proposed algorithm for model combination of 5NN base models (and KNN models in general) result in more effective resolution of conflicting predictions by the base models. Instantiation refers to the creation (implementation) of artifacts (constructs, models, methods) and demonstration that the artifacts can be implemented in a working system. All the methods and algorithms proposed in this thesis were implemented, tested and found to provide statistically significant improvements in predictive performance.

### 11.2.4 Rigorous design evaluation

Requirements number 3 and number 5 in table 11.1 are concerned with design evaluation, and the rigor that is applied to the evaluation processes. Hevner et al (2004) have stated that design evaluation involves the demonstration of utility, quality and effectiveness. Furthermore, the evaluation methods used to demonstrate (claim) the utility, quality and effectiveness of the methods and instantiations should also be evaluated. Hevner et al (2004) have further stated that designed artifacts should be evaluated using the methodologies that exist in the knowledge base for the field of research.

The evaluation methods that are available for predictive data mining originate from the area of Statistics, Machine Learning and Operations Research and were discussed in section 4.7. These evaluation methods enable the researcher to:

(1) Measure the predictive performance of a model in terms of overall predictive accuracy and error rate on all the classes, true positive, false positive, true negative, and false negative rates on each individual class.

(2) Measure the performance gains of using an aggregate model compared to using a single model.

(3) Conduct statistical tests, most commonly the Student's t-tests on means and F-tests on variance, to compare the predictive accuracy of two models.

(4) Conduct in-depth model analysis using ROC curves.

All the above methods were used for this thesis for the assessment of model performance. Machine learning research has traditionally concentrated on small datasets as exemplified by the datasets available from the UCI Machine Learning repository (Ascuncion & Newman, 2007; Blake & Merz, 1998). These datasets range in size from 100 instances to 10000 instances, and typically have a small number of predictive features. Researchers in machine learning have routinely used many small datasets (e.g. 30) to evaluate algorithm performance. However, as discussed in chapter 4, for experimental studies on aggregate modeling, bias and variance reduction, researchers have typically used small numbers of datasets ranging between two and nine datasets. The exception has been Ali and Pazzani (1996) who have used 30 small datasets. Performance evaluation using 30 small datasets can be feasibly conducted using a modest amount of time and computational resources.

Data mining poses new challenges in terms of evaluation. Typically, very large datasets are used as exemplified by the datasets available from the UCI KDD archive (Bay et al, 2000; Hettich & Bay, 1999). Datasets for data mining research range in size from 0.1 million instances to several million instances. Additionally these datasets have large numbers of potentially predictive features. In the author's opinion, the demonstration of rigor in evaluation, through the use of many very large datasets requires an excessively large amount of time and computational resources, which are not available to many researchers. In chapter 4 it was observed that experimental studies in dataset selection and aggregate modeling have been conducted by teams of researchers using between one and four very large datasets.

The author used a small number of datasets. Two small datasets (Abalone and Mushroom) and two large datasets (forest cover type and KDD Cup 1999) were used for feature selection. Two large datasets (forest cover type and KDD Cup 1999) and one small dataset (wine quality) were used for the training dataset selection and aggregate modeling studies. Twenty four models were created and tested for the three datasets, two algorithms, and four modeling methods. Many samples were taken from the large datasets, and the sample sizes used were larger than the typical dataset size for machine learning. Experiments were designed through the application of the scientific method and the evaluation methods listed above were employed.

### 11.2.5 Rigor and design as a search process

Requirements number 5 and number 6 in table 11.1 are concerned with the search process followed to arrive at good solutions for artifact design, and the rigor that is applied to the search process. Hevner et al (2004) have stated that rigor in the design process for design science research is derived from the effective use of the existing knowledge base (theoretical foundations and methodologies) of the field of research. A detailed assessment of the theoretical foundations of existing methods of dataset and feature selection was provided in chapters 2 and 3. A discussion was provided on how several existing theories can be applied to the task of designing feature selection and training dataset selection from large datasets for aggregate model implementation. The experiments presented in chapters 5, 7 and 8 were designed based on the assessments given in chapters 2 and 3 and the methodologies presented in chapter 4.

Hevner et al (2004) and Simon (1996) have observed that the design of artifacts is a search process aimed at the discovery of an effective solution to a problem. Hevner et al (2004) and Simon (1996) have characterized the design process as a generate-and-test cycle involving the generation of design alternatives and testing the alternatives against specific requirements. To the author's understanding, the generate-and-test cycle discussed by Hevner et al (2004) and Simon (1996) is identical to the scientific method that was discussed in chapter 4, and depicted in figure 4.3. Hevner et al (2004) have observed that an un-guided search for design alternatives would be intractable. It is usually prudent to employ heuristic strategies in order to generate designs for satisfactory solutions. In the field of Operations

Research, this approach is called *satisficing* (Simon, 1996). Heuristic search and *satisficing* for the scientific method are achieved through the cycle of: (*experiment-design)→(empirical-testing)→(empirical-observation)→(hypothesis-generation)→(experiment-design)*, as depicted in figure 4.3. The scientific method was followed for the studies reported in chapters 5, 7 and 8.

### 11.2.6 Research contributions for design science research

Requirement number 4 in table 11.1 is concerned with research contributions. Hevner et al (2004) have observed that any assessment of a research activity must answer the question: *'What are the new and interesting contributions?'* Hevner et al (2004) have stated that design science research must provide one or more of the following contributions: *design artifact, foundations* and *methodologies*. For this thesis, the author claims that the design artifacts that were discussed in sections 11.2.2 and 11.2.3 are research contributions to the field of predictive data mining. *Foundations* refer to the knowledge base of the field. The author further claims that the algorithms presented in chapters 5, 6 and 8 for feature selection and aggregate modelling are contributions to the field of predictive data mining.

Table 11.2 provides a summary of the new algorithms proposed in this thesis. The guidelines for feature selection and training dataset selection, new modeling methods, and theoretical models discussed in chapter 10 are a research contribution to the field of predictive data mining.

*Table 11.2: Summary of new algorithms*

| Algorithm category | Location | Description |
|---|---|---|
| Feature selection | Fig. 5.3 | Decision rule-base search algorithm for heuristic search of the best feature subset |
| OVA modeling | Fig. 6.3 | Algorithm for combining base model predictions for the See5 algorithm and for classification trees in general |
| pVn modeling | Fig. 6.4 | Algorithm for combining base model predictions for the 5NN algorithm and for the KNN algorithm in general |
| | Fig. 8.3 | Algorithm for class selection of pVn base model |
| | Fig. 8.6 | Modified algorithm for class selection for pVn base models |

*Methodologies* refer to the creative development and use of new evaluation methods and evaluation metrics. A modified version of Ali and Pazzani's (1996) performance improvement measures were presented in chapter 6 and used extensively for chapters 7 and 8. A modified version of Provost and Domingo's (2001) VUS estimate

was presented in section 9.2. In this thesis the author claims that these modified measures provide a modest extension to existing evaluation metrics for predictive modeling.

## 11.3 Limitations of the proposed dataset selection methods

Toulmin's argumentation model (Toulmin el al, 1979; Toulmin, 1958) which explains the structure of claims in scientific discourse, and Ngwenyama's (2007) analysis of scientific research claims were introduced in chapter 1. Recall that claims are supported by *data* (evidence*), warrants* (rules of inference) and *backing* (authoritative sources for *warrants*). Two additional components in Toulmin's model are *qualifiers* and *rebuttals*. *Qualifiers* are used to limit the strength of a *claim* and *rebuttals* provide an elaboration for the *qualifiers*. The claims made in this thesis concern the effectiveness of feature selection and training dataset selection, and aggregate modeling methods as discussed in chapters 5 to 8 and summarised in chapter 10.

A claim was made in chapter 5 that the use of many samples to measure class-feature and feature-feature correlations is an effective method for the accurate measurement of these correlations. However, the datasets used in the studies were of moderately high dimensionality. In practice there are many problem domains for which the dimensionality of the datasets are extremely high. The use of many samples to measure correlations coupled with robust correlation measures with quadratic time complexity is a daunting task. This issue was not addressed in this thesis and is left for future work. The *qualifier* for the *claim* is that the proposed methods of using many samples to measure correlations *are only appropriate when* the dimensionality of a dataset is not very high.

Claims were made in chapters 7, 8 and 9 that the proposed methods of base model design and training dataset selection for OVA and pVn modeling result in aggregate models that have a higher level of predictive performance compared to single *k*-class models. A *qualifier* was stated in section 8.5.3 that the proposed methods for boosted OVA and pVn model design are *only appropriate when* a dataset has a single *k*-class model confusion matrix with the sparse confusion matrix property. The situations where a non-sparse confusion matrix can be transformed into a sparse confusion matrix were also given in section 8.5.3.

## 11.4 Chapter Summary

The research outputs and claims of contributions in this thesis were assessed in the context of design science outputs and research contributions. The limitations of the proposed methods were also discussed. The conclusions for the thesis are presented in the next chapter.

# Chapter 12
# Conclusions

> 'You are my life. In you my peace, in you my joy, in you my strength, in you my God.'
> (Benjamin Dube, 2007)

## 12.1 Summary of the thesis

The central argument of this thesis is that, it is possible for predictive data mining to systematically select many dataset samples and employ different approaches (different from current practice) to feature selection, training dataset selection, and model construction. When a large amount of information in the large dataset is utilised in the modeling process, the resulting models should have a high level of predictive performance and should be reliable.

The discussions of chapters 2 argued that there is a need for methods for training dataset selection from large datasets, using as much data as possible with the objective of reducing the bias and variance components of the prediction error. The discussions of chapter 3 argued for the need for feature selection from large datasets, with the objective of using as much data as possible in order to reliably measure the class-feature and feature-feature correlations used in the feature selection process.

The experimental results of chapter 5 demonstrated that the use of the mean values for the correlations, obtained through the use of many samples, robust measure of correlations, and validation methods such as the use of fake variables, results in the identification of features which are relevant for the prediction task. The experimental results of chapter 5 also revealed that the incorporation of domain-specific definitions of the meaning of *low, medium* and *high* correlation into a feature subset search procedure results in the selection of good feature subsets for the prediction task at hand. The experimental results of chapters 7, 8 and 9 demonstrated that the use of the proposed methods for base model design and training dataset selection for OVA and pVn aggregate modeling has the potential to produce models which have a higher level of predictive performance compared to single models.

## 12.2 Conclusions and reflection

From a computational perspective it can be argued that the methods proposed in this thesis provide the following desirable outcomes: Firstly, the methods result in the use of large amounts of data which provide a large amount of information to the modeling process. Secondly, the methods for OVA and pVn modeling lead to the avoidance of un-necessary computations since the modeling effort is aimed at the creation of models that have a potential to increase predictive performance. From a statistical perspective it can be argued that the proposed methods provide the following desirable outcome: The methods result in the use of large amounts of data and at the same time avoid the problems of overfitting, data dredging and the modeling of phantom (chance) structure. From an Operations Research perspective it can be argued that the proposed methods provide the following desirable outcome: One of the uses of ROC analysis is used to determine the optimal operating point for a predictive model. The proposed OVA and pVn modeling methods have the potential to produce predictive models with higher optimal performance compared to single models.

It has been demonstrated that the use of large amounts of data with the methods proposed in this thesis, has the potential to provide predictive models with a high level of predictive performance. In general, no single method can be claimed to be suitable for all datasets and for all algorithms. Schaffer (1994) has argued that no single strategy for machine learning is better at generalisation (prediction) than all other strategies for all problem domains. In his study of noise-free datasets, Wolpert (1996) has demonstrated through the *no free lunch theorems for machine learning* that all algorithms are equivalent on average, in terms of predictive performance. The foregoing arguments can be easily extended to other computational domains. With the foregoing observations in mind, the author does not claim that the proposed methods will provide effective solutions for all data mining application domains. In order to establish the extent of applicability for the proposed methods additional empirical studies as discussed in the next section, will have to be conducted in future.

## 12.3 Future work

It was observed in chapter 5 that predictive features can be eliminated when robust correlation measures are used even when such features are good predictors for one or more local areas of the instance space. It will be useful in future to conduct studies for the identification of locally predictive features which are predictive of real structure as opposed to phantom (chance) structure. It was also observed in chapter 5 that predictive features for severely under-represented classes may be eliminated when robust correlation measures are employed. In future it will be useful to study feature selection methods that directly address this problem.

It was observed in chapters 5 and 10 that use of many samples to measure correlations coupled with robust correlation measures with quadratic time complexity is not a feasible approach for the estimation and validation of class-feature and feature-feature correlation coefficients for datasets of very high dimensionality. It will be useful in future to study feasible and reliable methods of correlation measurement for datasets of very high dimensionality.

The confusion matrix was used for the experiments of chapters 7 and 8 as a basis for the identification of confusion regions for a classification task. It will be useful in future to investigate other methods for the identification of confusion regions. Confusion graphs were used as input to the algorithms for determining the design of pVn models. The weights for the arcs of the confusion graphs were not used in the algorithm's decisions except in the case where a maximally connected graph had to be pre-processed. It will be useful to investigate how the arc weights in a confusion graph can be used to fine tune the decisions of these algorithms.

The dataset selection and aggregate modeling methods proposed in this thesis were directed at multi-class problems, and are not directly applicable to 2-class prediction problems unless a dataset is pre-processed through cluster analysis as discussed in section 8.5.3. It will be useful in future to investigate how the proposed OVA and pVn base model design and training dataset selection methods could be extended to 2-class problems.

It was stated in sections 8.5.3 that the proposed base model design and training dataset selection methods for boosted OVA and pVn aggregate models are only

applicable when the single *k*-class confusion matrix for a dataset has the *sparse confusion matrix* property. It will be useful to investigate different problem decomposition methods (different from OVA and pVn) for such datasets. For such problem decomposition methods it will be necessary to design training dataset selection methods for bias error reduction.

It was observed in chapters 7 and 10 that if an algorithm for the combination of base model predictions is able to resolve conflicting (tied) predictions then a high level of predictive performance is realised for an aggregate model. This was shown to be the case for 5NN classification. It will be useful in future to investigate methods of resolving conflicting (tied) predictions for classification tree algorithms.

# References

AHA, D. W. & BANKERT, R. L. (1996) A comparative evaluation of sequential feature selection algorithms IN FUSHER, D. & LENZ, H. J. (Eds.) *Learning from Data: Artificial Intelligence and Statistics* Springer-Verlag.

ALI, K. M. & PAZZANI, J. (1996) Error reduction through learning multiple descriptions. *Machine Learning,* 24,173-202.

ASUNCION, A. & NEWMAN, D. J. (2007) UCI machine learning repository [http://0-www.ics.uci.edu./~mlearn/MLRepository.html]. Irvine, CA, University of California, Department of Information and Computer Science.

BAY, S. D., KIBLER, D., PAZZANI, M. J. & SMYTH, P. (2000) The UCI KDD archive of large data sets for data mining research and experimentation. *ACM SIGKDD,* 2 (2), 81-85.

BEKKERMAN, R., EL-YANIV, R., TISHBY, N. & WINTER, Y. (2003) Distributional word clusters vs. words for text categorization. *Journal of Machine Learning Research,* 3, 1183-1208.

BERRY, M.J.A & LINOFF, G.S. (2000) *Mastering Data Mining: The Art and Science of Customer Relationship Management.* John Wiley & Sons.

BI, J., BENNET, K. P., EMBRECHTS, M., BRENEMAN, C. M. & SONG, M. (2003) Dimensionality Reduction via Sparse Support Vector Machines. *Journal of Machine Learning Research,* 3, 1229-1243.

BISHOP, C. M. (1995) *Neural Network for Pattern Recognition.* Oxford:Clarendon Press.

BLACKARD, J. A. (1998) Comparison of Neural Network and Discriminant Analysis in Predicting Forest Cover Types, PhD Thesis. *Department of Forest Sciences.* Fort Collins, Colorado, Colorado State University.

BLAKE, C. L. & MERZ, C. J. (1998) UCI Repository of Machine Learning Databases. *Department of Computer Science.* Irvine, University of California.

BLUM, A. L. & LANGLEY, P. (1997) Selection of relevant features and examples in machine learning. *Artificial Intelligence,* 97(1-2), 245-271.

BOSE, R. (2002) Customer Relationship Management: Key components for IT success. *Industrial Management and Data Systems,* 102 (2). 89-97.

BOSER, B.E., GUYON, I.M. & VAPNIK, V.N . (1992) A training algorithm for optimal margin classifiers. In D. HAUSSLER, editor, *5th Annual ACM Workshop on COLT*, 144-152.

BREIMAN, L. (1996) Bagging predictors. *Machine Learning,* 24, 123-140.

BREIMAN, L., FRIEDMAN, J. H., OLSHEN, R. A. & STONE, C. J. (1984) *Classification and Regression Trees.* Pacific Grove, CA:Wadsworth & Brooks.

CATLETT, G. (1991) Megainduction: a test flight. *Proceeding of Eighth Workshop on Machine Learning.* San Mateo, California, Morgan Kaufmann.

CHAN, P. & STOLFO, S. (1998) Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection. *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining.* AAAI

CHAWLA, N., MOORE, T. E., BOWYER, K. W., HALL, L. O., SPRINGER, C. & KEGELMEYER, P. (2001) Bagging is a small-data-set phenomenon. *In International Conference on Computer Vision and Pattern Recognition (CVPR), 2001.*

CLARK, D., SCHRETER, Z. & ADAMS, A. (1996) A quantitative comparison of distal and backpropagation. *Proceedings of the Seventh Australian Conference on Neural Networks (ACNN'96).* Canberra Australia.

COETSEE, J. (2007) *Private Communication.* Department of Statistics, University of Pretoria.

COHEN, J. (1988) *Statistical Power Analysis for the Behavioural Sciences, 2nd Edition.* Hillsdale: New Jersey Lawrence Erlbaum Associates.

COHEN, P.R. (1995) *Empirical Methods in Artificial Intelligence,* MIT Press: Cambridge, Massachusetts.

CORTEZ, P., CERDEIRA, F., ALMEIDA, F., MATOS, T. & REIS, J. (2009) Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems,* 42, 547-553.

COVER, T. M. & HART, P. E. (1967) Nearest Neighbor Pattern Classification. *IEEE Transaction on Information Theory,* IT-13 (1), 21-27.

DIETTERICH, T. (1995) Overfitting and Undercomputing in Machine Learning. *ACM Computing Surveys,* 27 (3), 326-327.

DIETTERICH, T. & BAKIRI, G. (1995) Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research,* 2, 263-286.

DIETTERICH, T. & KONG, E. (1995) Machine learning bias, statistical bias, and statistical variance of decision tree algorithms Technical Report. Corvallis, Oregon, Department of Computer Science, Oregon State University.

DIETTERICH, T. (1997) Fundamental experimental research in machine learning. Available at: http://web.engr.oregonstate.edu/~tgd/experimental-research/index.html (Cited: 27 October, 2009).

DIETTERICH, T. (1998) Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation,* 10, 1895-1923.

DOHERTY, K. A. J., ADAMS, R. G. & DAVEY, N. (2007) Unsupervised learning with normalised data and non-Euclidean norms. *Applied soft computing,* 7(1). 203-217.

DOMINGO, C., GALVADA, R. & WATANABE, O. (2002) Adaptive sampling methods for scaling up knowledge discovery algorithms. *Data Mining and Knowledge Discovery,* 6, 131-152.

DOMINGOS, P. (2000a) A unified bias-variance decomposition and its applications. In *Proceedings of the Seventeenth International Conference on Machine Learning ,* 231-238. CA:Morgan Kaufmann.

DOMINGOS, P. (2000b) Bayesian averaging of classifiers and the overfitting problem. In *Proceedings of the Seventeenth International Conference on Machine Learning,* 223-230. CA:Morgan Kaufmann.

DOMINGOS, P. (2001) When and how to subsample: Report on the KDD-2001 Panel. *SIGKDD Explorations,* 3(2), 74-75.

ENGELBRECHT, A. P. (2002) *Computational intelligence: An introduction,* West Sussex:John Wiley & Sons.

FAN, C., MULLER, M. & REZUCHA, I. (1962) Development of sampling plans by using sequential (item by item) selection techniques and digital computers. *Journal of the American Statistical Association* 57, 387-402.

FAN, W., LEE, W., STOLFO, J. & MILLER, M. (2000) A multiple model cost-sensitive approach for intrusion detection. *Lecture Notes in Computer Science*. Springer.

FAWCETT, T. (2001) Using rulesets to maximise ROC performance. *Proceedings of the IEEE International Conference on Data Mining (ICDM-2001),* 131-138.

FAWCETT, T. (2004) ROC graphs: Notes and practical considerations for researchers. HP Laboratories. Available from: http://home.comcast.net/~tom.fawcett/public_html/papers/ROC101.pdf (Cited: 1 March, 2010).

FAWCETT, T. (2006) An introduction to ROC analysis. *Pattern recognition Letters,* 27, 861-874.

FAYYAD, U., HAUSSLER, D. & STOLORZ, P. (1996) Mining Scientific Data *Communications of the ACM ,* 39 (11), 51-57.

FREUND, Y. & 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

FREY, L. J. & FISHER, D. H. (1999) Modeling Decision Tree Performance with the Power Law. IN HECKERMAN, D. & WHITTAKER, J. (Eds.) *Proceeding of the Seventh International Workshop on Artificial Intelligence and Statistics.* San Francisco, CA: Morgan Kauffman.

FRIEDMAN, J. (1997) On bias, variance, 0/1—loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery,* 1 (1), 55-77.

FU, Z., GOLDEN, B., LELE, S. & RAGHAVAN, S. (2003) Genetically engineered decision trees: population diversity produces smarter trees. *Operations Research,* 51 (6), 894-907.

FU, Z., GOLDEN, B.L., LELE, S., RAGHAVAN, S. & WASIL, E. (2006) Diversification for better decision trees. *Computers and Operations Research,* 51 (6), 894-907.

GEMAN, S., BIENENSTOCK, E. & DOURSAT, R. (1992) Neural networks and the bias/variance dilemma. *Neural computation,* 4, 1-58.

GIUDICI, P. (2003) *Applied Data Mining: Statistical Methods for Business and Industry,* Chichester:John Wiley & Sons.

GIUDICI, P. & FIGINI, s. (2009) *Applied Data Mining for Business and Industry, second edition,* Chichester:John Wiley & Sons.

GUYON, I. & ELISSEEFF, A. (2003) An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research,* 3 1157-1182.

HALL, L. O., BOWYER, K. W., KEGELMEYER, P., MOORE, T. E. & CHAO, C. (2000) Distributed learning on very large data sets. *Proceedings of the Sixth International ACM SIGKDD.*

HALL, M. A. (1999) Correlation-based Feature Selection for Machine Learning, PhD Thesis. *Department of Computer Science* Hamilton, New Zealand, University of Waikato.

HALL, M. A. (2000) Correlation-based feature selection for discrete and numeric class machine learning. *Proceedings of the Seventeenth International Conference on Machine Learning* San Francisco, CA, Morgan Kaufmann.

HALL, M. A. & HOLMES, G. (2003) Benchmarking attribute selection techniques for discrete class data mining. *IEEE Transactions on Knowledge and Data Engineering* 15 ( 6), 1437- 1447.

HAND, D. J. (1997) *Construction and Assessment of Classification Rules,* Chichester:John Wiley & Sons

HAND, D. J. (1998) Data mining: statistics and more? *The American Statistician,* 52 (2), 112-118.

HAND, D. J. (1999) Statistics and data mining: intersection disciplines. *SIGKDD Explorations,* 1 (1), 16-19.

HAND, D. J., MANILA, H. & SMYTH, P. (2001) *Principles of Data Mining,* Cambridge, MA:MIT Press.

HAND, D. J. & TILL, R.J. (2001) A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine Learning,* 45, 171-186.

HANLEY, J.A. & MCNEIL, B.J. (1982) The meaning and use of the area under the receiver operating characteristic (ROC) curve. *Radiology,* 143, 29-36.

HANSEN, L. K. & SALAMON, P. (1990) Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence. 1*2 (10), 993-1001.

HETTICH, S. & BAY, S. D. (1999) The UCI KDD archive [http://kdd.ics.uci.edu]. *Department of Information and Computer Science.* Irvine, CA, University of California.

HEVNER, A. R., MARCH, S. T., PARK, J. & RAM, S. (2004) Design science in information systems research. *MIS Quarterly,* 28 (1). 75-105.

HO, T. (1998) The random subspace method for constructing decision forests. *IEEE Trans. Pattern Analysis and Machine Intelligence,* 20 (8), 832-844.

HOEFFDING, W. (1963) Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association,* 58, 13-30.

IBA, W., WOGULIS, J. & LANGLEY, P. (1988) Trading off simplicity and coverage in incremental concept learning. IN ARBOR, A. (Ed.) *Proceedings of the 5th International Conference on Machine Learning.* Michigan:Morgan Kaufmann.

JAMES, G.M. (2003) Variance and bias for general loss functions. *Machine Learning,* 51,15-135.

JOHN, G. H. & LANGLEY, P. (1996) Static versus dynamic sampling for data mining. *Proceedings of the Second International Conference on Knowledge Discovery in Databases and Data Mining.* AAAI/MIT Press.

JONES, T. (1962) A note on sampling from tape files. *Communications of the ACM,* 5 ( 6). 343.

KANJI, G. (1999) *100 Statistical Tests,* London:Sage Publications.

KIM, E., KIM, W. & LEE, Y. (2002) Combination of multiple classifiers for the customer purchase behaviour prediction. *Decision Support Systems,* 34 ( 2), 167-175.

KINIVEN, J. & MANNILA, H. (1993) The power of sampling in knowledge discovery Technical Report C-1993-66. University of Helsinki.

KOHAVI, R., & JOHN, G.H. (1997) Wrappers for feature subset selection. *Artificial Intelligence,* 97, 273-324.

KOHAVI, R., MASON, R. J. & ZHENG, Z. (2004) Lessons and Challenges from Mining retail e-commerce data. *Machine Learning,* 57 83-113.

KOHAVI, R. & PROVOST, F. (1998) Glossary of terms. Special issue on applications of machine learning and the Knowledge Discovery process. *Machine Learning,* 30 (2). 271-274.

KOHAVI, R. & WOLPERT, D.H. (1996) Bias plus variance decomposition for zero-one loss functions. IN SAITTA, L. (Ed.) *Machine Learning: Proceedings of the Thirteenth International Conference,* 275-283. Morgan Kaufmann.

KONG, E. & DIETTERICH, T. (1995) Error-Correcting Output Coding Corrects Bias and Variance *Proceedings of the Twelfth International Conference on Machine Learning.* Morgan Kaufmann.

KROGH, A. & VEDELSBY, J. (1995) Neural network ensembles, cross validation and active learning. IN TESAURO, G., TOURETZKY, D. S. & LEEN, T. K. (Eds.) *Advances in Neural Information Processing Systems.* Cambridge MA: MIT Press.

KUBAT, M. & MATWIN, S. (1997) Addressing the curse of imbalanced data sets: One-sided selection. *Proceeding of the Fourteenth International Conference on Machine Learning.* San Francisco, CA, Morgan Kauffman.

KWOK, S. W. & CARTER, C. (1990) Multiple decision trees. *Uncertainty in Artificial Intelligence,* 4, 327-335.

LANGLEY, P., IBA, W. & THOMPSON, K. (1992) An analysis of Bayesian classifiers. *Proceedings, Tenth National Conference on Artificial Intelligence.* Menlo Park, CA, AAAI Press.

LASKOV, P., DÜSSEL, P., SCHÄFER, C. & RIECK, K. (2005) Learning intrusion detection: supervised or unsupervised? *ICAP: international conference on image analysis and processing.* Cagliari, Italy.

LEE, W., FAN, W., MILLER, M., STOLFO, S. & ZADOK, E. (2002) Toward cost-sensitive modeling for intrusion detection and response. *Journal of Computer Security,* 10 (1), 5-22.

LEE, W. & STOLFO, J. (2000) A framework for constructing features and models for intrusion detection systems. *ACM Transactions on Information and System Security,* 3 (4), 227-261.

LEUNG, K. & LECKIE, C. (2005) Unsupervised anomaly detection in network intrusion detection using clusters. IN ESTIVILL-CASTRO, V. (Ed.) *Proceedings of the Twenty-eighth Australasian conference on Computer Science* Newcastle, Australia, Australian Computer Society.

LINDEN, G., SMITH, B. & YORK, J. (2003) Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing,* 4(1).

LIU, H. & MOTODA, H. (1998) *Feature Selection for Knowledge Discovery and Data Mining.* Boston:Kluwer Academic Publishers.

LIU, H. & SETIONO, R. (1998a) Scalable feature selection for large sized databases. *Proceedings of the Fourth World Congress on Expert Systems (WCES'98)*. Morgan Kaufmann Publishers.

LIU, H. & SETIONO, R. (1998b) Some issues on scalable feature selection. *Expert Systems with Applications*, 15, 333-339.

LUGER, G. & STUBBLEFIELD, W. A. (1993) *Artificial Intelligence - Structures and Strategies for Complex Problem Solving, second edition.* CA:Benjamin Cummings Publishing Company.

LUTU, P. E. N. & ENGELBRECHT, A. P. (2006) A Comparative Study of Sample Selection methods for Classification. *South African Computer Journal,* 36, 69-85.

LUTU, P. E. N. & ENGELBRECHT, A. P. (2010) A decision rule-based method for feature selection in predictive data mining. *Expert Systems with Applications,* 37 (1), 602-609.

MANSON, N. J. (2006) Is operations research really research? *Orion,* 22 (2), 155-180.

MARCH, S. T. & SMITH, G. F. (1995) Design and natural science research on information technology. *Decision Support Systems,* 15, 251-266.

MARTÍNEZ-MUÑOZ, G., HERNÁNDEZ-LOBATO, D. & SUÁREZ, A. (2009) An analysis of ensemble pruning techniques based on ordered aggregation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2), 245-259.

MITCHELL, T. M. (1997) *Machine Learning,* Burr Ridge, IL:WCB/McGraw-Hill.

MONTGOMERY, D. C., RONGER, G. C. & HUBELE, N. F. (2004) *Engineering Statistics,* New York:Wiley.

MOORE, A.W. & LEE, M.S. (1994) ) Efficient algorithms for minimising cross validation error. In *Proceedings of the Eleventh International Conference on Machine Learning.* 190-198. New Brunswick, NJ: Morgan Kaufmann.

NEAGU, D., GUO, G. & WANG, S. (2006) An Effective Combination Based on Class-Wise Expertise of Diverse Classifiers for Predictive Toxicology Data Mining IN LI, X., ZAIANE, O. R. & LI, Z. (Eds.) *Advanced Data Mining and Applications.* Berlin, Springer Berlin / Heidelberg.

NEWELL, A. & SIMON, H. (1976) Computer science as empirical enemy: symbols and search. *Communication of the ACM,* 19 (2), 113-126.

NGWENYAMA, O. (2007) The seven basic claims of scientific research: an approach to analysing the structure of scientific argumentation in IS research papers. Working paper #iitm-2007-SR-187, Ryerson University, Toronto, Canada.

NGWENYAMA, O. K. & OSEI-BRYSON, K.-M. (2010) Using data mining to support information systems research: an approach for abduction and evaluation of hypotheses. To appear.

OATES, B. J. (1984) *Researching Information Systems and Computing.* London:SAGE Publications.

OLAFSSON, S., LI, X. & WU, S. (2008) Operations Research and data mining. *European Journal of Operations Research,* 19 (2) 113-126.

OLKEN, F. (1993) Random Sampling from Databases, PhD Thesis. *Department of Computer Science,* Berkeley. University of California at Berkeley.

OLKEN, F. & ROTEM, D. (1995) Random sampling from databases - A survey. (invited paper). *Statistics and Computing,* 5 (1), 25-42.

OOI, C. H., CHETTY, M. & TENG, S. W. (2007) Differential prioritization in feature selection and classifier aggregation for multiclass microarray datasets. *Data Mining and Knowledge Discovery,* 14, 329-366.

OSEI-BRYSON, K.-M. (2004) Evaluation of decision trees: a multi-criteria approach. *Computers and Operations Research,* 31 (11), 1933-1945.

OSEI-BRYSON, K.-M. (2007) Post-pruning in decision tree induction using multiple performance measures. *Computers and Operations Research,* 34, 3331-3345.

OSEI-BRYSON, K.-M. (2008) Post-pruning in regression tree induction: an integrated approach. *Expert Systems with Applications,* 34 (2), 1481-1490.

OSEI-BRYSON, K.-M., KAH, M. O. & KAH, J. M. L. (2008) Selecting predictive models for inclusion in an ensemble. *The 18th Triennial Conference of the International Federation of Operational Research Societies (IFORS 2008).* Sandton, Johannesburg, July 2008.

OSEI-BRYSON, K.-M. (2010) Towards supporting expert evaluation of clustering results using a data mining process model. *Information Sciences,* 180, 414-431.

PALMER, C. R. & FALOUTSOS, C. (2000) Density biased sampling: an improved method for data mining and clustering. *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data.* Dallas, Texas United States, ACM.

PEARL, J. (1984) *Heuristics: Intelligent Strategies for Computer Problem Solving,* Reading, MA:Addison-Wesley.

PHUA, C., LEE, V., SMITH, K. & GAYLER, R. (2005) A comprehensive survey of data mining-based fraud detection research. *Artificial Intelligence Review.* (SCI).

PROVOST, F., JENSEN, D. & OATES, T. (1999) Efficient progressive sampling. *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* San Diego, CA, ACM.

PROVOST, F. & DOMINGOS, P. (2001) Well trained PETS: improving probability estimation trees. Working paper #IS-00-04, Stern School of Business, New York University, New York, NY 10012.

PROVOST, F. & FAWCETT, T. (2001) Robust classification for imprecise environments. *Machine Learning,* 42, 203-231.

QUINLAN, J. R. (1986) Induction of decision trees. *Machine Learning,* 1 81-106.

QUINLAN, J. R. (1993) *C4.5: Programs for Machine Learning,* San Francisco, CA:Morgan Kauffman.

QUINLAN, J. R. (2004) An Informal Tutorial, Rulequest Research. URL: http://www.rulequest.com (accessed: 28 October, 2005).

RAO, P. S. R. S. (2000) *Sampling Methodologies with Applications,* CRC, Florida:Chapman & Hall.

RIFKIN, R. & KLAUTAU, A. (2004) In defense of one-vs-all classification. *The Journal of Machine Learning Research,* 5, 101-141.

RYGIELSKI, C., WANG, J.-C. & YEN, D. C. (2002) Data Mining techniques for customer relationship management. *Technology in Society,* 24, 483-502.

SAMOILENKO, S. & OSEI-BRYSON, K.-M. (2008) Increasing the discriminatory power of DEA in the presence of the sample heterogeneity with cluster analysis and decision trees. *Expert Systems with Applications,* 34, 1568-1581.

SCHAFFER, C. (1994) A conservation law for generalisation performance. *Proceedings of the Eleventh Conference on Machine Learning,* 259-265, CA: Morgan-Kaufmann.

SCHAPIRE, R. (2003) The boosting approach to machine learning: An overview. *MSRI Workshop on Nonlinear Estimation and Classification.* Springer.

SCHLIMMER, J. S. (1987) Concept acquisition through representational adjustment (Technical Report 87-19). Doctoral dissertation. *Department of Information and Computer Science.* Irvine, University of California.

SHANON, C. E. & WEAVER, W. (1962) *The Mathematical Theory of Communication,* Urbana:University of Illinois Press.

SHEARER, C. (2000) The CRISP-DM model: the new blue print for data mining. *Journal of Data Warehousing,* 5(4), 13-22.

SHIN, S. W. & LEE, C. H. (2006) Using Attack-Specific Feature Subsets for Network Intrusion Detection. *Proceedings of the 19th Australian Conference on Artificial Intelligence.* Hobart, Australia.

SIMON, H. A. (1996) *The Science of the Artificial, 3rd Edition.* Cambridge, MA:MIT Press.

SMYTH, P. (2001) Data Mining at the interface of computer science and statistics. IN GROSSMAN, R. L., KAMATH, C., KEGELMEYER, P., KUMAR, V. & NAMBURU, R. R. (Eds.) *Data Mining for Scientific and Engineering Applications.* Dordrech, Netherlands, Kluwer Academic Publishers.

STIRLING, W. D. (2008) CAST - Computer Assisted Statistics Teaching. *Massey University, NZ.*

STOLFO, S. J., FAN, W., LEE, W., PRODROMIDIS, A. & CHAN, P. (2000) Cost-based modeling for fraud and intrusion detection: results from the JAM project. *DARPA Information Survivability Conference and Exposition.* Hilton Head, SC, USA.

STOLORZ, P. & DEAN, C. (1996) QuakeFinder: A scalable data mining system for detecting earthquake from the space. *Proceedings of the Second International Conference on Data Mining KDD-96.* Portland, Oregon, AAAI.

STOPPIGLIA, H., DREYFUS, G., DUBOIS, R. & OUSSAR, Y. (2003) Ranking a random feature for variable and feature selection. *Journal of Machine Learning Research,* 3, 1399-1414.

SUN, J. & LI, H. (2008) Financial distress prediction based on serial combination of multiple classifiers. *Expert Systems With Applications,* 35 (5), 818-827.

THEUSINGER, C. & HUBER, K. P. (2000) Analysing the footsteps of your customer. *Proceedings of WEBKDD-2000.*

THOMAS, D. B., LUK, W., P.H.W., L. & J.D., V. (2007) Gaussian random number generators. *ACM Computer Survey,* 39 (4), 11:1-11:38.

TOIVONEN, H. (1996) Sampling large databases for association rules. *Proceedings of the Twenty-second Conference on Very Large Databases – VLDDB96.* Mumbai India, Morgan Kaufmann Publishers.

TOULMIN, S. E. (1958) *The Uses of Argumentation.* Cambridge, United Kingdom:Cambridge University Press.

TOULMIN, S., RIEKE, R. & JANIK, A. (1979) *An Introduction to Reasoning.* New York:Macmillan Publishing Co.

VAISHNAVI, V. & KUECHLER, W. (2004/5) Design Research in Information Systems. URL: http://desrist.org/design-research-in-information-systems (accessed 27 October, 2009).

VALIANT, L. G. (1984) A theory of the learnable. *Communications of the ACM,* 27 (11), 1134-1142.

VAN DER PUTTEN, P. & VAN SOMEREN, M. (2004) A bias-variance analysis of a real world learning problem: the COIL challenge 2000. *Machine Learning,* 57, 177-195.

VAPNIK, V. N. & CHERVONENKIS, A. Y. (1971) On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications,* 16, 264-280.

VUK, M. & CURK, T. (2006) ROC curve, lift chart and calibration plot. *Metodološki Zvezki,* 3, 89-108.

WATANABE, O. (2005) Sequential sampling techniques for algorithmic learning theory. *Theoretical Computer Science,* 348, 3-14.

WAUGH, S. (1995) Extending and Benchmarking Cascade-Correlation, PhD thesis. *Computer Science Department.* Hobart, Tasmania, University of Tasmania.

WILCOX, R. R. (2001) *Fundamentals of Modern Statistical Methods: Substantially Improving Power and Accuracy,* New York:Springer-Verlag.

WITTEN, H. I. & FRANK, E. (2005) *Data Mining: Practical Machine Learning Tools and Techniques, second edition,* San Francisco:Morgan Kaufmann.

WOLPERT, D. H. (1996) The lack of a priori distinctions between learning algorithms. *Neural Computation,* 8(7) 1341-1390.

WOLPERT, D. H. & Macready, G. (1997) No free lunch theorems for optimisation. *IEEE Transactions on Evolutionary Computation,* 1(1), 67-82.

WU, X., KUMAR, V., QUINLAN, J.R., GHOSH, J., YANG, Q., MOTODA, H., McLACHLAN, G.J., NG, A., LIU, B., YU, P.S., ZHOU, Z.-H., STEINBACH, M., HAND, D.J. & STEINBERG, D. (2008) Top 10 algorithms in data mining. *Knowledge Information Systems,* 14, 1-37.

YU, L. & LIU, H. (2004) Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research,* 5, 1205-1224.

# Appendices

The appendices in this section provide definitions of the symbols used, general definitions of statistical measures and descriptive statistics for the datasets used in the experiments. Details of correlation measurements, details for the 5NN aggregation algorithm, and OVA and pVn model performance are provided. Information is also provided on suggestions on how to use commonly available statistical and database software to implement some of the steps for the proposed feature and training dataset selection methods. Finally, a list of publications and conference presentations arising from the research is given. The table below summarises the appendix contents.

*Table of appendices*

| Appendix | Title | Description |
|---|---|---|
| A | Definition of symbols | Definition of symbols used in the thesis |
| B | Definitions of statistical measures | Definitions of statistical measures used in the thesis |
| C | Descriptive statistics for datasets | Descriptive statistics for forest cover type, KDD Cup 1999, Abalaone3C and mushroom |
| D | Correlation measurements | Details of correlation measurements and feature selection of chapter 4 |
| E | Algorithm for 5NN aggregation | Details of algorithm for the combination of 5NN base model predictions |
| F | Predictive performance of OVA and pVn models | Detailed results for accuracy for single and aggregate models for chapters 7 and 8. |
| G | ROC analysis details | Computation of the AUC for one-versus-rest ROC analysis. Details of AUC computation results. |
| H | Using statistical and database software to implement dataset selection methods | Suggestions for using commonly available statistical and database software to implement dataset selection |
| I | Publications and Conference Presentations | Publications and conference presentations arising out of the work reported in this thesis |

# Appendix A

# Definition of symbols

*Table A.1: Symbols used in the thesis*

| Symbol | Meaning |
|---|---|
| $accuracy$ | The predictive accuracy of a model |
| $B_i,..,B_v$ | Binary features created through the process of binarisation of a qualitative feature with *v* levels |
| $corr(X,Y)$ | The sample correlation coefficient between two random variables *X* and *Y* |
| $corr_{cf}(f)$ | The sample correlation coefficient between a feature *f* and a class variable *C* |
| $corr_{ff}(f)$ | The mean correlation between feature *f* and all other currently selected features |
| $c_1,..,c_k$ | The *k* levels of a class variable (number of classes for a prediction task) |
| $C$ | A class variable for classification |
| $conf$ | Probabilistic score assigned by a model to a class prediction as the level of confidence in the prediction |
| $d$ | The number of predictive features (variables) that define the *d*-dimensional instance space for classification modeling |
| $1-\delta$ | The probability of a learner being able to induce a hypothesis from data as in PAC |
| $error$ | The prediction error of a model |
| $error_D, error_R$ | Error difference and error ratio for measuring performance gains |
| $error_S, error_A$ | Prediction errors of a single model and aggregate model respectively |
| $\varepsilon$ | Prediction error as in PAC |
| $E$ | Entropy function |
| $f$ | A feature (predictor) used in predictive modeling |
| $\phi$ | The phi coefficient for measuring the level of association between two qualitative variables |
| $g_i$ | A region of the instance space |
| $G$ | The Gini concentration coefficient |
| $h$ | A hypothesis as defined in machine learning |
| $H$ | A set of hypotheses as defined in machine learning |
| $H_0$ *and* $H_a$ | The null hypothesis and alternative hypotheses for statistical hypothesis testing |
| $k$ | Number of classes for a classification problem |
| $K$ | Number of folds for cross validation |
| $L_1,..,L_V$ | Levels of a qualitative (nominal or ordinal) variable |
| $\lambda$ | Cut-off score value for ROC analysis |

Table A.1 continued

| Symbol | Meaning |
|---|---|
| $m$ | A mapping or a function |
| $M_A$ | General reference to a predictive model |
| $\mu_A$ | The population mean value of predictive accuracy of a model A |
| $n$ | The size of a sample taken from a parent dataset |
| $n_t$ | For sequential random sampling, $n_t$ is the number of records already selected |
| $N$ | The size of the parent dataset / database from which samples are taken |
| $ova_i$ | The i[th] sub-problem for the prediction of class $c_i$ in OVA classification |
| $p$ | Probability of obtaining an experimental result given that the null hypothesis is true (p value) |
| $P$ | Percentage value for a confidence interval (*P*% confidence interval) |
| $P_r$ | Probability |
| $PT$ | The number of partitions of a parent dataset |
| $pred$ | Output of a predictive model |
| $\pi_c$ and $\pi_d$ | The probabilities of concordance and discordance used in the computation of Kendall's tau |
| $r_{XY}$ , $r$ | Pearson's sample correlation coefficient for two random variables *X* and *Y* |
| $\tau_{XY}$ | Kendall's sample correlation coefficient for two random variables X and Y |
| $R^d$ | Super domain of real values for the random variables $X_1,..X_d$ |
| $RMsize, RQsize$ | For sequential random sampling, RMsize is the number of records still to be processed; RQsize is the number of records required for the sample |
| $S_X$ | The sample standard deviation for random variable X |
| $SU$ | Symmetrical uncertainty coefficient |
| $\sigma_X$ | The population standard deviation for random variable X |
| $si$ and $spi$ | Situations for feature subset search |
| $t$ | For sequential random sampling, t is the number of records processed so far |
| $T$ | The *T* statistic for statistical hypothesis testing |
| $u$ | Number of unselected features for heuristic feature subset search |
| $v$ | Number of levels for a qualitative variable |
| $V$ | Cramer's V statistic for measuring the level of association between two qualitative variables |
| $VC(H)$ | The Vapnik-Chervonenkis dimension of a set of hypotheses *H* for a learning task |
| $w$ | Number of features currently selected/processed by a feature selection method/algorithm |
| $W$ | Number of candidate features for heuristic feature subset search |

| Table A1 continued | |
|---|---|
| Symbol | Meaning |
| $\boldsymbol{x}$ and $x_1,..x_d$ | A vector of predictive features (predictor variable ) values ( an instance) |
| $x_q$ | A query (or test) instance to be classified / assigned a predicted value |
| $X, Y$ | Random variables |
| $Z$ | The Z statistic for statistical hypothesis testing |
| $Z_P$ | Constant for the calculation of the *P%* confidence interval of the mean |
| $z\%$ | Percentage of values to remove from each tail when winsorising variable values |
| **Confusion matrix and ROC analysis symbols:** | |
| $Pos$ | Total number of positive instances |
| $Neg$ | Total number of negative instances |
| $TP$ | Number of positive instances predicted as positive |
| $FN$ | Number of positive instances predicted as negative |
| $TN$ | Number of negative instances predicted as negative |
| $FP$ | Number of negative instances predicted as positive |
| $TPRATE$ | Fraction of the positive instances predicted as positive |
| $FNRATE$ | Fraction of the positive instances predicted as negative |
| $TNRATE$ | Fraction of the negative instances predicted as negative |
| $FPRATE$ | Fraction of the negative instances predicted as positive |
| $YRATE$ | Fraction of test instances predicted as positive (used for lift analysis) |

# Appendix B

# Definitions of statistical measures

A detailed discussion of the statistical measures used in this thesis is provided in this appendix. The entropy measure, Gini index of concentration, and measures of association (correlation) were used in the discussions of chapters 3, 4, 5 and 7.

## B.1 Entropy definitions

The entropy function *E(X)* (Giudici, 2003; Shanon & Weaver, 1962) measures the amount of uncertainty, heterogeneity, information or randomness in the values of the qualitative or quantitative discrete random variable *X* and is defined as

$$E(X) = -\sum_i P_r(x_i) \log_2 P_r(x_i) \tag{B.1}$$

where $P_r(x_i)$ which is used as a shorthand notation for $P_r(X = L_t)$ is the probability that variable *X* has the value (level) $L_t$. The entropy of the random variable *X*, conditioned on the values of a second random variable *Y* is denoted as *E(X|Y)* and is defined as

$$E(X \mid Y) = -\sum_j P_r(y_j) \sum_i P_r(x_i \mid y_j) \log_2 P_r(x_i \mid y_j) \tag{B.2}$$

where $P_r(x_i \mid y_i)$ which is used as a shorthand for $P_r\left((X = L_i) \mid (Y = L_j)\right)$ is the conditional probability that random variable *X* has the value (level) $L_t$ given that random variable *Y* has the value (level) $L_j$ and is defined as

$$P_r(x_i \mid y_j) = \frac{P_r(x_i, y_j)}{P_r(y_j)} \tag{B.3}$$

where $P_r(x_i, y_j)$ is the probability of values $x_i$ and $y_j$ appearing together. The joint entropy of two random variables *X* and *Y* denoted as *E(X,Y)* is defined as

$$E(X,Y) = -\sum_i P_r(x_i, y_j) \log_2 P_r(x_i, y_j) \tag{B.4}$$

The difference between the entropy of *X, E(X)* and the entropy of *X* conditioned on *Y*, *E(X|Y)* is called the information gain *IG(X,Y)* and is defined as

$$IG(X,Y) = E(X) - E(X \mid Y) \tag{B.5}$$

$$IG(X,Y) = E(Y) - E(Y \mid X) \tag{B.6}$$

$$IG(X,Y) = E(X) + E(Y) - E(X,Y) \tag{B.7}$$

The information gain measures the amount of reduction in the entropy of *X* when the values of *X* are grouped based on the values of *Y*. As indicated by the equations (B.5) and (B.6), information gain *IG(X,Y)* is a symmetric measure from which the symmetrical uncertainty coefficient *SU* is derived. The *SU* coefficient is defined as

$$SU = 2.0x\left[\frac{IG(X,Y)}{E(X) + E(Y)}\right] \tag{B.8}$$

The SU coefficient was used for the experiments of chapters 5 and 7 as a measure of correlation (association) for qualitative features.

## B.2 Measures of association

### B.2.1 Pearson's correlation coefficient

Pearson's sample correlation coefficient, *r* (Wilcox, 2001), between two random variables *X* and *Y* is defined as

$$r_{XY} = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{(n-1)S_X S_Y} \tag{B.9}$$

where $S_X$ and $S_Y$ are the standard deviations of *X* and *Y* respectively, and *n* is the sample size.

### B.2.2 Kendall's correlation coefficient

Kendall's rank correlation coefficient *tau* (Wilcox, 2001) is defined as

$$\tau = \pi_c - \pi_d \tag{B.10}$$

where $\pi_c$ and $\pi_d$ are the probabilities of concordance and discordance respectively. A pair of observations, $(x_1, y_1)$ and $(x_2, y_2)$ shows concordance if $x_1 > x_2$ and $y_1 > y_2$ or $x_1 < x_2$ and $y_1 < y_2$, otherwise the pair shows discordance. The values $\pi_c$ and $\pi_d$ are computed for all possible pairs for a data sample. For a data sample of size *n*, there are $\frac{n(n-1)}{2}$ possible pairs. However, some pairs will be tied i.e. having neither concordance nor discordance.

### B.2.3 Pearson's chi-square statistic

Pearson's chi-square statistic measures the level of association between two qualitative random variables *X* and *Y* (Giudici, 2003). The statistic is computed using the frequencies in a contingency table. A contingency table is a cross-tabulation which gives the frequencies of co-occurrence of the values (levels) of the variables *X* and *Y*. Pearson's chi-square statistic is defined as

$$\chi^2 = \sum_{i=1}^{I}\sum_{j=1}^{J}\frac{(n_{ij} - n^*_{ij})^2}{n^*_{ij}} \tag{B.11}$$

where *I* and *J* are respectively the number of rows and columns in the contingency table, $n_{ij}$ are the observed frequencies in the cells of the contingency table and, $n^*_{ij}$ are the expected frequencies for the cells of the contingency table under the null hypothesis of independence between *X* and *Y*.

The $\phi$ coefficient and Cramer's *V* coefficients are derived from Pearson's chi-square coefficient, and have the same interpretation as Pearson's *r* coefficient. The $\phi$ coefficient is defined as (Giudici, 2003)

$$\phi^2 = \frac{\chi^2}{n} \tag{B.12}$$

and Cramer's *V* coefficient is defined as

$$v^2 = \frac{\chi^2}{n.\min\{I-1, J-1\}} \tag{B.13}$$

The $\phi$ coefficient, Cramer's *V* coefficient, and symmetrical uncertainty coefficient can all be used to measure the level of association between two qualitative features.

## B.3 Gini concentration coefficient

Suppose there are *n* entities on which a given property *EP* has been measured yielding *n* pairs of measurement values $\{(1, EP_1),...,(i, EP_i),..,(n, EP_n)\}$ where *i* identifies the $i^{th}$ entity and $EP_i$ identifies the measurement value for the $i^{th}$ entity. Let $F_i$ be the cumulative percentage of the count of entities from the first to the $i^{th}$ entity. Let $Q_i$ be the cumulative percentage of the measurement values from the first measurement, $EP_1$ to the $i^{th}$ measurement, $EP_i$. A summary statistic of the concentration of the measured property *EP* among the *n* entities is called the Gini concentration coefficient defined as

$$Gini = \frac{\sum_{i=1}^{n-1}(F_i - Q_i)}{\sum_{i=1}^{n-1} F_i} \tag{B.14}$$

The *Gini* measure equals 0 for minimum concentration and 1 for maximum concentration. Minimum concentration means that all *n* entities have equal values of the property *EP*. Maximum concentration means that only one entity possesses the property *EP* and all other *n-1* entities have a value of 0 for *EP*.

The Gini concentration coefficient is related to the Area Under the ROC curve (AUC) as follows: The EP property corresponds to the scores that are assigned by a probabilistic classifier. The AUC was discussed in section 4.7.

## B.4 Computation of confidence intervals for the mean

A *P%* confidence interval for the mean is an interval that is expected with probability *P%* to contain the true value of the population mean (Mitchell, 1997). Laplace's estimate of the confidence interval of the population mean is defined as

$$CI = \left( \bar{x} - Z_P \frac{S_X}{\sqrt{n}}, \bar{x} + Z_P \frac{S_X}{\sqrt{n}} \right) \tag{B.15}$$

where $\bar{x}$ is the sample mean for random variable *X*, $S_X$ is the sample standard deviation, and *n* is the sample size (Wilcox, 2001; Mitchell, 1997). Different values of $Z_P$ are used to obtain different confidence intervals. A value of $Z_P = 1.96$ is used for the 95% confidence interval. A value of $Z_P = 2.58$ is used for the 99% confidence interval (Wilcox, 2001; Mitchell, 1997).

# Appendix C

# Descriptive statistics for the datasets

The descriptive statistics for the datasets used in the experiments are presented in this section.

## C.1 Forest cover type dataset

Figure C.1 provides the class frequencies and a graphic representation for the forest cover type dataset classes. Tables C.1 and C.2 show the descriptive statistics for the qualitative and quantitative variables in the forest cover type dataset.

| Class label | Class name | Percent |
|---|---|---|
| 1 | Spruce / fir | 36.5 |
| 2 | Lodgepole pine | 48.8 |
| 3 | Panderosa pine | 6.2 |
| 4 | Cottonwood / Willow | .5 |
| 5 | Aspen | 1.6 |
| 6 | Douglas-fir | 3.0 |
| 7 | Krummholz | 3.5 |

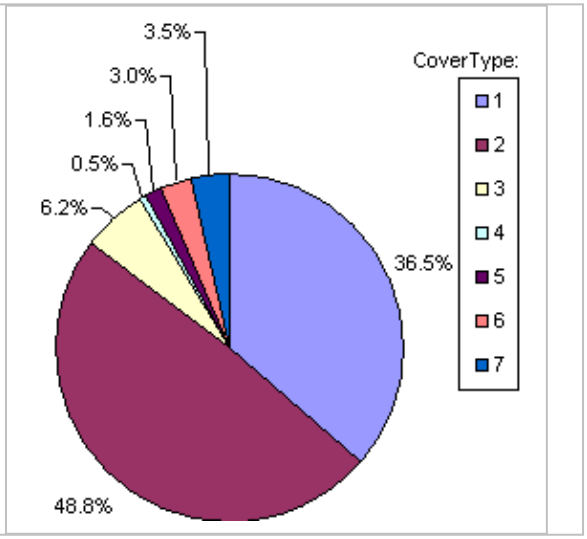


*Figure C.1: Class frequencies for the forest cover type class variable (covertype)*

*Table C.1: Descriptive statistics for the quantitative variables in the forest cover type dataset*

| | Minimum | Maximum | Mean | Standard Deviation | Coefficient of variation (CV) |
|---|---|---|---|---|---|
| Aspect | 0 | 360 | 155.7 | 111.9 | 0.7 |
| Elevation | 1859 | 3858 | 2959.4 | 280.0 | 0.1 |
| Slope | 0 | 66 | 14.1 | 7.5 | 0.5 |
| HorizDistToHydro | 0 | 1397 | 269.4 | 212.5 | 0.8 |
| VertDistToHydro | -173 | 601 | 46.4 | 58.3 | 1.3 |
| HorizDistToRoad | 0 | 7117 | 2350.2 | 1559.3 | 0.7 |
| HillShade9am | 0 | 254 | 212.2 | 26.8 | 0.1 |
| HillShadeNoon | 0 | 254 | 223.3 | 19.8 | 0.1 |
| HillShade3pm | 0 | 254 | 142.5 | 38.3 | 0.3 |
| HorizDistToFire | 0 | 7173 | 1980.3 | 1324.2 | 0.7 |

*Table C.2: Descriptive statistics for the qualitative variables for the forest cover type dataset*

| Variable name | Percentage for '0' | Percentage for '1' | Variable name | Percentage for '0' | Percentage for '1' |
|---|---|---|---|---|---|
| WildernessArea1 | 55.1 | 44.9 | SoilType19 | 99.3 | 0.7 |
| WildernessArea2 | 94.9 | 5.1 | SoilType20 | 98.4 | 1.6 |
| WildernessArea3 | 56.4 | 43.6 | SoilType21 | 99.9 | 0.1 |
| WildernessArea4 | 93.6 | 6.4 | SoilType22 | 94.3 | 5.7 |
| SoilType1 | 99.5 | 0.5 | SoilType23 | 90.1 | 9.9 |
| SoilType2 | 98.7 | 1.3 | SoilType24 | 96.3 | 3.7 |
| SoilType3 | 99.2 | 0.8 | SoilType25 | 99.9 | 0.1 |
| SoilType4 | 97.9 | 2.1 | SoilType26 | 99.6 | 0.4 |
| SoilType5 | 99.7 | 0.3 | SoilType27 | 99.8 | 0.2 |
| SoilType6 | 98.9 | 1.1 | SoilType28 | 99.8 | 0.2 |
| SoilType7 | 99.98 | 0.02 | SoilType29 | 80.2 | 19.8 |
| SoilType8 | 99.97 | 0.03 | SoilType30 | 94.8 | 5.2 |
| SoilType9 | 99.8 | 0.2 | SoilType31 | 95.6 | 4.4 |
| SoilType10 | 94.4 | 5.6 | SoilType32 | 91 | 9 |
| SoilType11 | 97.9 | 2.1 | SoilType33 | 92.2 | 7.8 |
| SoilType12 | 94.8 | 5.2 | SoilType34 | 99.7 | 0.3 |
| SoilType13 | 97 | 3 | SoilType35 | 99.7 | 0.3 |
| SoilType14 | 99.9 | 0.1 | SoilType36 | 100 | 0 |
| SoilType15 | 100 | 0 | SoilType37 | 99.9 | 0.1 |
| SoilType16 | 99.5 | 0.5 | SoilType38 | 97.3 | 2.7 |
| SoilType17 | 99.4 | 0.6 | SoilType39 | 97.6 | 2.4 |
| SoilType18 | 99.7 | 0.3 | SoilType40 | 98.5 | 1.5 |

## C.2 KDD Cup 1999 dataset

Figure C.2 provides the class frequencies and a graphic representation for the KDD Cup 1999 dataset classes. Tables C.3 and C.4 give the descriptive statistics for the variables in the KDD Cup 1999 dataset.

| Class | Frequency | Percent |
|---|---|---|
| DOS | 10851 | 20.9 |
| NORMAL | 35794 | 68.9 |
| PROBE | 4107 | 7.9 |
| R2L | 1126 | 2.2 |
| U2R | 52 | 0.1 |
| Total | 51930 | 100.0 |

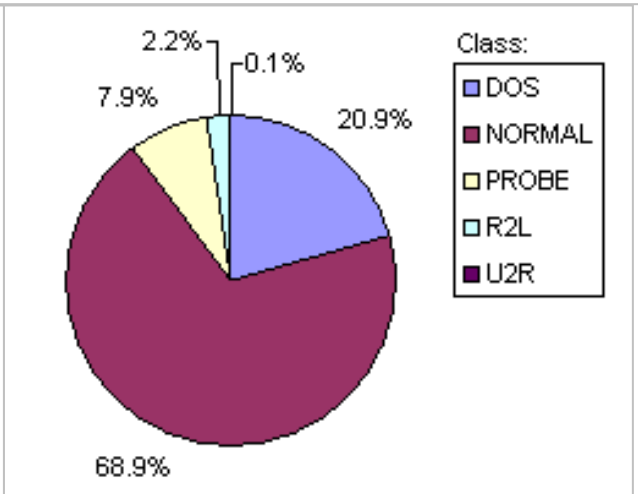


*Figure C.2: Class frequencies for the KDD Cup 1999 training dataset derived class variable (class)*

*Table C.3: Descriptive statistics for the quantitative variables for the KDD Cup 1999 training dataset*

| Variable name | Minimum | Maximum | Mean | Standard Deviation | Coefficient of variation (CV) |
|---|---|---|---|---|---|
| Counted | 0 | 511 | 53.3 | 120.4 | 2.3 |
| DiffSrvRate | 0 | 1 | 0.1 | 0.2 | 2.0 |
| DstBytes | 0 | 5,155,468.00 | 3,758.50 | 99,612.90 | 26.5 |
| DstHostCount | 1 | 255 | 191 | 93.2 | 0.5 |
| DstHostDiffSrvRate | 0 | 1 | 0.2 | 0.3 | 1.5 |
| DstHostRerrorRate | 0 | 1 | 0.1 | 0.2 | 2.0 |
| DstHostSameSrcPortRate | 0 | 1 | 0.3 | 0.4 | 1.3 |
| DstHostSameSrvRate | 0 | 1 | 0.6 | 0.4 | 0.7 |
| DstHostSerrorRate | 0 | 1 | 0.1 | 0.3 | 3.0 |
| DstHostSrvCount | 1 | 255 | 120.9 | 107.3 | 0.9 |
| DstHostSrvDiffHostRate | 0 | 1 | 0 | 0.1 | undefined |
| DstHostSrvRerrorRate | 0 | 1 | 0.1 | 0.2 | 2.0 |
| DstHostSrvSerrorRate | 0 | 1 | 0.1 | 0.3 | 3.0 |
| Duration | 0 | 58,329.00 | 455.5 | 2,140.00 | 4.7 |
| Hot | 0 | 30 | 0.3 | 2.4 | 8.0 |
| NumAccessFiles | 0 | 8 | 0 | 0.1 | undefined |
| NumCompromised | 0 | 884 | 0.1 | 5.5 | 55.0 |
| NumFailedLogins | 0 | 5 | 0 | 0 | undefined |
| NumFileCreations | 0 | 28 | 0 | 0.3 | undefined |
| NumOutboundCmds | 0 | 0 | 0 | 0 | undefined |
| NumRoot | 0 | 993 | 0.1 | 6.2 | 62.0 |
| NumShells | 0 | 2 | 0 | 0 | undefined |
| RerrorRate | 0 | 1 | 0.1 | 0.2 | 2.0 |
| RootShell | 0 | 1 | 0 | 0 | undefined |
| SameSrvRate | 0 | 1 | 0.8 | 0.4 | 0.5 |
| SerrorRate | 0 | 1 | 0.1 | 0.3 | 3.0 |
| SrcBytes | 0 | 693,000,000.00 | 23,327.40 | 3,047,960.00 | 130.7 |
| SrvCount | 0 | 511 | 20 | 73.9 | 3.7 |
| SrvDiffHostRate | 0 | 1 | 0.1 | 0.3 | 3.0 |
| SrvRerrorRate | 0 | 1 | 0.1 | 0.3 | 3.0 |
| SrvSerrorRate | 0 | 1 | 0.1 | 0.3 | 3.0 |
| SUAttempted | 0 | 2 | 0 | 0 | undefined |
| Urgent | 0 | 3 | 0 | 0 | undefined |
| WrongFragment | 0 | 3 | 0.1 | 0.4 | 4.0 |

*Table C.4: Descriptive statistics for the qualitative variables for the KDD Cup 1999 training dataset*

| Variable | Level description | Level names | Frequency% |
|---|---|---|---|
| ProtocolType | 3 levels | icmp | 7.3 |
| | | tcp | 53.5 |
| | | udp | 39.2 |
| Service | 64 levels | domain_u | 11.3 |
| | | ftp_data | 9.1 |
| | | http | 14.3 |
| | | private | 19.4 |
| | | smtp | 9.9 |
| | | all other services | 36 |
| Flag | 9 levels | SF | 82.1 |
| | | S0 | 10.7 |
| | | all other flags | 7.2 |
| Land | 2 levels | 0 | 99.96 |
| | | 1 | 0.04 |
| LoggedIn | 2 levels | 0 | 67 |
| | | 1 | 33 |
| IsHostLogin | 2 levels | 0 | 100 |
| | | 1 | 0 |
| IsGuestLogin | 2 levels | 0 | 98.7 |
| | | 1 | 1.3 |

## C.3 Abalone3C dataset

Figure C.3 provides the class frequencies and graphic representation for the abalone3C dataset classes. Table C.5 gives the descriptive statistics for the variables.

| Class | Frequency | Percentage |
|---|---|---|
| young | 1407 | 33.7 |
| middle | 1447 | 34.6 |
| old | 1323 | 31.7 |
| Total | 4177 | 100.0 |

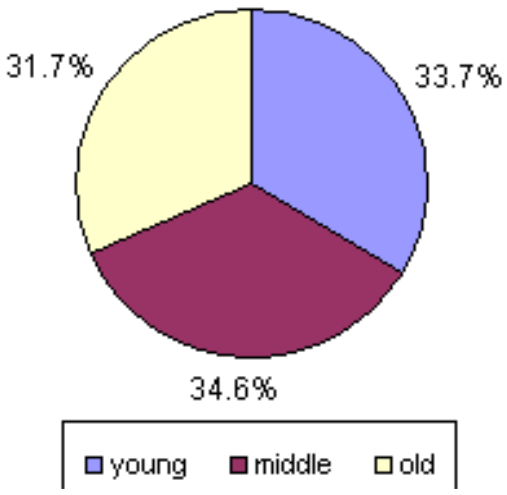


*Figure C.3: Class frequencies for the abalone3C class variable (age)*

*Table C.5: Descriptive statistics for the quantitative variables of abalone3C*

| Variable | Minimum | Maximum | Mean | Standard Deviation | Coefficient of variation (CV) |
|---|---|---|---|---|---|
| Length | 15.0 | 163.0 | 104.8 | 24.0 | 0.2 |
| Diameter | 11.0 | 130.0 | 81.6 | 19.8 | 0.2 |
| Height | 0.0 | 226.0 | 27.9 | 8.4 | 0.3 |
| WholeWeight | 0.4 | 565.1 | 165.7 | 98.1 | 0.6 |
| ShuckledWeight | 0.2 | 297.6 | 71.9 | 44.4 | 0.6 |
| VisceraWeight | 0.1 | 152.0 | 36.1 | 21.9 | 0.6 |
| ShellWeight | 0.3 | 201.0 | 47.8 | 27.8 | 0.6 |

The qualitative variable gender has three levels with absolute frequencies of: 1528 for male (M), 1307 for female (F) and 1342 for infant (I).

## C.4 Wine quality datasets

Figure C.4 provides the class frequencies and graphic representation for the wine quality (white) dataset classes. The two minority classes: 3 (20 instances) and 9 (5 instances) were removed from the dataset. Table C.6 gives the descriptive statistics for the variables.

| Class | Frequency | Percentage |
|---|---|---|
| 4 | 163 | 3.3 |
| 5 | 1457 | 29.9 |
| 6 | 2198 | 45.1 |
| 7 | 880 | 18.1 |
| 8 | 175 | 3.6 |
| Total | 4873 | 100.0 |

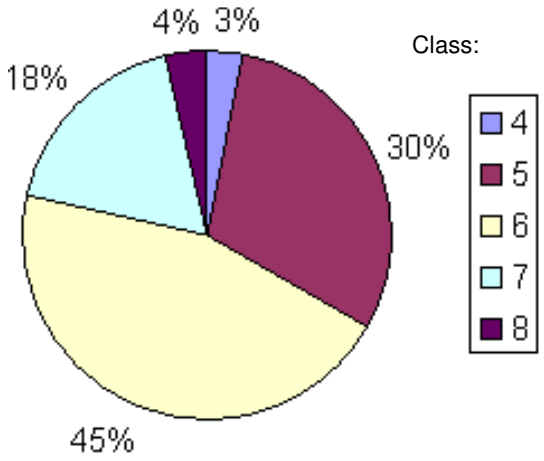


*Figure C.4: Class frequencies for the wine quality (white) class variable (quality)*

*Table C.6 Descriptive statistics for the Wine quality (white) dataset variables*

| Variable | Minimum | Maximum | Mean | Standard Deviation | Coeff of variation (CV) |
|---|---|---|---|---|---|
| FixedAcidity | 3.8 | 14.2 | 6.9 | 0.8 | 0.1 |
| VolatileAcidity | 0.1 | 1.1 | 0.3 | 0.1 | 0.4 |
| CitricAcid | 0.0 | 1.7 | 0.3 | 0.1 | 0.4 |
| ResidualSugar | 0.6 | 65.8 | 6.4 | 5.1 | 0.8 |
| Chlorides | 0.0 | 0.3 | 0.0 | 0.0 | 0.5 |
| FreeSulfurDioxide | 2.0 | 289.0 | 35.3 | 17.0 | 0.5 |
| TotalSulfurDioxide | 9.0 | 440.0 | 138.4 | 42.5 | 0.3 |
| Density | 1.0 | 1.0 | 1.0 | 0.0 | 0.0 |
| pH | 2.7 | 3.8 | 3.2 | 0.2 | 0.0 |
| Sulphates | 0.2 | 1.1 | 0.5 | 0.1 | 0.2 |
| Alcohol | 8.0 | 14.2 | 10.5 | 1.2 | 0.1 |

## C.5 Mushroom dataset

Table C.7 gives the descriptive statistics for the mushroom dataset. The variables for this dataset are all qualitative nominal.

*Table C.8 Descriptive statistics for the mushroom dataset variables*

| Variable | Level description | Level name | Frequ-ency% | Variable | Level description | Level name | Frequ-ency% |
|---|---|---|---|---|---|---|---|
| CapShape | 6 levels | FLAT | 39.1 | StalkRoot | 5 levels | EQUAL | 16.3 |
| | | CONVEX | 45.1 | | | BULBOUS | 45.2 |
| | | All other | 15.8 | | | UNKNOWN | 29.5 |
| CapSurface | 4 levels | GROOVES | 0.05 | | | All other | 9.0 |
| | | SMOOTH | 31.9 | StalkSfAbvRing | 4 levels | SILKY | 28.3 |
| | | FIBROUS | 29.2 | | | SMOOTH | 63.2 |
| | | SCALY | 38.8 | | | All other | 8.5 |
| CapColor | 10 levels | WHITE | 12.4 | StalkSfBlRing | 4 levels | SILKY | 27.4 |
| | | RED | 17.8 | | | SMOOTH | 60.3 |
| | | YELLOW | 12.7 | | | All other | 12.3 |
| | | BROWN | 27.6 | StalkClAbvRing | 9 levels | WHITE | 56.4 |
| | | GRAY | 24.9 | | | PINK | 22.2 |
| | | All other | 4.6 | | | All other | 21.4 |
| Bruises? | 2 levels | NO | 59.9 | StalkClBlRing | 9 levels | WHITE | 55.1 |
| | | BRUISES | 40.1 | | | PINK | 22.2 |
| Odor | 9 levels | FOUL | 25.7 | | | All other | 22.6 |
| | | NONE | 45.2 | VeilType | 1 level | PARTIAL | 100.0 |
| | | All other | 29.1 | VeilColor | 4 levels | WHITE | 97.6 |
| GillAttach | 2 levels | FREE | 97.4 | | | All other | 2.4 |
| | | ATTACHED | 2.6 | RingNumber | 3 levels | ONE | 92.3 |
| GillSpace | 2 levels | CROWDED | 18.9 | | | All other | 7.7 |
| | | CLOSE | 81.1 | RingType | 5 levels | LARGE | 15.4 |
| GillSize | 2 levels | NARROW | 30.1 | | | PENDANT | 47.1 |
| | | BROAD | 69.9 | | | EVANESCENT | 36.3 |
| GillColor | 12 levels | WHITE | 14.6 | | | All other | 1.1 |
| | | PINK | 18.5 | SporePrintColor | 9 levels | BLACK | 23.8 |
| | | BUFF | 20.5 | | | WHITE | 28.8 |
| | | BROWN | 13.2 | | | CHOCOLATE | 19.4 |
| | | All other | 33.1 | | | BROWN | 24.9 |
| | | GRAY | 8.9 | | | All other | 3.1 |
| StalkShape | 2 levels | ENLARGING | 42.2 | Population | 6 levels | SOLITARY | 20.3 |
| | | TAPERING | 57.8 | | | SEVERAL | 48.3 |
| Habitat | 7 levels | PATHS | 13.6 | | | SCATTERED | 16.3 |
| | | LEAVES | 10.2 | | | All other | 15.0 |
| | | GRASSES | 28.6 | **Class** | **2 levels** | **EDIBLE** | **53.3** |
| | | WOODS | 37.5 | | | **POISONOUS** | **46.7** |
| | | All other | 10.1 | | | | |

# Appendix D

# Correlation measurements for feature selection

The details of feature selection discussed in chapters 5 and 7 are presented in this appendix. Tables D.1 to D.4 show the class-feature correlations and the number of features selected by the t-test and the probes using Pearson's r and Kendall's tau measures of correlation for the forest cover type dataset.

*Table D.1: Feature selection for Forest cover type*

| Sample size for correlation measurement | Selection criteria (Number of selected features) | Top 10 features | | | | |
|---|---|---|---|---|---|---|
| | | Feature | Mean $Corr_{cf}$ | StDev | 95% CI of mean | |
| | | | | | Low | High |
| 100 | **Pearson's r** t-test (3) | WildernessArea4 | 0.2 | 0.06 | 0.16 | 0.24 |
| | | SoilType38 | 0.14 | 0.04 | 0.12 | 0.16 |
| | | Elevation | 0.14 | 0.04 | 0.12 | 0.16 |
| 500 and 1000 | Pearson's r t-test (6) (WildernessArea1 is selected for sample size 500, SoilType10 is selected for size 1000) | WildernessArea4 | 0.22 | 0.02 | 0.21 | 0.23 |
| | | SoilType12 | 0.16 | 0.02 | 0.15 | 0.17 |
| | | SoilType22 | 0.14 | 0.03 | 0.12 | 0.16 |
| | | Elevation | 0.13 | 0.02 | 0.12 | 0.14 |
| | | WildernessArea1 | 0.12 | 0.01 | 0.11 | 0.13 |
| | | SoilType38 | 0.12 | 0.02 | 0.11 | 0.13 |
| 100 | **Kendall's tau:** t-test (20) Uniform probe (26) Uniform binary probe (21) Gaussian probe (31) | WildernessArea4 | 0.58 | 0.15 | 0.49 | 0.67 |
| | | SoilType12 | 0.51 | 0.19 | 0.39 | 0.63 |
| | | SoilType38 | 0.44 | 0.1 | 0.38 | 0.50 |
| | | SoilType22 | 0.43 | 0.17 | 0.32 | 0.54 |
| | | SoilType10 | 0.4 | 0.13 | 0.32 | 0.48 |
| | | SoilType39 | 0.38 | 0.17 | 0.27 | 0.49 |
| | | SoilType4 | 0.35 | 0.2 | 0.23 | 0.47 |
| | | SoilType23 | 0.35 | 0.15 | 0.26 | 0.44 |
| | | SoilType11 | 0.32 | 0.16 | 0.22 | 0.42 |
| | | SoilType30 | 0.31 | 0.19 | 0.19 | 0.43 |
| 500 | **Kendall's tau** t-test (35) Uniform probe (47) Uniform binary probe (44) Gaussian probe (47) | WildernessArea4 | 0.81 | 0.03 | 0.79 | 0.83 |
| | | SoilType12 | 0.72 | 0.08 | 0.67 | 0.77 |
| | | SoilType38 | 0.6 | 0.08 | 0.55 | 0.65 |
| | | SoilType39 | 0.58 | 0.09 | 0.52 | 0.64 |
| | | SoilType2 | 0.58 | 0.15 | 0.49 | 0.67 |
| | | SoilType22 | 0.57 | 0.1 | 0.51 | 0.63 |
| | | SoilType4 | 0.57 | 0.12 | 0.50 | 0.64 |
| | | SoilType6 | 0.56 | 0.11 | 0.49 | 0.63 |
| | | SoilType13 | 0.56 | 0.11 | 0.49 | 0.63 |
| | | SoilType40 | 0.48 | 0.11 | 0.41 | 0.55 |
| 1000 | **Kendall's tau:** t-test (38) Uniform probe (48) Uniform binary probe (47) Gaussian probe (49) | WildernessArea4 | 0.86 | 0.02 | 0.85 | 0.87 |
| | | SoilType12 | 0.7 | 0.07 | 0.66 | 0.74 |
| | | SoilType1 | 0.69 | 0.05 | 0.66 | 0.72 |
| | | SoilType38 | 0.68 | 0.08 | 0.63 | 0.73 |
| | | SoilType39 | 0.68 | 0.08 | 0.63 | 0.73 |
| | | SoilType2 | 0.64 | 0.1 | 0.58 | 0.70 |
| | | SoilType4 | 0.64 | 0.05 | 0.61 | 0.67 |
| | | SoilType6 | 0.6 | 0.1 | 0.54 | 0.66 |
| | | SoilType22 | 0.59 | 0.1 | 0.53 | 0.65 |
| | | SoilType10 | 0.58 | 0.05 | 0.55 | 0.61 |

*Table D.2: Feature selection for forest cover type using Kendall's tau and a Gaussian probe*

| Rank | Feature | Kendall's tau | | Feature 95% CI | | Gaussian probe 95% CI | | Select |
|---|---|---|---|---|---|---|---|---|
| | | Mean | Stdev | Low | High | Low | High | |
| 1 | WildernessArea4 | 0.86 | 0.02 | 0.84 | 0.87 | 0.02 | 0.05 | yes |
| 2 | SoilType12 | 0.70 | 0.07 | 0.66 | 0.75 | 0.02 | 0.05 | yes |
| 3 | SoilType1 | 0.69 | 0.05 | 0.65 | 0.72 | 0.02 | 0.05 | yes |
| 4 | SoilType38 | 0.68 | 0.08 | 0.63 | 0.73 | 0.02 | 0.05 | yes |
| 5 | SoilType39 | 0.68 | 0.08 | 0.62 | 0.73 | 0.02 | 0.05 | yes |
| 6 | SoilType2 | 0.64 | 0.10 | 0.58 | 0.70 | 0.02 | 0.05 | yes |
| 7 | SoilType4 | 0.64 | 0.05 | 0.61 | 0.67 | 0.02 | 0.05 | yes |
| 8 | SoilType6 | 0.60 | 0.10 | 0.54 | 0.67 | 0.02 | 0.05 | yes |
| 9 | SoilType22 | 0.59 | 0.10 | 0.53 | 0.65 | 0.02 | 0.05 | yes |
| 10 | SoilType10 | 0.58 | 0.05 | 0.55 | 0.61 | 0.02 | 0.05 | yes |
| 11 | SoilType3 | 0.55 | 0.10 | 0.48 | 0.61 | 0.02 | 0.05 | yes |
| 12 | SoilType40 | 0.55 | 0.10 | 0.49 | 0.61 | 0.02 | 0.05 | yes |
| 13 | SoilType13 | 0.53 | 0.10 | 0.47 | 0.59 | 0.02 | 0.05 | yes |
| 14 | SoilType11 | 0.48 | 0.08 | 0.43 | 0.52 | 0.02 | 0.05 | yes |
| 15 | SoilType35 | 0.44 | 0.09 | 0.39 | 0.50 | 0.02 | 0.05 | yes |
| 16 | SoilType18 | 0.44 | 0.17 | 0.34 | 0.54 | 0.02 | 0.05 | yes |
| 17 | SoilType17 | 0.43 | 0.16 | 0.34 | 0.53 | 0.02 | 0.05 | yes |
| 18 | SoilType26 | 0.43 | 0.16 | 0.33 | 0.53 | 0.02 | 0.05 | yes |
| 19 | SoilType34 | 0.40 | 0.18 | 0.29 | 0.51 | 0.02 | 0.05 | yes |
| 20 | SoilType23 | 0.40 | 0.04 | 0.37 | 0.43 | 0.02 | 0.05 | yes |
| 21 | WildernessArea2 | 0.39 | 0.12 | 0.31 | 0.47 | 0.02 | 0.05 | yes |
| 22 | SoilType5 | 0.36 | 0.22 | 0.22 | 0.50 | 0.02 | 0.05 | yes |
| 23 | SoilType19 | 0.35 | 0.17 | 0.25 | 0.46 | 0.02 | 0.05 | yes |
| 24 | SoilType30 | 0.34 | 0.10 | 0.28 | 0.40 | 0.02 | 0.05 | yes |
| 25 | SoilType16 | 0.33 | 0.13 | 0.25 | 0.41 | 0.02 | 0.05 | yes |
| 26 | SoilType21 | 0.32 | 0.20 | 0.20 | 0.44 | 0.02 | 0.05 | yes |
| 27 | SoilType29 | 0.30 | 0.04 | 0.27 | 0.32 | 0.02 | 0.05 | yes |
| 28 | WildernessArea1 | 0.28 | 0.03 | 0.27 | 0.30 | 0.02 | 0.05 | yes |
| 29 | SoilType9 | 0.28 | 0.16 | 0.19 | 0.38 | 0.02 | 0.05 | yes |
| 30 | Elevation | 0.28 | 0.01 | 0.27 | 0.29 | 0.02 | 0.05 | yes |
| 31 | SoilType24 | 0.26 | 0.09 | 0.20 | 0.32 | 0.02 | 0.05 | yes |
| 32 | SoilType14 | 0.23 | 0.22 | 0.10 | 0.37 | 0.02 | 0.05 | yes |
| 33 | SoilType31 | 0.22 | 0.08 | 0.17 | 0.27 | 0.02 | 0.05 | yes |
| 34 | SoilType28 | 0.21 | 0.15 | 0.12 | 0.31 | 0.02 | 0.05 | yes |
| 35 | SoilType32 | 0.21 | 0.02 | 0.19 | 0.22 | 0.02 | 0.05 | yes |
| 36 | SoilType33 | 0.18 | 0.04 | 0.16 | 0.21 | 0.02 | 0.05 | yes |
| 37 | SoilType8 | 0.18 | 0.16 | 0.08 | 0.27 | 0.02 | 0.05 | yes |
| 38 | SoilType20 | 0.16 | 0.03 | 0.14 | 0.18 | 0.02 | 0.05 | yes |
| 39 | HorizDistToRoad | 0.16 | 0.01 | 0.15 | 0.17 | 0.02 | 0.05 | yes |
| 40 | HorizDistToFire | 0.16 | 0.01 | 0.15 | 0.16 | 0.02 | 0.05 | yes |
| 41 | SoilType27 | 0.15 | 0.15 | 0.05 | 0.24 | 0.02 | 0.05 | yes |
| 42 | Slope | 0.12 | 0.02 | 0.11 | 0.14 | 0.02 | 0.05 | yes |
| 43 | HillShade9am | 0.08 | 0.02 | 0.07 | 0.10 | 0.02 | 0.05 | yes |
| 44 | VertDistToHydro | 0.07 | 0.02 | 0.06 | 0.08 | 0.02 | 0.05 | yes |
| 45 | HorizDistToHydro | 0.07 | 0.02 | 0.06 | 0.08 | 0.02 | 0.05 | yes |
| 46 | WildernessArea3 | 0.07 | 0.03 | 0.05 | 0.09 | 0.02 | 0.05 | yes |
| 47 | HillShadeNoon | 0.07 | 0.02 | 0.06 | 0.08 | 0.02 | 0.05 | yes |
| 48 | Aspect | 0.05 | 0.02 | 0.03 | 0.06 | 0.02 | 0.05 | yes |
| 49 | HillShade3pm | 0.04 | 0.02 | 0.03 | 0.06 | 0.02 | 0.05 | yes |
| **50** | **Probe1GaussCont** | **0.04** | **0.02** | **0.02** | **0.05** | **0.02** | **0.05** | **no** |

Tables D.4 and D.5 show the class-feature correlations using Pearson's r, Kendall's tau and SU coefficient, and the number of features selected by the t-test, probes and decision rule-based algorithm for the KDDCup 1999 dataset.

*Table D.3 Features selected by the decision rule-based search algorithm for different inputs*

| Input feature set selected by: | Number of selected features | | Top 10 features for all methods | |
|---|---|---|---|---|
| | | | Feature | mean $corr_{cf}$ |
| No pre-selection (54 features + 3 probes) | 42 | | WildernessArea4 | 0.855 |
| | | | SoilType2 | 0.642 |
| Gaussian probe (49 features) | 41 | | SoilType40 | 0.547 |
| | | | SoilType38 | 0.676 |
| Uniform probe (48 features) | 41 | | SoilType4 | 0.638 |
| | | | SoilType1 | 0.686 |
| Uniform binary probe (47 features) | 41 | | SoilType3 | 0.548 |
| | | | SoilType6 | 0.603 |
| t-test for means (36 features) | 36 | | SoilType13 | 0.527 |
| | | | SoilType39 | 0.676 |

*Table D.4: Feature selection for KDD Cup 1999*

| Sample size for correlation measurement | Selection criteria (Number of selected features) | Top 10 features | | | | |
|---|---|---|---|---|---|---|
| | | | | | 95% CI of mean | |
| | | Feature | Mean $Corr_{cf}$ | StDev | Low | High |
| 1000 | Pearson's r:<br><br>t-test (21)<br>Uniform probe (32)<br>Uniform binary probe (31)<br>Gaussian probe (31) | SameSrvRate | 0.53 | 0.02 | 0.52 | 0.54 |
| | | SerrorRate | 0.51 | 0.02 | 0.50 | 0.52 |
| | | DstHostSerrorRate | 0.51 | 0.02 | 0.50 | 0.52 |
| | | Counted | 0.51 | 0.02 | 0.50 | 0.52 |
| | | SrvSerrorRate | 0.50 | 0.02 | 0.49 | 0.51 |
| | | DstHostSrvSerrorRate | 0.50 | 0.02 | 0.49 | 0.51 |
| | | Flag | 0.43 | 0.02 | 0.42 | 0.44 |
| | | DstHostRerrorRate | 0.36 | 0.03 | 0.34 | 0.38 |
| | | SrvRerrorRate | 0.35 | 0.03 | 0.33 | 0.37 |
| | | RerrorRate | 0.34 | 0.03 | 0.32 | 0.36 |
| 500 | Kendall's tau:<br><br>t-test (34)<br>Uniform probe (36)<br>Uniform binary probe (36)<br>Gaussian probe (36) | SrvSerrorRate | 0.90 | 0.02 | 0.89 | 0.91 |
| | | SerrorRate | 0.87 | 0.02 | 0.86 | 0.88 |
| | | NumCompromised | 0.85 | 0.03 | 0.83 | 0.87 |
| | | DstHostSrvSerrorRate | 0.83 | 0.04 | 0.81 | 0.85 |
| | | WrongFragment | 0.81 | 0.04 | 0.78 | 0.84 |
| | | DstHostSerrorRate | 0.81 | 0.02 | 0.80 | 0.82 |
| | | SrvRerrorRate | 0.80 | 0.04 | 0.78 | 0.82 |
| | | Hot | 0.78 | 0.04 | 0.76 | 0.80 |
| | | DstHostSrvRerrorRate | 0.76 | 0.05 | 0.73 | 0.79 |
| | | RerrorRate | 0.76 | 0.05 | 0.73 | 0.79 |
| 1000 | Kendall's tau:<br><br>t-test (30)<br>Uniform probe (36)<br>Uniform binary probe (35)<br>Gaussian probe (36) | SerrorRate | 0.92 | 0.01 | 0.91 | 0.93 |
| | | NumCompromised | 0.92 | 0.03 | 0.90 | 0.94 |
| | | SrvSerrorRate | 0.91 | 0.01 | 0.90 | 0.92 |
| | | WrongFragment | 0.9 | 0.01 | 0.89 | 0.91 |
| | | DstHostSrvSerrorRate | 0.85 | 0.01 | 0.84 | 0.86 |
| | | DstHostSrvRerrorRate | 0.85 | 0.01 | 0.84 | 0.86 |
| | | SrvRerrorRate | 0.85 | 0.02 | 0.84 | 0.86 |
| | | Hot | 0.84 | 0.03 | 0.82 | 0.86 |
| | | DstHostSerrorRate | 0.84 | 0.02 | 0.83 | 0.85 |
| | | RerrorRate | 0.82 | 0.03 | 0.80 | 0.84 |

*Table D.5: Feature selection for KDD Cup1999 using Kendall's tau and the Gaussian probe*

| Rank | Feature | Mean | StDev | Feature 95% CI | | Gauss probe 95% CI | | Select |
|---|---|---|---|---|---|---|---|---|
| | | | | Low | High | Low | High | |
| 1 | SerrorRate | 0.92 | 0.01 | 0.91 | 0.92 | 0.02 | 0.04 | yes |
| 2 | NumCompromised | 0.92 | 0.03 | 0.90 | 0.93 | 0.02 | 0.04 | yes |
| 3 | SrvSerrorRate | 0.91 | 0.01 | 0.91 | 0.92 | 0.02 | 0.04 | yes |
| 4 | WrongFragment | 0.90 | 0.01 | 0.89 | 0.91 | 0.02 | 0.04 | yes |
| 5 | DstHostSrvSerrorRate | 0.85 | 0.01 | 0.85 | 0.86 | 0.02 | 0.04 | yes |
| 6 | DstHostSrvRerrorRate | 0.85 | 0.01 | 0.84 | 0.85 | 0.02 | 0.04 | yes |
| 7 | SrvRerrorRate | 0.85 | 0.02 | 0.83 | 0.86 | 0.02 | 0.04 | yes |
| 8 | Hot | 0.84 | 0.03 | 0.83 | 0.86 | 0.02 | 0.04 | yes |
| 9 | DstHostSerrorRate | 0.84 | 0.02 | 0.82 | 0.85 | 0.02 | 0.04 | yes |
| 10 | RerrorRate | 0.82 | 0.03 | 0.80 | 0.84 | 0.02 | 0.04 | yes |
| 11 | SameSrvRate | 0.82 | 0.01 | 0.81 | 0.83 | 0.02 | 0.04 | yes |
| 12 | DstHostRerrorRate | 0.80 | 0.03 | 0.79 | 0.82 | 0.02 | 0.04 | yes |
| 13 | DiffSrvRate | 0.73 | 0.02 | 0.71 | 0.74 | 0.02 | 0.04 | yes |
| 14 | NumRoot | 0.68 | 0.10 | 0.62 | 0.74 | 0.02 | 0.04 | yes |
| 15 | Counted | 0.63 | 0.01 | 0.62 | 0.64 | 0.02 | 0.04 | yes |
| 16 | DstBytes | 0.58 | 0.06 | 0.55 | 0.62 | 0.02 | 0.04 | yes |
| 17 | SrcBytes | 0.49 | 0.05 | 0.46 | 0.52 | 0.02 | 0.04 | yes |
| 18 | SrvDiffHostRate | 0.46 | 0.08 | 0.41 | 0.50 | 0.02 | 0.04 | yes |
| 19 | DstHostSrvDiffHostRate | 0.44 | 0.05 | 0.41 | 0.47 | 0.02 | 0.04 | yes |
| 20 | Flag | 0.43 | 0.02 | 0.41 | 0.44 | 0.02 | 0.04 | yes |
| 21 | SrvCount | 0.42 | 0.02 | 0.41 | 0.44 | 0.02 | 0.04 | yes |
| 22 | DstHostCount | 0.37 | 0.03 | 0.35 | 0.39 | 0.02 | 0.04 | yes |
| 23 | DstHostSrvCount | 0.31 | 0.04 | 0.29 | 0.34 | 0.02 | 0.04 | yes |
| 24 | NumFailedLogins | 0.30 | 0.23 | 0.16 | 0.44 | 0.02 | 0.04 | yes |
| 25 | NumFileCreations | 0.30 | 0.08 | 0.25 | 0.35 | 0.02 | 0.04 | yes |
| 26 | DstHostSameSrcPortRate | 0.28 | 0.05 | 0.25 | 0.31 | 0.02 | 0.04 | yes |
| 27 | Duration | 0.25 | 0.02 | 0.24 | 0.27 | 0.02 | 0.04 | yes |
| 28 | Service | 0.24 | 0.01 | 0.23 | 0.24 | 0.02 | 0.04 | yes |
| 29 | DstHostSameSrvRate | 0.22 | 0.04 | 0.20 | 0.25 | 0.02 | 0.04 | yes |
| 30 | NumShells | 0.20 | 0.16 | 0.11 | 0.30 | 0.02 | 0.04 | yes |
| 31 | NumAccessFiles | 0.18 | 0.20 | 0.06 | 0.30 | 0.02 | 0.04 | yes |
| 32 | ProtocolType | 0.15 | 0.02 | 0.14 | 0.16 | 0.02 | 0.04 | yes |
| 33 | DstHostDiffSrvRate | 0.14 | 0.04 | 0.12 | 0.17 | 0.02 | 0.04 | yes |
| 34 | RootShell | 0.11 | 0.15 | 0.02 | 0.20 | 0.02 | 0.04 | no |
| 35 | LoggedIn | 0.08 | 0.01 | 0.08 | 0.09 | 0.02 | 0.04 | yes |
| 36 | IsGuestLogin | 0.04 | 0.01 | 0.03 | 0.05 | 0.02 | 0.04 | yes |
| 37 | Urgent | 0.03 | 0.11 | -0.03 | 0.10 | 0.02 | 0.04 | no |
| **38** | **Probe1GaussCont** | **0.03** | **0.02** | **0.02** | **0.04** | **0.02** | **0.04** | **no** |

Tables D.7 and D.9 show the class-feature correlations using Pearson's r, Kendall's tau and the SU coefficient, and the number of features selected by the t-test, probes and decision rule-based algorithm for the abalone3C and mushroom datasets. Table D.8 shows the feature-feature correlations for abalone3C.

*Table D.6: KDD Cup 1999 feature selection by decision rule*

| Input feature set selected by: | Number of selected features | | Top 10 for no-preselection (32 features selected) | |
|---|---|---|---|---|
| | | | Feature | mean $corr_{cf}$ |
| No pre-selection (41 features + 3 probes) | 32 | | SerrorRate | 0.92 |
| | | | DstHostRerrorRate | 0.81 |
| Gaussian probe (36 features) | 34 | | NumRoot | 0.68 |
| | | | WrongFragment | 0.90 |
| Uniform probe (36 features) | 34 | | Flag | 0.43 |
| | | | NumFailedLogins | 0.30 |
| Uniform binary probe (35 features) | 34 | | DstHostSerrorRate | 0.84 |
| | | | DstHostSrvCount | 0.31 |
| t-test for means (30 features) | 30 | | SrvCount | 0.42 |
| | | | DstHostCount | 0.37 |

*Table D.7: Feature selection for Abalone using Pearson's r and Kendall's tau*

| Sample size | Selection criteria (Number of selected features) | Selected features | | | | |
|---|---|---|---|---|---|---|
| | | Feature | Mean $Corr_{CF}$ | StDev | 95% CI of mean | |
| | | | | | Low | High |
| 500 and 1000 | Pearson's r:<br><br>t-test (5)<br>probes do not eliminate any features | Diameter | 0.41 | 0.02 | 0.40 | 0.42 |
| | | ShellWeight | 0.4 | 0.02 | 0.39 | 0.41 |
| | | WholeWeight | 0.38 | 0.02 | 0.37 | 0.39 |
| | | VisceraWeight | 0.38 | 0.02 | 0.37 | 0.39 |
| | | ShuckledWeight | 0.34 | 0.02 | 0.33 | 0.35 |
| 500 | Kendall's tau:<br>t-test (6)<br>probes do not eliminate any features | Height | 0.52 | 0.03 | 0.50 | 0.54 |
| | | ShellWeight | 0.53 | 0.03 | 0.51 | 0.55 |
| | | Diameter | 0.5 | 0.03 | 0.48 | 0.52 |
| | | VisceraWeight | 0.49 | 0.03 | 0.47 | 0.51 |
| | | ShuckledWeight | 0.45 | 0.03 | 0.43 | 0.47 |
| | | WholeWeight | 0.5 | 0.03 | 0.48 | 0.52 |
| 1000 | Kendall's tau:<br>t-test (7)<br><br>probes do not eliminate any features | ShellWeight | 0.52 | 0.02 | 0.51 | 0.53 |
| | | Height | 0.51 | 0.02 | 0.50 | 0.52 |
| | | Diameter | 0.5 | 0.02 | 0.49 | 0.51 |
| | | WholeWeight | 0.49 | 0.02 | 0.48 | 0.50 |
| | | VisceraWeight | 0.49 | 0.02 | 0.48 | 0.50 |
| | | ShuckledWeight | 0.45 | 0.02 | 0.44 | 0.46 |
| | | Length | 0.17 | 0.01 | 0.16 | 0.18 |
| 1000 | Decision rule (3) | ShellWeight | 0.53 | 0.03 | 0.51 | 0.55 |
| | | Length | 0.17 | 0.01 | 0.16 | 0.18 |
| | | Gender | 0.12 | 0.01 | 0.11 | 0.13 |

*Table D.8: Abalone3C feature-feature correlations*

| Feature1 | Feature2 | $corr_{ff}$ | | Feature1 | Feature2 | $corr_{ff}$ |
|---|---|---|---|---|---|---|
| Length | Diameter | 0.92 | | Height | ShellWeight | 0.79 |
| Length | Height | 0.75 | | WholeWeight | ShuckledWeight | 0.88 |
| Length | WholeWeight | 0.88 | | WholeWeight | VisceraWeight | 0.87 |
| Length | ShuckledWeight | 0.84 | | WholeWeight | ShellWeight | 0.86 |
| Length | VisceraWeight | 0.83 | | ShuckledWeight | VisceraWeight | 0.80 |
| Length | ShellWeight | 0.83 | | ShuckledWeight | ShellWeight | 0.76 |
| Diameter | Height | 0.77 | | VisceraWeight | ShellWeight | 0.80 |
| Diameter | WholeWeight | 0.88 | | Length | Gender | 0.11 |
| Diameter | ShuckledWeight | 0.83 | | Diameter | Gender | 0.46 |
| Diameter | VisceraWeight | 0.83 | | Height | Gender | 0.47 |
| Diameter | ShellWeight | 0.85 | | WholeWeight | Gender | 0.48 |
| Height | WholeWeight | 0.78 | | ShuckledWeight | Gender | 0.46 |
| Height | ShuckledWeight | 0.72 | | VisceraWeight | Gender | 0.49 |
| Height | VisceraWeight | 0.76 | | ShellWeight | Gender | 0.47 |

*Table D9: Feature selection for mushroom using SU coefficients*

| Sample size for SU measurement | Selection criteria (Number of selected features) | Selected features or top 5 features | | | |
|---|---|---|---|---|---|
| | | Feature | Mean *SU* | StDev | 95% CI of mean |
| 500 | t-test (4) | Ordor | 0.55 | 0.03 | 0.02 |
| | | SporePrintColor | 0.3 | 0.02 | 0.01 |
| | | RingType | 0.23 | 0.01 | 0.01 |
| | | GillColor | 0.2 | 0.02 | 0.01 |
| 500 | Uniform probe (15)<br>Uniform binary probe (14)<br>Gaussian probe (21) | Ordor | 0.55 | 0.03 | 0.02 |
| | | SporePrintColor | 0.3 | 0.02 | 0.01 |
| | | StalkSfAbvRing | 0.28 | 0.03 | 0.02 |
| | | GillSize | 0.24 | 0.03 | 0.02 |
| | | StalkSfBlRing | 0.23 | 0.03 | 0.02 |
| 500 | Decision rule (14) | Ordor | 0.55 | 0.03 | 0.02 |
| | | SporePrintColor | 0.30 | 0.02 | 0.02 |
| | | StalkSfAbvRing | 0.28 | 0.03 | 0.02 |
| | | GillSize | 0.24 | 0.03 | 0.02 |
| | | StalkSfBlRing | 0.23 | 0.03 | 0.02 |
| | | RingType | 0.23 | 0.01 | 0.01 |
| | | GillColor | 0.20 | 0.02 | 0.01 |
| | | StalkClAbvRing | 0.18 | 0.02 | 0.01 |
| | | Bruises | 0.17 | 0.03 | 0.02 |
| | | StalkClBlRing | 0.15 | 0.02 | 0.01 |
| | | Population | 0.14 | 0.02 | 0.01 |
| | | GillSpace | 0.14 | 0.03 | 0.02 |
| | | habitat | 0.11 | 0.01 | 0.01 |
| | | StalkRoot | 0.10 | 0.01 | 0.01 |

# Appendix E

# Algorithm for breadth first generation of a search space

This appendix provides the details of the standard breadth-first search algorithm and the *BreadthFirstGenerate* algorithm which is based on the breadth first algorithm. The *BreadthFirstGenerate* algorithm was used for the generation of all possible tied predictions as discussed in section 6.4. The standard breadth-first search algorithm (Luger & Stubblefield, 1993) is given in figure E.1. The *BreadthFirstGenerate* algorithm is given in figure E.2.

Both algorithms use the lists OPEN, CLOSED and CHILDREN. The OPEN list holds the states that are still to be expanded. The CLOSED list holds all states that have been generated so far. The CHILDREN list is used to temporarily hold all the children (successor states) of the current state while the children are being generated. The major difference between the breadth-first-search algorithm and the BreadthFirstGenerate algorithm is that the breadth-first-search algorithm specifically searches for a goal state while the BreadthFirstGenerate algorithm simply generates all the possible states in the search space.

**Breadth-first-search**

```
1. OPEN = [start_state]
2. CLOSED = [ ]
3. while OPEN ≠ [ ]
   begin
        3.1  Remove leftmost state from OPEN, and call it X
        3.2  if X is the goal state
                  return X
        else
        3.2  generate children of X and put them on the CHILDREN list
        3.3 eliminate children of X on OPEN (prevent cycles)
        3.4  put X on CLOSED
        3.5 put all states on CHILDREN list on right end of OPEN
  end
```

*Figure E.1: Breadth-first search algorithm*

```
BreadthFirstGenerate( )
1. OPEN = [start_state]
2. CLOSED = [ ]
3. while OPEN ≠ [ ]
   begin
        3.1  Remove leftmost state from OPEN and  call it X
        3.2  generate children of X and put them on the CHILDREN list
        3.3  put X on CLOSED
        3.4 put all states on CHILDREN list on right end of OPEN
  end
```

*Figure E.2: BreadthFirstGenerate algorithm*

For the generation of all possible tied predictions, the predictions are assigned numbers 1,2,..,*k* corresponding to the *k* classes for the prediction task. The start state contains the first number (1). Each state {1,.., *j*} has the children *j*+1, *j*+2,..,*k*. When the *BreadthFirstGenerate* algorithm has finished executing, all the possible states (tied predictions) are available on the CLOSED list.

Given a search space represented by a search tree with a constant branching factor *B*, the number of states (paths) of length *L* generated by a search algorithm is given by (Luger & Stubblefield, 1993: pg 146)

$$States = B + B^2 + B^3 + ... + B^L \quad \text{(E.1)}$$

which reduces to:

$$States = B(B^L - 1)/(B - 1) \quad \text{(E.2)}$$

For the *BreadthFirstGenerate* algorithm, the branching factor for level 1 of the tree is *k* -1 and reduces by 1 for successive levels. The maximum path length is *k* so that

$$States = (k-1) + (k-2)^2 + ... + (k-(k-1))^k \quad \text{(E.3)}$$

which reduces to:

$$States = \sum_{j=1}^{k} (k-j)^j \quad \text{(E.4)}$$

# Appendix F

# Predictive performance of single OVA and pVn models

The detailed results for predictive accuracy and TPRATE values for the single *k*-class, OVA aggregate and pVn aggregate models using the 5NN and See5 algorithms are provided in this appendix. Each table shows the accuracy and class TPRATE values for 10 test samples, as well as the mean, 95% confidence interval of the mean, standard deviation and variance. The mean values for performance were discussed in chapters 7 and 8. The variance values were used for the F-tests discussed in chapter 8.

## F.1 5NN single 7-class and aggregate models for forest cover type

Tables F.1 to F.4 give the details of predictive accuracy and TRATE values for the 5NN single 7-class, OVA and pVn aggregate models forest cover type.

*Table F.1: Predictive performance of the 5NN single 7- class model for forest cover type*

| Test set | Accuracy on all classes | 5NN single model (equal class distribution) TPRATE% for class: | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 75.4 | 68 | 48 | 58 | 98 | 88 | 72 | 96 |
| 2 | 71.4 | 60 | 46 | 50 | 90 | 92 | 70 | 92 |
| 3 | 75.1 | 60 | 48 | 64 | 96 | 88 | 76 | 94 |
| 4 | 73.7 | 66 | 50 | 48 | 90 | 92 | 74 | 96 |
| 5 | 72.6 | 54 | 42 | 56 | 92 | 94 | 76 | 94 |
| 6 | 76.9 | 72 | 50 | 48 | 94 | 94 | 82 | 98 |
| 7 | 74.6 | 60 | 50 | 58 | 90 | 90 | 76 | 98 |
| 8 | 76 | 60 | 58 | 68 | 90 | 86 | 72 | 98 |
| 9 | 75.4 | 60 | 52 | 58 | 94 | 92 | 74 | 98 |
| 10 | 76 | 68 | 44 | 60 | 90 | 96 | 78 | 96 |
| **Mean** | **74.7** | **62.8** | **48.8** | **56.8** | **92.4** | **91.2** | **75.0** | **96.0** |
| **StDev** | **1.7** | **5.4** | **4.4** | **6.6** | **3.0** | **3.2** | **3.4** | **2.1** |
| **Variance** | **2.9** | **29.5** | **19.7** | **43.7** | **8.7** | **10.0** | **11.8** | **4.4** |
| **Mean & CI** | **74.7±1.0** | **62.8±3.4** | **48.8±2.8** | **56.8±4.1** | **92.4±1.8** | **91.2±2.0** | **75.0±2.1** | **96.0±1.3** |

*Table F.2: Predictive performance of the 5NN un-boosted OVA aggregate model for forest cover type*

| Test set | Accuracy on all classes | 5NN un-boosted OVA aggregate model. TPRATE% for class: | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 78.3 | 74 | 58 | 72 | 96 | 86 | 64 | 98 |
| 2 | 80.3 | 68 | 64 | 68 | 92 | 94 | 78 | 98 |
| 3 | 82.9 | 78 | 58 | 76 | 92 | 96 | 84 | 96 |
| 4 | 80.6 | 84 | 58 | 70 | 86 | 88 | 82 | 96 |
| 5 | 80.9 | 70 | 58 | 68 | 88 | 100 | 86 | 96 |
| 6 | 79.1 | 62 | 60 | 70 | 88 | 100 | 78 | 96 |
| 7 | 79.1 | 62 | 52 | 70 | 92 | 98 | 82 | 98 |
| 8 | 82.0 | 66 | 66 | 76 | 88 | 100 | 82 | 96 |
| 9 | 82.0 | 68 | 58 | 74 | 88 | 98 | 92 | 96 |
| 10 | 79.4 | 68 | 52 | 74 | 88 | 98 | 80 | 96 |
| **Mean** | **80.5** | **70.0** | **58.4** | **71.8** | **89.8** | **95.8** | **80.8** | **96.6** |
| **StDev** | **1.5** | **6.9** | **4.4** | **3.0** | **3.0** | **5.0** | **7.2** | **1.0** |
| **Variance** | **2.3** | **48.0** | **19.4** | **9.3** | **9.3** | **25.3** | **51.7** | **0.9** |
| **Mean&CI** | **80.5±0.9** | **70±4.3** | **58.4±2.7** | **71.8±1.9** | **89.8±1.9** | **95.8±3.1** | **80.8±4.5** | **96.6±0.6** |

*TableF.3: Predictive performance of the 5NN boosted OVA aggregate model for forest cover type*

| Test set | Accuracy on all classes | 5NN boosted OVA aggregate model. TPRATE% for class: | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 82.9 | 74 | 62 | 74 | 100 | 98 | 74 | 98 |
| 2 | 82.3 | 68 | 70 | 70 | 100 | 98 | 72 | 98 |
| 3 | 82.6 | 78 | 58 | 72 | 100 | 94 | 80 | 96 |
| 4 | 83.7 | 84 | 62 | 68 | 100 | 98 | 78 | 96 |
| 5 | 81.4 | 70 | 60 | 68 | 100 | 96 | 80 | 96 |
| 6 | 81.4 | 62 | 60 | 72 | 100 | 98 | 82 | 96 |
| 7 | 80.9 | 62 | 58 | 74 | 100 | 96 | 78 | 98 |
| 8 | 82.3 | 66 | 72 | 72 | 100 | 96 | 74 | 96 |
| 9 | 82.3 | 68 | 64 | 70 | 100 | 98 | 80 | 96 |
| 10 | 80.6 | 68 | 54 | 70 | 100 | 98 | 78 | 96 |
| **Mean** | **82.0** | **70.0** | **62.0** | **71.0** | **100.0** | **97.0** | **77.6** | **96.6** |
| **StDev** | **1.0** | **6.9** | **5.5** | **2.2** | **0.0** | **1.4** | **3.2** | **1.0** |
| **Variance** | **0.9** | **48.0** | **30.2** | **4.7** | **0.0** | **2.0** | **10.5** | **0.9** |
| **Mean &CI** | **82.0±0.6** | **70.0±4.3** | **62.0±3.4** | **71.0±1.3** | **100.0±0.0** | **97.0±0.9** | **77.6±2.0** | **96.6±0.6** |

*Table F.4: Predictive performance of the 5NN pVn aggregate model for forest cover type*

| Test set | Accuracy on all classes | 5NN pVn aggregate model. TPRATE% for class: | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 78.3 | 68 | 52 | 70 | 98 | 90 | 70 | 100 |
| 2 | 75.1 | 60 | 60 | 66 | 94 | 90 | 68 | 88 |
| 3 | 81.4 | 82 | 58 | 68 | 100 | 94 | 74 | 94 |
| 4 | 79.7 | 80 | 56 | 64 | 96 | 94 | 76 | 92 |
| 5 | 79.1 | 70 | 60 | 62 | 98 | 98 | 78 | 88 |
| 6 | 80.0 | 70 | 58 | 60 | 98 | 98 | 82 | 94 |
| 7 | 76.0 | 66 | 52 | 64 | 94 | 94 | 72 | 90 |
| 8 | 77.7 | 62 | 60 | 66 | 98 | 90 | 74 | 94 |
| 9 | 80.3 | 64 | 62 | 70 | 98 | 96 | 74 | 98 |
| 10 | 78.0 | 56 | 60 | 60 | 96 | 98 | 82 | 94 |
| **Mean** | **78.6** | **67.8** | **57.8** | **65.0** | **97.0** | **94.2** | **75.0** | **93.2** |
| **StDev** | **2.0** | **8.2** | **3.5** | **3.7** | **1.9** | **3.3** | **4.6** | **3.9** |
| **Variance** | **3.8** | **68.0** | **12.0** | **13.6** | **3.8** | **11.1** | **21.6** | **15.3** |
| **Mean&CI** | **78.6±1.2** | **67.8±5.1** | **57.8±2.1** | **65.0±2.3** | **97.0±1.2** | **94.2±2.1** | **75.0±2.9** | **93.2±2.4** |

## F.2 See5 single 7-class and aggregate models for forest cover type

Tables F.5 to F.8 give the details of predictive accuracy and TPRATE values for the See5 single 7-class, OVA and pVn aggregate models for the forest cover type dataset.

*Table F.5: Predictive performance of the See5 single 7-class model for forest cover type*

| **Test set** | Accuracy on **all classes** | See5 single model (equal class distribution). TPRATE% for class: | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| 1 | 77.1 | 56 | 60 | 68 | 100 | 92 | 70 | 94 |
| 2 | 76 | 68 | 58 | 60 | 96 | 86 | 70 | 94 |
| 3 | 78 | 52 | 66 | 68 | 98 | 86 | 80 | 96 |
| 4 | 76 | 52 | 62 | 64 | 96 | 86 | 80 | 92 |
| 5 | 77.1 | 66 | 62 | 54 | 94 | 92 | 80 | 92 |
| 6 | 78.9 | 58 | 74 | 58 | 98 | 90 | 78 | 96 |
| 7 | 73.7 | 54 | 58 | 54 | 96 | 82 | 74 | 98 |
| 8 | 76 | 56 | 66 | 56 | 96 | 82 | 78 | 98 |
| 9 | 78.9 | 58 | 66 | 64 | 98 | 82 | 88 | 96 |
| 10 | 77.4 | 54 | 66 | 62 | 96 | 84 | 80 | 100 |
| **Mean** | **76.91** | **57.40** | **63.80** | **60.80** | **96.80** | **86.20** | **77.80** | **95.60** |
| **StDev** | **1.57** | **5.50** | **4.85** | **5.27** | **1.69** | **3.94** | **5.37** | **2.63** |
| **Variance** | **2.47** | **30.27** | **23.51** | **27.73** | **2.84** | **15.51** | **28.84** | **6.93** |
| **Mean & CI** | **76.9±1.0** | **57.4±3.4** | **63.8±3.0** | **60.8±3.3** | **96.8±1.0** | **86.2±2.4** | **77.8±3.3** | **95.6±1.6** |

*Table F.6: Predictive performance of See5 un-boosted OVA aggregate model for forest cover type*

| Test sample | Accuracy on all classes | See5 un-boosted OVA aggregate model. TPRATE% for class: | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 74.9 | 64 | 44 | 68 | 84 | 96 | 80 | 88 |
| 2 | 75.7 | 62 | 52 | 66 | 88 | 94 | 78 | 90 |
| 3 | 75.1 | 60 | 50 | 60 | 92 | 90 | 78 | 96 |
| 4 | 73.7 | 60 | 40 | 62 | 88 | 96 | 80 | 90 |
| 5 | 74.3 | 60 | 44 | 64 | 86 | 98 | 78 | 90 |
| 6 | 77.1 | 68 | 58 | 60 | 86 | 98 | 80 | 90 |
| 7 | 74.6 | 58 | 50 | 66 | 88 | 94 | 72 | 94 |
| 8 | 75.1 | 54 | 52 | 66 | 82 | 96 | 80 | 100 |
| 9 | 76.9 | 64 | 50 | 66 | 86 | 92 | 82 | 98 |
| 10 | 75.4 | 56 | 58 | 62 | 86 | 90 | 84 | 92 |
| **Mean** | **75.3** | **60.6** | **49.8** | **64.0** | **86.6** | **94.4** | **79.2** | **92.8** |
| **StDev** | **1.1** | **4.1** | **5.8** | **2.8** | **2.7** | **3.0** | **3.2** | **4.0** |
| **Variance** | **1.1** | **16.9** | **34.2** | **8.0** | **7.2** | **8.7** | **10.0** | **16.2** |
| **Mean&CI** | **75.3±0.7** | **60.6±2.6** | **49.8±3.6** | **64.0±1.8** | **86.6±1.7** | **94.4±1.8** | **79.2±2.0** | **92.8±2.5** |

*Table F.7: Predictive performance of See5 boosted OVA aggregate model for forest cover type*

| **Test set** | Accuracy on all classes | See5 boosted OVA aggregate model. TPRATE% for class: | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| 1 | 79.4 | 70 | 70 | 72 | 96 | 82 | 70 | 96 |
| 2 | 80.3 | 68 | 66 | 70 | 94 | 92 | 74 | 98 |
| 3 | 80 | 60 | 66 | 66 | 100 | 90 | 78 | 100 |
| 4 | 77.7 | 62 | 66 | 62 | 94 | 86 | 78 | 96 |
| 5 | 78.9 | 60 | 70 | 60 | 96 | 92 | 78 | 96 |
| 6 | 80.3 | 70 | 76 | 54 | 96 | 90 | 78 | 98 |
| 7 | 78.9 | 62 | 74 | 60 | 96 | 90 | 72 | 98 |
| 8 | 78.6 | 66 | 70 | 60 | 92 | 90 | 76 | 96 |
| 9 | 80.6 | 72 | 66 | 62 | 94 | 90 | 80 | 100 |
| 10 | 79.1 | 60 | 74 | 66 | 96 | 82 | 76 | 100 |
| **Mean** | **79.38** | **65.00** | **69.80** | **63.20** | **95.40** | **88.40** | **76.00** | **97.80** |
| **StDev** | **0.92** | **4.74** | **3.82** | **5.35** | **2.12** | **3.75** | **3.13** | **1.75** |
| **Variance** | **0.84** | **22.44** | **14.62** | **28.62** | **4.49** | **14.04** | **9.78** | **3.07** |
| **Mean & CI** | **79.4±0.6** | **65.0±2.9** | **69.8±2.4** | **63.2±3.3** | **95.4±1.3** | **88.4±2.3** | **76.0±1.9** | **97.8±1.1** |

*Table F.8: Predictive performance of the See5 pVn aggregate model for forest cover type*

| Test set | Accuracy on all classes | See5 pVn aggregate model. TPRATE% for class: | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 78 | 72 | 56 | 72 | 94 | 84 | 78 | 90 |
| 2 | 79.1 | 70 | 58 | 74 | 92 | 92 | 82 | 86 |
| 3 | 80.6 | 64 | 62 | 76 | 100 | 88 | 78 | 96 |
| 4 | 79.4 | 62 | 68 | 76 | 94 | 88 | 82 | 86 |
| 5 | 80 | 62 | 66 | 74 | 96 | 88 | 86 | 88 |
| 6 | 79.7 | 64 | 74 | 58 | 92 | 92 | 88 | 90 |
| 7 | 78.6 | 66 | 58 | 70 | 98 | 86 | 76 | 96 |
| 8 | 80.3 | 62 | 72 | 70 | 94 | 90 | 80 | 94 |
| 9 | 83.7 | 68 | 74 | 76 | 92 | 92 | 88 | 96 |
| 10 | 79.1 | 56 | 64 | 72 | 94 | 86 | 84 | 98 |
| **Mean** | **79.85** | **64.60** | **65.20** | **71.80** | **94.60** | **88.60** | **82.20** | **92.00** |
| **StDev** | **1.56** | **4.62** | **6.75** | **5.37** | **2.67** | **2.84** | **4.26** | **4.52** |
| **Variance** | **2.44** | **21.38** | **45.51** | **28.84** | **7.16** | **8.04** | **18.18** | **20.44** |
| **Mean& CI** | **79.9±1.0** | **64.6±2.9** | **65.2±4.2** | **71.8±3.3** | **94.6±1.7** | **88.6±1.8** | **82.2±2.6** | **92.0±2.8** |

## F.3 5NN single 5-class and aggregate models for KDD Cup 1999

Tables F.9 to F.12 give the details of predictive accuracy and TPRATE values for the 5NN single 5-class, OVA and pVn aggregate models KDD Cup 1999.

*Table F.9: Predictive performance of the 5NN single 5-class model for KDD Cup 1999*

| Test set | Accuracy on all classes | 5NN single model (equal class distribution for NORMAL, DOS, PROBE, R2L). TPRATE% for class: | | | | |
|---|---|---|---|---|---|---|
| | | **NORMAL** | **DOS** | **PROBE** | **R2L** | **U2R** |
| 1 | 69.7 | 81.4 | 80 | 95.7 | 60 | 31.4 |
| 2 | 72 | 87.1 | 72.9 | 97.1 | 71.4 | 31.4 |
| 3 | 65.7 | 87.1 | 51.4 | 98.6 | 60 | 31.4 |
| 4 | 71.1 | 94.3 | 61.4 | 94.3 | 72.9 | 32.9 |
| 5 | 68.3 | 81.4 | 62.9 | 92.9 | 72.9 | 31.4 |
| 6 | 66.3 | 85.7 | 60 | 94.3 | 60 | 31.4 |
| 7 | 69.7 | 87.1 | 71.4 | 94.3 | 64.3 | 31.4 |
| 8 | 66.3 | 81.4 | 67.1 | 94.3 | 57.1 | 31.4 |
| 9 | 69.7 | 82.8 | 71.4 | 98.6 | 64.3 | 31.4 |
| 10 | 66.6 | 75.7 | 64.3 | 97.1 | 64.3 | 31.4 |
| **Mean** | **68.54** | **84.40** | **66.28** | **95.72** | **64.72** | **31.55** |
| **StDev** | **2.22** | **5.02** | **8.06** | **2.01** | **5.81** | **0.47** |
| **Variance** | **4.94** | **25.20** | **65.02** | **4.05** | **33.76** | **0.22** |
| **Mean & CI** | **68.5 ± 1.4** | **84.4 ± 3.1** | **66.3 ± 5.0** | **95.7 ± 1.2** | **64.7 ± 3.6** | **31.6 ± 0.3** |

*Table F.10: Predictive performance of the 5NN OVA un-boosted aggregate model for KDD Cup 1999*

| Test set | Accuracy on all classes | 5NN un-boosted OVA aggregate model. TPRATE % for class: | | | | |
|---|---|---|---|---|---|---|
| | | NORMAL | DOS | PROBE | R2L | U2R |
| 1 | 73.7 | 90 | 81.4 | 94.3 | 61.4 | 41.4 |
| 2 | 73.4 | 92.9 | 68.6 | 95.7 | 67.1 | 42.9 |
| 3 | 72.3 | 98.6 | 58.6 | 98.6 | 62.9 | 42.9 |
| 4 | 73.1 | 97.1 | 61.4 | 94.3 | 71.4 | 41.4 |
| 5 | 71.7 | 85.7 | 64.3 | 92.9 | 72.9 | 42.9 |
| 6 | 69.4 | 91.4 | 57.1 | 94.3 | 61.4 | 42.9 |
| 7 | 73.7 | 94.3 | 68.6 | 95.7 | 67.1 | 42.9 |
| 8 | 69.1 | 87.1 | 65.7 | 94.3 | 55.7 | 42.9 |
| 9 | 74.3 | 98.6 | 71.4 | 97.1 | 61.4 | 42.9 |
| 10 | 72.9 | 91.4 | 62.9 | 94.3 | 72.9 | 42.9 |
| **Mean** | **72.4** | **92.7** | **66.0** | **95.2** | **65.4** | **42.6** |
| **StDev** | **1.8** | **4.5** | **7.1** | **1.7** | **5.8** | **0.6** |
| **Variance** | **3.2** | **20.3** | **49.7** | **2.8** | **33.6** | **0.4** |
| **CI of mean** | **1.1** | **2.8** | **4.4** | **1.0** | **3.6** | **0.4** |
| **Mean&CI** | **72.4±1.1** | **92.7±2.8** | **66.0±4.4** | **95.2±1.0** | **65.4±3.6** | **42.6±0.4** |

*Table F.11: Predictive performance of the 5NN OVA boosted aggregate model for KDD Cup 1999*

| Test set | Accuracy on all classes | 5NN boosted OVA aggregate model. TPRATE% for class: | | | | |
|---|---|---|---|---|---|---|
| | | NORMAL | DOS | PROBE | R2L | U2R |
| 1 | 73.7 | 90 | 82.9 | 94.3 | 58.6 | 42.9 |
| 2 | 73.4 | 94.3 | 68.6 | 95.7 | 65.7 | 42.9 |
| 3 | 70.0 | 98.6 | 52.9 | 98.6 | 57.1 | 42.9 |
| 4 | 72.3 | 97.1 | 61.4 | 94.3 | 65.7 | 42.9 |
| 5 | 70.9 | 85.7 | 64.3 | 92.9 | 74.3 | 37.1 |
| 6 | 68.0 | 90 | 58.6 | 94.3 | 57.1 | 40 |
| 7 | 71.4 | 94.3 | 70 | 95.7 | 58.6 | 38.6 |
| 8 | 68.3 | 85.7 | 67.1 | 94.3 | 55.7 | 38.6 |
| 9 | 72.3 | 98.6 | 71.4 | 98.6 | 54.3 | 38.6 |
| 10 | 70.0 | 90 | 62.9 | 95.7 | 61.4 | 40 |
| **Mean** | **71.0** | **92.4** | **66.0** | **95.4** | **60.9** | **40.5** |
| **StDev** | **2.0** | **4.9** | **8.2** | **1.9** | **6.1** | **2.3** |
| **Variance** | **3.9** | **23.7** | **66.5** | **3.5** | **37.3** | **5.1** |
| **CI of mean** | **1.2** | **3.0** | **5.1** | **1.2** | **3.8** | **1.4** |
| **Mean&CI** | **71.0±1.2** | **92.4±3.0** | **66.0±5.1** | **95.4±1.2** | **60.9±3.8** | **40.5±1.4** |

*Table F.12: Predictive performance of the 5NN pVn aggregate model for KDD Cup 1999*

| Test sample | Accuracy on all classes | 5NN pVn aggregate model. TPRATE% for class: | | | | |
|---|---|---|---|---|---|---|
| | | NORMAL | DOS | PROBE | R2L | U2R |
| 1 | 79.4 | 97.1 | 100 | 98.6 | 72.9 | 28.6 |
| 2 | 82.0 | 98.6 | 98.6 | 98.6 | 88.6 | 27.1 |
| 3 | 80.6 | 100 | 95.7 | 100 | 81.4 | 25.7 |
| 4 | 82.0 | 100 | 98.6 | 98.6 | 85.7 | 25.7 |
| 5 | 81.0 | 100 | 98.6 | 97.1 | 85.7 | 25.7 |
| 6 | 78.0 | 95.7 | 94.3 | 100 | 77.1 | 21.4 |
| 7 | 83.0 | 100 | 98.6 | 98.6 | 87.1 | 31.4 |
| 8 | 80.0 | 98.6 | 100 | 97.1 | 74.3 | 28.6 |
| 9 | 77.1 | 98.6 | 91.4 | 100 | 72.9 | 22.9 |
| 10 | 80.0 | 98.6 | 97.1 | 95.7 | 88.6 | 20 |
| **Mean** | **80.3** | **98.7** | **97.3** | **98.4** | **81.4** | **25.7** |
| **StDev** | **1.8** | **1.4** | **2.7** | **1.4** | **6.6** | **3.5** |
| **Variance** | **3.4** | **2.0** | **7.5** | **2.1** | **42.9** | **12.2** |
| **Mean&CI** | **80.3±1.1** | **98.7±0.9** | **97.3±1.7** | **98.4±0.9** | **81.4±4.1** | **25.7±2.2** |

## F.4 See5 single 5-class and aggregate models for KDD Cup 1999

Tables F.13 to F.16 give the details of predictive accuracy and TPRATE values for the See5 single 5-class, OVA and pVn aggregate models KDD Cup 1999.

*Table F.13: Predictive performance of the See5 single model for KDD Cup 1999*

| **Test set** | Accuracy on all classes | See5 single model (equal class distribution for NORMAL, DOS, PROBE, R2L). TPRATE% for class: | | | | |
|---|---|---|---|---|---|---|
| | | **NORMAL** | **DOS** | **PROBE** | **R2L** | **U2R** |
| 1 | 65.1 | 84.3 | 84.3 | 44.3 | 35.7 | 77.1 |
| 2 | 66.0 | 91.4 | 75.7 | 38.6 | 47.1 | 77.1 |
| 3 | 63.1 | 91.4 | 77.1 | 34.3 | 35.7 | 77.1 |
| 4 | 63.7 | 88.6 | 85.7 | 35.7 | 31.4 | 77.1 |
| 5 | 67.1 | 82.9 | 95.7 | 34.3 | 45.7 | 77.1 |
| 6 | 63.4 | 90.0 | 85.7 | 31.4 | 32.9 | 77.1 |
| 7 | 65.4 | 90.0 | 81.4 | 38.6 | 40.0 | 77.1 |
| 8 | 60.0 | 80.0 | 78.6 | 31.4 | 32.9 | 77.1 |
| 9 | 63.4 | 84.3 | 77.1 | 38.6 | 40.0 | 77.1 |
| 10 | 61.1 | 77.1 | 78.6 | 37.1 | 35.7 | 77.1 |
| **Mean** | **63.83** | **86.00** | **81.99** | **36.43** | **37.71** | **77.10** |
| **StDev** | **2.17** | **5.03** | **6.07** | **3.90** | **5.38** | **0.00** |
| **Variance** | **4.72** | **25.30** | **36.84** | **15.19** | **28.97** | **0.00** |
| **Mean & CI** | **63.8±1.3** | **86.0±3.1** | **82.0±3.8** | **36.4±2.4** | **37.7±3.3** | **77.1±0.0** |

*Table F.14: Predictive performance of the See5 un-boosted OVA aggregate model for KDD Cup1999*

| Test set | Accuracy on all classes | See5 Class TPRATE% - boosted AGGREGATE MODEL | | | | |
|---|---|---|---|---|---|---|
| | | **NORMAL** | **DOS** | **PROBE** | **R2L** | **U2R** |
| 1 | 62.3 | 97.1 | 45.7 | 88.6 | 34.3 | 45.7 |
| 2 | 65.7 | 100 | 54.3 | 87.1 | 41.4 | 45.7 |
| 3 | 60.9 | 98.6 | 42.9 | 85.7 | 31.4 | 45.7 |
| 4 | 66.6 | 98.6 | 61.4 | 88.6 | 38.6 | 45.7 |
| 5 | 64.9 | 98.6 | 54.3 | 84.3 | 41.1 | 45.7 |
| 6 | 61.1 | 97.1 | 37.1 | 91.4 | 34.3 | 45.7 |
| 7 | 64.3 | 98.6 | 55.7 | 87.1 | 34.3 | 45.7 |
| 8 | 62.3 | 97.1 | 51.4 | 90 | 27.1 | 45.7 |
| 9 | 62.6 | 100 | 52.9 | 88.6 | 25.7 | 45.7 |
| 10 | 62.3 | 97.1 | 45.7 | 88.6 | 34.3 | 45.7 |
| **Mean** | **63.3** | **98.3** | **50.1** | **88.0** | **34.3** | **45.7** |
| **StDev** | **2.0** | **1.1** | **7.2** | **2.0** | **5.3** | **0.0** |
| **Variance** | **3.8** | **1.3** | **51.5** | **4.2** | **27.7** | **0.0** |
| **Mean & CI** | **63.3±1.2** | **98.3±0.7** | **50.1±4.4** | **88.0±1.3** | **34.3±3.3** | **45.7±0.0** |

*Table F.15: Predictive performance of the See5 boosted OVA aggregate model for KDD Cup1999*

| Test set | Accuracy on all classes | See5 boosted OVA aggregate model. TPRATE% for class: | | | | |
|---|---|---|---|---|---|---|
| | | **NORMAL** | **DOS** | **PROBE** | **R2L** | **U2R** |
| 1 | 63.1 | 97.1 | 65.7 | 88.6 | 24.3 | 40.0 |
| 2 | 63.7 | 100.0 | 61.4 | 90.0 | 27.1 | 40.0 |
| 3 | 60.9 | 100.0 | 50.0 | 88.6 | 25.7 | 40.0 |
| 4 | 61.7 | 100.0 | 61.4 | 90.0 | 17.1 | 40.0 |
| 5 | 61.4 | 98.6 | 54.3 | 84.3 | 30.0 | 40.0 |
| 6 | 59.1 | 98.6 | 42.9 | 92.9 | 21.4 | 40.0 |
| 7 | 62.9 | 100.0 | 61.4 | 88.6 | 24.3 | 40.0 |
| 8 | 60.3 | 98.6 | 52.9 | 90.0 | 20.0 | 40.0 |
| 9 | 61.1 | 100.0 | 60.0 | 91.4 | 14.3 | 40.0 |
| 10 | 62.3 | 98.6 | 52.9 | 88.6 | 31.4 | 40.0 |
| **Mean** | **61.65** | **99.15** | **56.29** | **89.30** | **23.56** | **40.00** |
| **StDev** | **1.40** | **1.00** | **6.88** | **2.26** | **5.44** | **0.00** |
| **Variance** | **1.95** | **1.00** | **47.38** | **5.09** | **29.55** | **0.00** |
| **Mean & CI** | **61.7±0.9** | **99.2±0.6** | **56.3±4.3** | **89.3±1.4** | **23.6±3.4** | **40.0±0.0** |

*Table F.16: Predictive performance of the See5 pVn aggregate model for KDD Cup 1999*

| Test sample ID | Accuracy on all classes | See5 pVn aggregate model. TPRATE% for class: | | | | |
|---|---|---|---|---|---|---|
| | | NORMAL | DOS | PROBE | R2L | U2R |
| 1 | 74 | 97.1 | 67.1 | 98.6 | 30 | 77.1 |
| 2 | 79.1 | 98.6 | 57.1 | 97.1 | 65.7 | 77.1 |
| 3 | 78 | 98.6 | 60 | 97.1 | 57.1 | 77.1 |
| 4 | 83.4 | 98.6 | 87.1 | 95.7 | 58.6 | 77.1 |
| 5 | 85.1 | 100 | 84.3 | 97.1 | 67.1 | 77.1 |
| 6 | 78 | 97.1 | 64.3 | 100 | 51.4 | 77.1 |
| 7 | 81.1 | 98.6 | 71.4 | 95.7 | 62.9 | 77.1 |
| 8 | 77.7 | 97.1 | 70 | 97.1 | 47.1 | 77.1 |
| 9 | 74.9 | 98.6 | 55.7 | 97.1 | 45.7 | 77.1 |
| 10 | 78.3 | 97.1 | 67.1 | 94.3 | 55.7 | 77.1 |
| **Mean** | **78.96** | **98.14** | **68.41** | **96.98** | **54.13** | **77.10** |
| **StDev** | **3.45** | **0.99** | **10.51** | **1.57** | **11.16** | **0.00** |
| **Variance** | **11.88** | **0.98** | **110.42** | **2.48** | **124.50** | **0.00** |
| **Mean & CI** | **79.0 ± 2.1** | **98.1 ± 0.6** | **68.4 ± 6.5** | **97.0 ± 1.0** | **54.1 ± 6.9** | **77.1 ± 0.0** |

## F.5 Single and aggregate models for wine quality (white)

Tables F.17 through F.24 give the details of predictive accuracy and TPRATE values for the 5NN single and aggregate models for the wine quality (white) dataset. Tables F.25 and F.26 provide the statistical test results for the comparison of the single and aggregate models.

*Table F.17: Predictive performance of the 5NN single model for Wine quality*

| Test set | Accuracy on all classes | 5NN single model TPRATE% for class: | | | | |
|---|---|---|---|---|---|---|
| | | 4 | 5 | 6 | 7 | 8 |
| 1 | 31.2 | 8 | 56 | 22 | 54 | 8 |
| 2 | 30 | 14 | 58 | 30 | 44 | 4 |
| 3 | 29.2 | 10 | 56 | 24 | 44 | 12 |
| 4 | 28.8 | 6 | 54 | 30 | 46 | 8 |
| 5 | 33.2 | 14 | 54 | 34 | 54 | 10 |
| 6 | 32.4 | 12 | 54 | 34 | 54 | 8 |
| 7 | 30.8 | 14 | 46 | 36 | 44 | 14 |
| 8 | 34 | 18 | 50 | 36 | 54 | 12 |
| 9 | 35.2 | 10 | 64 | 38 | 50 | 14 |
| 10 | 31.6 | 10 | 56 | 30 | 50 | 12 |
| **Mean** | **31.6** | **11.6** | **54.8** | **31.4** | **49.4** | **10.2** |
| **StDev** | **2.1** | **3.5** | **4.7** | **5.3** | **4.5** | **3.2** |
| **Variance** | **4.3** | **12.3** | **22.4** | **27.6** | **20.5** | **10.2** |
| **Mean & CI** | **31.6±1.3** | **11.6±2.2** | **54.8±2.9** | **31.4±3.3** | **49.4±2.8** | **10.2±2.0** |

*Table F.18: Predictive performance of the 5NN un-boosted OVA model for Wine quality*

| Test set | Accuracy on all classes | 5NN OVA un-boosted model TPRATE% for class: | | | | |
|---|---|---|---|---|---|---|
| | | 4 | 5 | 6 | 7 | 8 |
| 1 | 30.4 | 16 | 54 | 24 | 50 | 8 |
| 2 | 32.8 | 14 | 60 | 28 | 56 | 6 |
| 3 | 35.2 | 10 | 64 | 34 | 54 | 14 |
| 4 | 28.8 | 6 | 58 | 32 | 40 | 8 |
| 5 | 33.6 | 14 | 60 | 34 | 52 | 8 |
| 6 | 30.8 | 14 | 54 | 32 | 48 | 6 |
| 7 | 30.4 | 12 | 58 | 26 | 44 | 12 |
| 8 | 34 | 18 | 58 | 30 | 52 | 12 |
| 9 | 32.8 | 10 | 56 | 30 | 54 | 14 |
| 10 | 32.8 | 12 | 68 | 24 | 48 | 12 |
| **Mean** | **32.2** | **12.6** | **59.0** | **29.4** | **49.8** | **10.0** |
| **StDev** | **1.9** | **3.4** | **4.3** | **3.8** | **4.9** | **3.1** |
| **Variance** | **3.8** | **11.6** | **18.9** | **14.3** | **24.4** | **9.8** |
| **Mean & CI** | **32.2±1.2** | **12.6±2.1** | **59.0±2.7** | **29.9±2.3** | **49.8±3.1** | **10.0±1.9** |

*Table F.19: Predictive performance of the 5NN boosted OVA model for Wine quality*

| Test set | Accuracy on all classes | 5NN OVA boosted model TPRATE% for class: | | | | |
|---|---|---|---|---|---|---|
| | | 4 | 5 | 6 | 7 | 8 |
| 1 | 33.2 | 16 | 64 | 24 | 52 | 10 |
| 2 | 33.2 | 16 | 68 | 16 | 58 | 8 |
| 3 | 35.2 | 12 | 68 | 26 | 54 | 16 |
| 4 | 29.2 | 6 | 66 | 22 | 44 | 8 |
| 5 | 35.2 | 16 | 64 | 34 | 52 | 10 |
| 6 | 29.6 | 14 | 62 | 16 | 50 | 6 |
| 7 | 35.2 | 14 | 68 | 34 | 44 | 16 |
| 8 | 35.6 | 18 | 62 | 28 | 56 | 14 |
| 9 | 34.4 | 12 | 64 | 22 | 60 | 14 |
| 10 | 34.8 | 12 | 70 | 28 | 52 | 12 |
| **Mean** | **33.6** | **13.6** | **65.6** | **25.0** | **52.2** | **11.4** |
| **StDev** | **2.3** | **3.4** | **2.8** | **6.3** | **5.3** | **3.5** |
| **Variance** | **5.5** | **11.4** | **7.8** | **40.2** | **28.0** | **12.5** |
| **Mean & CI** | **33.6±1.5** | **13.6±2.1** | **65.6±1.7** | **25.0±3.9** | **52.2±3.3** | **11.4±2.2** |

*Table F.20: Predictive performance of the 5NN pVn model for Wine quality*

| Test set | Accuracy on all classes | 5NN pVn aggregate model TPRATE% for class: | | | | |
|---|---|---|---|---|---|---|
| | | 4 | 5 | 6 | 7 | 8 |
| 1 | 33.2 | 16 | 44 | 52 | 46 | 8 |
| 2 | 34.8 | 12 | 56 | 50 | 50 | 6 |
| 3 | 36 | 12 | 58 | 50 | 46 | 14 |
| 4 | 31.2 | 4 | 54 | 58 | 32 | 8 |
| 5 | 37.6 | 14 | 60 | 60 | 44 | 10 |
| 6 | 32.4 | 12 | 54 | 54 | 34 | 8 |
| 7 | 32.4 | 10 | 56 | 46 | 36 | 14 |
| 8 | 35.6 | 18 | 52 | 42 | 54 | 12 |
| 9 | 34 | 6 | 50 | 50 | 50 | 14 |
| 10 | 38.4 | 10 | 66 | 54 | 50 | 12 |
| **Mean** | **34.6** | **11.4** | **55.0** | **51.6** | **44.2** | **10.6** |
| **StDev** | **2.4** | **4.2** | **5.9** | **5.3** | **7.6** | **3.0** |
| **Variance** | **5.6** | **17.8** | **34.9** | **28.3** | **58.2** | **8.9** |
| **Mean & CI** | **34.6±1.5** | **11.4±2.6** | **55.0±3.7** | **51.6±3.3** | **44.2±4.7** | **10.6±1.9** |

*Table F.21: Predictive performance of the See5 single model for Wine quality*

| Test set | Accuracy on all classes | See5 single model TPRATE% for class: | | | | |
|---|---|---|---|---|---|---|
| | | 4 | 5 | 6 | 7 | 8 |
| 1 | 38.4 | 28 | 70 | 32 | 54 | 8 |
| 2 | 37.6 | 24 | 70 | 34 | 52 | 8 |
| 3 | 38.4 | 28 | 74 | 32 | 50 | 8 |
| 4 | 33.6 | 20 | 64 | 26 | 46 | 12 |
| 5 | 36.4 | 28 | 70 | 32 | 48 | 4 |
| 6 | 37.2 | 30 | 72 | 30 | 46 | 8 |
| 7 | 36.8 | 28 | 70 | 36 | 44 | 6 |
| 8 | 37.2 | 28 | 66 | 36 | 46 | 10 |
| 9 | 38 | 26 | 70 | 34 | 50 | 10 |
| 10 | 34 | 20 | 74 | 30 | 42 | |
| **Mean** | **36.8** | **26.0** | **70.0** | **32.2** | **47.8** | **8.2** |
| **StDev** | **1.7** | **3.5** | **3.1** | **3.0** | **3.7** | **2.3** |
| **Variance** | **2.9** | **12.4** | **9.8** | **9.3** | **13.7** | **5.4** |
| **Mean & CI** | **36.8±1.0** | **26.0±2.2** | **70.0±1.9** | **32.2±1.9** | **47.8±2.3** | **8.2±1.4** |

*Table F.22: Predictive performance of the See5 un-boosted model for Wine quality*

| Test set | Accuracy on all classes | See5 un-boosted OVA model TPRATE% for class: | | | | |
|---|---|---|---|---|---|---|
| | | 4 | 5 | 6 | 7 | 8 |
| 1 | 34 | 42 | 64 | 14 | 36 | 14 |
| 2 | 34.8 | 38 | 68 | 20 | 40 | 8 |
| 3 | 38 | 42 | 70 | 12 | 48 | 18 |
| 4 | 29.6 | 26 | 68 | 6 | 42 | 6 |
| 5 | 36.4 | 48 | 70 | 10 | 42 | 12 |
| 6 | 32 | 34 | 66 | 10 | 40 | 10 |
| 7 | 36 | 46 | 58 | 14 | 46 | 16 |
| 8 | 38 | 46 | 60 | 18 | 52 | 14 |
| 9 | 34 | 40 | 58 | 14 | 38 | 20 |
| 10 | 35.6 | 40 | 64 | 16 | 44 | 14 |
| **Mean** | **34.8** | **40.2** | **64.6** | **13.4** | **42.8** | **13.2** |
| **StDev** | **2.6** | **6.5** | **4.6** | **4.1** | **4.8** | **4.3** |
| **Variance** | **6.8** | **42.2** | **21.4** | **16.9** | **23.3** | **18.8** |
| **Mean & CI** | **34.8±1.6** | **40.2±4.0** | **64.6±2.9** | **13.4±2.6** | **42.8±3.0** | **13.2±2.7** |

*Table F.23: Predictive performance of the See5 boosted model for Wine quality*

| Test set | Accuracy on all classes | See5 boosted OVA model TPRATE% for class: | | | | |
|---|---|---|---|---|---|---|
| | | 4 | 5 | 6 | 7 | 8 |
| 1 | 36.4 | 42 | 68 | 14 | 42 | 16 |
| 2 | 36 | 38 | 72 | 16 | 46 | 8 |
| 3 | 37.6 | 42 | 74 | 6 | 48 | 18 |
| 4 | 31.2 | 26 | 72 | 4 | 46 | 8 |
| 5 | 36.4 | 48 | 72 | 6 | 42 | 14 |
| 6 | 33.2 | 34 | 66 | 10 | 44 | 12 |
| 7 | 36.4 | 46 | 62 | 8 | 50 | 16 |
| 8 | 38.8 | 46 | 64 | 14 | 56 | 14 |
| 9 | 35.2 | 40 | 62 | 12 | 42 | 20 |
| 10 | 34.8 | 40 | 66 | 8 | 46 | 14 |
| **Mean** | **35.6** | **40.2** | **67.8** | **9.8** | **46.2** | **14.0** |
| **StDev** | **2.2** | **6.5** | **4.5** | **4.0** | **4.4** | **3.9** |
| **Variance** | **4.7** | **42.2** | **20.0** | **16.4** | **19.1** | **15.1** |
| **Mean & CI** | **35.6±1.3** | **40.2±4.0** | **67.8±2.8** | **9.8±2.5** | **46.2±2.7** | **14.0±2.4** |

*Table F.24: Predictive performance of the See5 pVn model for Wine quality*

| Test set | Accuracy on all classes | See5 pVn model TPRATE% for class: | | | | |
|---|---|---|---|---|---|---|
| | | 4 | 5 | 6 | 7 | 8 |
| 1 | 42 | 34 | 56 | 44 | 60 | 16 |
| 2 | 39.6 | 28 | 54 | 42 | 64 | 10 |
| 3 | 42.4 | 40 | 58 | 36 | 64 | 14 |
| 4 | 38 | 26 | 58 | 38 | 58 | 10 |
| 5 | 40.8 | 38 | 56 | 42 | 54 | 14 |
| 6 | 39.2 | 38 | 50 | 42 | 50 | 16 |
| 7 | 42.8 | 34 | 62 | 48 | 54 | 16 |
| 8 | 41.2 | 38 | 50 | 40 | 64 | 14 |
| 9 | 38.8 | 32 | 46 | 42 | 56 | 18 |
| 10 | 40.8 | 36 | 60 | 32 | 60 | 16 |
| **Mean** | **40.6** | **34.4** | **55.0** | **40.6** | **58.4** | **14.4** |
| **StDev** | **1.6** | **4.6** | **5.0** | **4.4** | **4.9** | **2.6** |
| **Variance** | **2.6** | **21.2** | **25.1** | **19.6** | **23.8** | **6.9** |
| **Mean & CI** | **40.6±1.0** | **34.4±2.9** | **55.0±3.1** | **40.6±2.7** | **58.4±3.0** | **14.4±1.6** |

*Table F.25: Statistical tests for 5NN single and aggregate model comparison for wine quality*

| Wine quality white: 5NN models | | | | | | |
|---|---|---|---|---|---|---|
| Group names and mean accuracy /TPRATE:10 test sets | | Student's paired t-test (9 df) | | | Performance improvement measures | |
| Group A aggregate model | Group B single model | 95% CI of mean difference | P value (2 tail) | Group A better than Group B? | Diff(A,B)% | Ratio(A,B) |
| OVA un-boosted All classes-A (32.2 ± 1.2) | All classes-S (31.6 ± 1.3) | [-1.2, 2.2] | 0.511 | no | 0.5 | 0.01 |
| OVA un-boosted Class4-A (12.6 ± 2.1) | Class4-S (11.6 ± 2.2) | [-0.9, 2.9] | 0.273 | no | 1.0 | 0.01 |
| OVA un-boosted Class5-A 59.0 ± 2.7) | Class5-S (54.8 ± 2.9) | [-0.3, 8.7] | 0.066 | yes | 4.2 | 0.09 |
| OVA un-boosted Class6-A (29.9 ± 2.3) | Class6-S (31.4 ± 3.3) | [-6.2, 2.2] | 0.311 | no | -2.0 | -0.03 |
| OVA un-boosted Class7-A (49.8 ± 3.1) | Class7-S (49.4 ± 2.8) | [-4.1, 4.9] | 0.846 | no | 0.4 | 0.01 |
| OVA un-boosted Class8-A (10.0 ± 1.9) | Class8-S (10.2 ± 2.0) | [-1.3, 0.9] | 0.678 | no | -0.2 | 0.00 |
| | | | | | | |
| OVA boosted All classes-A (33.6 ± 1.5) | All classes-S (31.6 ± 1.3) | [0.1, 3.7] | 0.041 | yes | 1.9 | 0.03 |
| OVA boosted Class4-A 13.6 ± 2.1) | Class4-S (11.6 ± 2.2) | [0.3, 3.7] | 0.023 | yes | 2.0 | 0.02 |
| OVA boosted Class5-A (65.6 ± 1.7) | Class5-S (54.8 ± 2.9) | [6.9, 14.7] | 0.000 | yes | 10.8 | 0.24 |
| OVA boosted Class6-A (25.0 ± 3.9) | Class6-S (31.4 ± 3.3) | [-11.8, -1.0] | 0.025 | no | -6.4 | -0.09 |
| OVA boosted Class7-A (52.2 ± 3.3) | Class7-S (49.4 ± 2.8) | [-1.6, 7.3] | 0.191 | no | 2.8 | 0.06 |
| OVA un-boosted Class8-A (11.4 ± 2.2) | Class8-S (10.2 ± 2.0) | [-0.2, 2.9] | 0.081 | yes | 1.2 | 0.01 |
| | | | | | | |
| pVn All classes-A (34.6 ± 1.5) | All classes-S (31.6 ± 1.3) | [1.0, 4.9] | 0.008 | yes | 2.9 | 0.04 |
| pVn Class4-A (11.4 ± 2.6) | Class4-S (11.6 ± 2.2) | [-2.7, 2.3] | 0.859 | no | -0.2 | 0.00 |
| pVn Class5-A (55.0 ± 3.7) | Class5-S (54.8 ± 2.9) | -5.6, 6.0] | 0.939 | no | 0.2 | 0.00 |
| pVn Class6-A (51.6 ± 3.3) | Class6-S (31.4 ± 3.3) | [14.3, 26.1] | 0.000 | yes | 20.2 | 0.29 |
| pVn Class7-A (44.2 ± 4.7) | Class7-S (49.4 ± 2.8) | [-11.0, 0.6] | 0.074 | no | -5.2 | -0.10 |
| pVn Class8-A (10.6 ± 1.9) | Class8-S (10.2 ± 2.0) | [-0.2, 1.0] | 0.168 | no | 0.4 | 0.00 |

*Table F.26: Statistical tests for See5 single and aggregate model comparison for wine quality*

| **Wine quality white: See5 models** | | | | | | |
|---|---|---|---|---|---|---|
| Group names and mean accuracy /TPRATE:10 test sets | | Student's paired t-test (9 df) | | | Performance improvement measures | |
| Group A aggregate model | Group B single model | 95% CI of mean difference | P value (2 tail) | Group A better than Group B? | Diff(A,B)% | Ratio(A,B) |
| OVA un-boosted All classes-A (34.8 ± 1.6) | All classes-S (36.8 ± 1.0) | [-3.7, -0.2] | 0.034 | no | -1.9 | -0.03 |
| OVA un-boosted Class4-A (40.2 ± 4.0) | Class4-S (26.0 ± 2.2) | [10.3, 18.1] | 0.000 | yes | 14.2 | 0.19 |
| OVA un-boosted Class5-A 64.6 ± 2.9) | Class5-S (70.0 ± 1.9) | [-9.1, -1.7] | 0.009 | no | -5.4 | -0.18 |
| OVA un-boosted Class6-A (13.4 ± 2.9) | Class6-S (32.2 ± 1.9) | [-20.8, -16.8] | 0.000 | no | -18.8 | -0.28 |
| OVA un-boosted Class7-A (42.8 ± 3.8) | Class7-S (47.8 ± 2.8) | [-10.3, 0.3] | 0.062 | no | -5.0 | -0.10 |
| OVA un-boosted Class8-A (13.2 ± 2.7) | Class8-S (8.2 ± 1.4) | [0.7, 9.1] | 0.028 | yes | 5.0 | 0.05 |
| | | | | | | |
| OVA boosted All classes-A (35.6 ± 1.3) | All classes-S (36.8 ± 1.0) | [-2.4, 0.1] | 0.062 | no | -1.2 | -0.02 |
| OVA boosted Class4-A (40.2 ± 4.0) | Class4-S (26.0 ± 2.2) | [10.3, 18.1] | 0.000 | yes | 14.2 | 0.19 |
| OVA boosted Class5-A (67.8 ± 2.8) | Class5-S (70.0 ± 1.9) | [-6.0, 1.6] | 0.227 | no | -2.2 | -0.07 |
| OVA boosted Class6-A (9.8 ± 2.5) | Class6-S (32.2 ± 1.9) | [-24.8, -20.0] | 0.000 | no | -22.4 | -0.33 |
| OVA boosted Class7-A (46.2 ± 2.7) | Class7-S (47.8 ± 2.8) | [-6.5, 3.3] | 0.475 | no | -1.6 | -0.03 |
| OVA un-boosted Class8-A (14.0 ± 2.4) | Class8-S (8.2 ± 1.4) | [1.8, 9.7] | 0.100 | yes | 5.8 | 0.06 |
| | | | | | | |
| pVn All classes-A (40.6 ± 1.0) | All classes-S (36.8 ± 1.0) | [2.4, 5.1] | 0.000 | yes | 3.8 | 0.06 |
| pVn Class4-A (34.4 ± 2.9) | Class4-S (26.0 ± 2.2) | [5.8, 11.0] | 0.000 | yes | 8.4 | 0.11 |
| pVn Class5-A (55.0 ± 3.1) | Class5-S (70.0 ± 1.9) | [-18.9, -11.1] | 0.000 | no | -15.0 | -0.50 |
| pVn Class6-A (40.6 ± 2.7) | Class6-S (32.2 ± 1.9) | [5.6, 11.2] | 0.000 | yes | 8.4 | 0.12 |
| pVn Class7-A (58.4 ± 3.0) | Class7-S (47.8 ± 2.8) | [7.0, 14.2] | 0.000 | yes | 10.6 | 0.20 |
| pVn Class8-A (14.4 ± 1.6] | Class8-S (8.2 ± 1.4) | [2.9, 9.1] | 0.002 | yes | 6.2 | 0.07 |

# Appendix G

# ROC analysis details

The computational method for the AUC and the detailed results for ROC analysis are provided in this appendix. The ROC analysis that was conducted for the experiments was discussed in chapter 9. The method used to compute the Area Under the ROC curve (AUC) is depicted in figure G.1 and table G.1. Figure G.1 shows a ROC curve created with three points corresponding to three threshold points $\lambda_1, \lambda_2$ and $\lambda_3$. The x-axis and y-axis respectively represent the FPRATE and TPRATE of a probabilistic classifier. Threshold averaging was used for the computation of the AUC. Recall from chapter 9 that for threshold averaging, the co-ordinates of each point on the ROC curve are obtained by computing the mean FPRATE (x co-ordinate) and mean TPRATE (y co-ordinate) for one threshold value $\lambda_i$. The mean FPRATE and TPRATE values were computed for 10 test sets. The areas of regions A1 to A7 were used to compute the AUC as shown in table G.1.

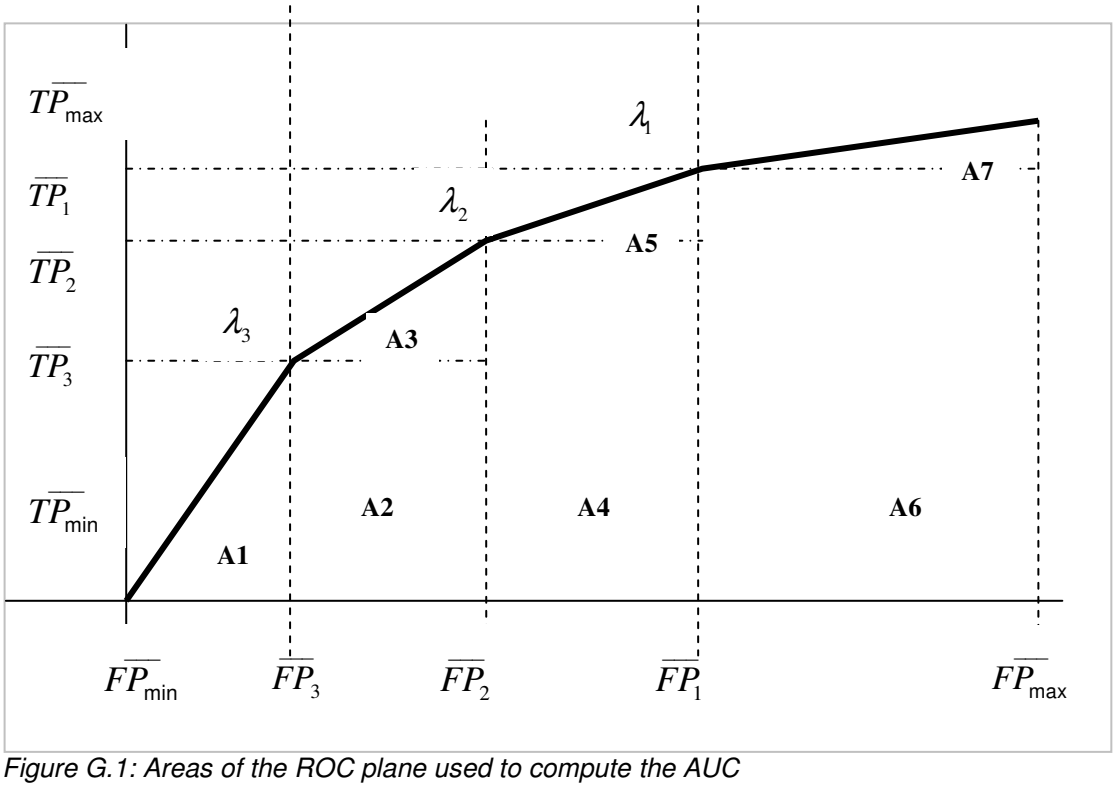


*Figure G.1: Areas of the ROC plane used to compute the AUC*

*Table G.1: Method used for the computation of the AUC for probabilistic classifiers*

| Area code | Computation |
|---|---|
| A1 | ½ * (FP3 * TP3) |
| A2 | (FP2 – FP3) * TP3 |
| A3 | ½ * (FP2 – FP3) * (TP2 – TP3) |
| A4 | (FP1 – FP2) * TP2 |
| A5 | ½ * (FP1 – FP2) * (TP1 - TP2) |
| A6 | (FPmax - FP1) * TP1 |
| A7 | ½ * (FPmax - FP1) * (TPmax – TP1) |
| TOTAL | A1 + A2 + A3 + A4 + A5 + A6 + A7 |
| $AUC_{above}$ | (TOTAL – area under 45deg line) |
| AUC | TOTAL |

Tables G.2 to G.7 provide the details of the FPRATE values (FP1, FP2, FP3) and TPRATE values (TP1, TP2, TP3) and AUC values for the forest cover type, KDD Cup 1999 and wine quality datasets. The AUC is the area between the x-axis, y-axis and ROC curve. $AUC_{above}$ is the area between the 45 degree line and the ROC curve. The threshold values of 0.6, 0.8 and 1.0 for the 5NN classifiers correspond to the number of nearest neighbours (3, 4, 5) used by the 5NN algorithm to determine the winning class. The threshold values of 0.5, 0.75 and 1.0 were used for the See5 classifiers. The positive class column represents a *one-vs-rest* classifier which predicts the indicated class as the positive class and all the other classes as negative classes.

*Table G.2: One-vs-rest AUC for the 5NN forest cover type models*

| 5NN forest cover type models: TPRATE, FPRATE, AUC and Mean AUC | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Mean values for thresholds | | | | | | | |
| | | $\lambda_1 = 0.6$ | | $\lambda_2 = 0.8$ | | $\lambda_3 = 1.0$ | | | |
| Model | Positive class | FP1 | TP1 | FP2 | TP2 | FP3 | TP3 | *AUC* | $AUC_{above}$ |
| single 5NN | 1 | 0.04 | 0.62 | 0.02 | 0.37 | 0.00 | 0.17 | 0.79 | 0.29 |
| | 2 | 0.03 | 0.48 | 0.01 | 0.27 | 0.00 | 0.09 | 0.73 | 0.23 |
| | 3 | 0.03 | 0.51 | 0.01 | 0.32 | 0.00 | 0.15 | 0.75 | 0.25 |
| | 4 | 0.03 | 0.92 | 0.02 | 0.78 | 0.01 | 0.53 | 0.95 | 0.45 |
| | 5 | 0.03 | 0.88 | 0.02 | 0.78 | 0.01 | 0.48 | 0.93 | 0.43 |
| | 6 | 0.05 | 0.70 | 0.02 | 0.44 | 0.01 | 0.18 | 0.83 | 0.33 |
| | 7 | 0.03 | 0.95 | 0.01 | 0.82 | 0.01 | 0.64 | 0.97 | 0.47 |
| | | | | | | **Mean:** | | **0.85** | **0.35** |
| OVA unboosted 5NN | 1 | 0.03 | 0.70 | 0.03 | 0.69 | 0.03 | 0.58 | 0.83 | 0.33 |
| | 2 | 0.03 | 0.58 | 0.03 | 0.57 | 0.02 | 0.49 | 0.78 | 0.28 |
| | 3 | 0.03 | 0.72 | 0.03 | 0.72 | 0.02 | 0.60 | 0.85 | 0.35 |
| | 4 | 0.02 | 0.90 | 0.02 | 0.87 | 0.01 | 0.67 | 0.94 | 0.44 |
| | 5 | 0.04 | 0.96 | 0.04 | 0.96 | 0.03 | 0.89 | 0.96 | 0.46 |
| | 6 | 0.05 | 0.81 | 0.05 | 0.80 | 0.03 | 0.67 | 0.88 | 0.38 |
| | 7 | 0.03 | 0.97 | 0.03 | 0.97 | 0.02 | 0.91 | 0.97 | 0.47 |
| | | | | | | **Mean:** | | **0.89** | **0.39** |
| OVA boosted 5NN | 1 | 0.03 | 0.70 | 0.03 | 0.69 | 0.03 | 0.58 | 0.83 | 0.33 |
| | 2 | 0.03 | 0.62 | 0.03 | 0.60 | 0.02 | 0.51 | 0.80 | 0.30 |
| | 3 | 0.03 | 0.71 | 0.03 | 0.71 | 0.02 | 0.61 | 0.84 | 0.34 |
| | 4 | 0.02 | 1.00 | 0.02 | 1.00 | 0.01 | 1.00 | 0.99 | 0.49 |
| | 5 | 0.04 | 0.97 | 0.03 | 0.94 | 0.02 | 0.82 | 0.97 | 0.47 |
| | 6 | 0.04 | 0.78 | 0.04 | 0.75 | 0.03 | 0.63 | 0.87 | 0.37 |
| | 7 | 0.03 | 0.97 | 0.03 | 0.97 | 0.02 | 0.91 | 0.97 | 0.47 |
| | | | | | | **Mean:** | | **0.90** | **0.40** |
| pVn 5NN | 1 | 0.05 | 0.68 | 0.03 | 0.62 | 0.01 | 0.36 | 0.82 | 0.32 |
| | 2 | 0.04 | 0.57 | 0.03 | 0.50 | 0.02 | 0.30 | 0.77 | 0.27 |
| | 3 | 0.04 | 0.65 | 0.03 | 0.52 | 0.01 | 0.34 | 0.81 | 0.31 |
| | 4 | 0.03 | 0.97 | 0.02 | 0.83 | 0.01 | 0.68 | 0.98 | 0.48 |
| | 5 | 0.04 | 0.94 | 0.03 | 0.89 | 0.02 | 0.79 | 0.96 | 0.46 |
| | 6 | 0.05 | 0.75 | 0.03 | 0.65 | 0.01 | 0.39 | 0.86 | 0.36 |
| | 7 | 0.02 | 0.93 | 0.01 | 0.67 | 0.01 | 0.67 | 0.96 | 0.46 |
| | | | | | | **Mean:** | | **0.88** | 0.38 |

*Table G.3: One-vs-rest AUC for the 5NN KDD Cup 1999 models*

| 5NN KDD Cup 1999 models: TPRATE, FPRATE, AUC and Mean AUC | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Model | Positive class | Mean values for thresholds | | | | | | $AUC$ | $AUC_{above}$ |
| | | $\lambda_1 = 0.6$ | | $\lambda_2 = 0.8$ | | $\lambda_3 = 1.0$ | | | |
| | | FP1 | TP1 | FP2 | TP2 | FP3 | TP3 | | |
| single 5NN | NORM | 0.22 | 0.84 | 0.13 | 0.84 | 0.11 | 0.80 | 0.86 | 0.36 |
| | R2L | 0.06 | 0.65 | 0.05 | 0.60 | 0.04 | 0.53 | 0.80 | 0.30 |
| | DOS | 0.01 | 0.66 | 0.01 | 0.63 | 0.01 | 0.61 | 0.83 | 0.33 |
| | PROBE | 0.09 | 0.96 | 0.09 | 0.96 | 0.07 | 0.93 | 0.94 | 0.44 |
| | U2R | 0.01 | 0.31 | 0.01 | 0.26 | 0.01 | 0.20 | 0.65 | 0.15 |
| | | | | | | **Mean:** | | **0.82** | **0.32** |
| OVA unboosted 5NN | NORM | 0.14 | 0.92 | 0.13 | 0.92 | 0.10 | 0.92 | 0.91 | 0.41 |
| | R2L | 0.07 | 0.65 | 0.07 | 0.62 | 0.06 | 0.57 | 0.79 | 0.29 |
| | DOS | 0.00 | 0.66 | 0.00 | 0.66 | 0.00 | 0.65 | 0.83 | 0.33 |
| | PROBE | 0.08 | 0.95 | 0.08 | 0.95 | 0.08 | 0.95 | 0.94 | 0.44 |
| | U2R | 0.01 | 0.43 | 0.01 | 0.43 | 0.01 | 0.31 | 0.71 | 0.21 |
| | | | | | | **Mean:** | | **0.83** | **0.33** |
| OVA boosted 5NN | NORM | 0.15 | 0.92 | 0.13 | 0.92 | 0.10 | 0.92 | 0.91 | 0.41 |
| | R2L | 0.07 | 0.61 | 0.06 | 0.59 | 0.05 | 0.52 | 0.77 | 0.27 |
| | DOS | 0.00 | 0.66 | 0.00 | 0.66 | 0.00 | 0.66 | 0.83 | 0.33 |
| | PROBE | 0.08 | 0.95 | 0.08 | 0.95 | 0.08 | 0.95 | 0.94 | 0.44 |
| | U2R | 0.01 | 0.40 | 0.01 | 0.40 | 0.01 | 0.29 | 0.70 | 0.20 |
| | | | | | | **Mean:** | | **0.83** | **0.33** |
| pVn 5NN | NORM | 0.16 | 0.99 | 0.15 | 0.99 | 0.12 | 0.98 | 0.93 | 0.43 |
| | R2L | 0.07 | 0.81 | 0.06 | 0.81 | 0.06 | 0.78 | 0.88 | 0.38 |
| | DOS | 0.00 | 0.97 | 0.00 | 0.97 | 0.00 | 0.72 | 0.98 | 0.48 |
| | PROBE | 0.00 | 0.98 | 0.00 | 0.98 | 0.00 | 0.98 | 0.99 | 0.49 |
| | U2R | 0.01 | 0.26 | 0.01 | 0.20 | 0.00 | 0.05 | 0.63 | 0.13 |
| | | | | | | **Mean:** | | **0.88** | **0.33** |

*Table G.4: One-vs-rest AUC for the 5NN Wine quality models*

| 5NN Wine quality (white) models: Mean TPRATE, mean FPRATE, AUC and Mean AUC | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Mean values for thresholds | | | | | | | |
| | | $\lambda_1 = 0.6$ | | $\lambda_2 = 0.8$ | | $\lambda_3 = 1.0$ | | | |
| Model | positive class | FP1 | TP1 | FP2 | TP2 | FP3 | TP3 | *AUC* | $AUC_{above}$ |
| single 5NN | 4 | 0.04 | 0.12 | 0.02 | 0.08 | 0.01 | 0.04 | 0.54 | 0.04 |
| | 5 | 0.20 | 0.48 | 0.10 | 0.29 | 0.04 | 0.11 | 0.65 | 0.15 |
| | 6 | 0.17 | 0.22 | 0.04 | 0.08 | 0.01 | 0.01 | 0.53 | 0.03 |
| | 7 | 0.23 | 0.42 | 0.10 | 0.16 | 0.04 | 0.03 | 0.59 | 0.09 |
| | 8 | 0.02 | 0.10 | 0.01 | 0.07 | 0.00 | 0.03 | 0.54 | 0.04 |
| | | | | | | **Mean** | **AUC:** | **0.57** | **0.07** |
| OVA un-boosted 5NN | 4 | 0.05 | 0.13 | 0.05 | 0.13 | 0.04 | 0.09 | 0.54 | 0.04 |
| | 5 | 0.27 | 0.59 | 0.25 | 0.55 | 0.12 | 0.34 | 0.67 | 0.17 |
| | 6 | 0.22 | 0.29 | 0.19 | 0.26 | 0.11 | 0.16 | 0.54 | 0.04 |
| | 7 | 0.28 | 0.50 | 0.25 | 0.42 | 0.17 | 0.33 | 0.61 | 0.11 |
| | 8 | 0.02 | 0.10 | 0.02 | 0.10 | 0.02 | 0.10 | 0.54 | 0.04 |
| | | | | | | **Mean** | **AUC:** | **0.58** | **0.08** |
| OVA boosted 5NN | 4 | 0.05 | 0.14 | 0.05 | 0.14 | 0.05 | 0.10 | 0.54 | 0.04 |
| | 5 | 0.31 | 0.66 | 0.29 | 0.59 | 0.13 | 0.35 | 0.68 | 0.18 |
| | 6 | 0.13 | 0.25 | 0.09 | 0.20 | 0.02 | 0.08 | 0.56 | 0.06 |
| | 7 | 0.29 | 0.52 | 0.27 | 0.48 | 0.17 | 0.35 | 0.62 | 0.12 |
| | 8 | 0.03 | 0.11 | 0.03 | 0.11 | 0.02 | 0.11 | 0.54 | 0.04 |
| | | | | | | **Mean** | **AUC:** | **0.59** | **0.09** |
| pVn 5NN | 4 | 0.05 | 0.11 | 0.04 | 0.09 | 0.02 | 0.02 | 0.53 | 0.03 |
| | 5 | 0.23 | 0.53 | 0.15 | 0.39 | 0.04 | 0.16 | 0.66 | 0.16 |
| | 6 | 0.27 | 0.50 | 0.17 | 0.32 | 0.04 | 0.12 | 0.62 | 0.12 |
| | 7 | 0.22 | 0.44 | 0.15 | 0.28 | 0.06 | 0.08 | 0.60 | 0.10 |
| | 8 | 0.02 | 0.11 | 0.02 | 0.09 | 0.01 | 0.06 | 0.55 | 0.05 |
| | | | | | | **Mean** | **AUC:** | **0.59** | **0.09** |

*Table G.5: One-vs-rest AUC for the See5 forest cover type models*

| See5 forest cover type models: TPRATE, FPRATE, AUC and Mean AUC | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Mean values for thresholds | | | | | | | |
| | | $\lambda_1 = 0.5$ | | $\lambda_2 = 0.75$ | | $\lambda_3 = 1.0$ | | | |
| Model | Positive class | FP1 | TP1 | FP2 | TP2 | FP3 | TP3 | *AUC* | $AUC_{above}$ |
| single See5 | 1 | 0.03 | 0.57 | 0.01 | 0.28 | 0.00 | 0.04 | 0.77 | 0.27 |
| | 2 | 0.06 | 0.63 | 0.03 | 0.39 | 0.00 | 0.04 | 0.79 | 0.29 |
| | 3 | 0.03 | 0.61 | 0.01 | 0.41 | 0.00 | 0.04 | 0.79 | 0.29 |
| | 4 | 0.03 | 0.94 | 0.02 | 0.90 | 0.00 | 0.08 | 0.96 | 0.46 |
| | 5 | 0.03 | 0.86 | 0.02 | 0.77 | 0.00 | 0.00 | 0.92 | 0.42 |
| | 6 | 0.05 | 0.78 | 0.03 | 0.60 | 0.00 | 0.05 | 0.87 | 0.37 |
| | 7 | 0.03 | 0.96 | 0.02 | 0.85 | 0.01 | 0.03 | 0.97 | 0.47 |
| | | | | | | **Mean:** | | **0.87** | **0.37** |
| OVA unboosted 5NNSee5 | 1 | 0.05 | 0.61 | 0.05 | 0.60 | 0.00 | 0.01 | 0.78 | 0.28 |
| | 2 | 0.05 | 0.50 | 0.05 | 0.50 | 0.00 | 0.00 | 0.72 | 0.22 |
| | 3 | 0.04 | 0.64 | 0.04 | 0.62 | 0.00 | 0.02 | 0.80 | 0.30 |
| | 4 | 0.01 | 0.87 | 0.01 | 0.85 | 0.00 | 0.00 | 0.93 | 0.43 |
| | 5 | 0.04 | 0.94 | 0.04 | 0.94 | 0.00 | 0.01 | 0.95 | 0.45 |
| | 6 | 0.07 | 0.79 | 0.07 | 0.79 | 0.00 | 0.08 | 0.86 | 0.36 |
| | 7 | 0.03 | 0.93 | 0.03 | 0.93 | 0.00 | 0.00 | 0.95 | 0.45 |
| | | | | | | **Mean:** | | **0.86** | **0.36** |
| OVA boosted See5 | 1 | 0.03 | 0.63 | 0.02 | 0.52 | 0.00 | 0.04 | 0.80 | 0.30 |
| | 2 | 0.07 | 0.67 | 0.07 | 0.62 | 0.01 | 0.01 | 0.80 | 0.30 |
| | 3 | 0.02 | 0.63 | 0.02 | 0.62 | 0.00 | 0.08 | 0.80 | 0.30 |
| | 4 | 0.01 | 0.95 | 0.01 | 0.94 | 0.00 | 0.04 | 0.97 | 0.47 |
| | 5 | 0.04 | 0.87 | 0.04 | 0.87 | 0.00 | 0.09 | 0.92 | 0.42 |
| | 6 | 0.04 | 0.76 | 0.04 | 0.76 | 0.00 | 0.05 | 0.86 | 0.36 |
| | 7 | 0.01 | 0.98 | 0.01 | 0.97 | 0.00 | 0.22 | 0.98 | 0.48 |
| | | | | | | **Mean:** | | **0.88** | **0.38** |
| pVn See5 | 1 | 0.04 | 0.65 | 0.02 | 0.54 | 0.00 | 0.08 | 0.81 | 0.31 |
| | 2 | 0.06 | 0.65 | 0.05 | 0.61 | 0.00 | 0.04 | 0.80 | 0.30 |
| | 3 | 0.04 | 0.72 | 0.03 | 0.68 | 0.00 | 0.09 | 0.84 | 0.34 |
| | 4 | 0.01 | 0.95 | 0.01 | 0.89 | 0.00 | 0.01 | 0.97 | 0.47 |
| | 5 | 0.02 | 0.89 | 0.02 | 0.81 | 0.00 | 0.00 | 0.93 | 0.43 |
| | 6 | 0.05 | 0.82 | 0.04 | 0.78 | 0.00 | 0.12 | 0.89 | 0.39 |
| | 7 | 0.02 | 0.92 | 0.01 | 0.88 | 0.00 | 0.03 | 0.95 | 0.45 |
| | | | | | | **Mean:** | | **0.88** | **0.38** |

*Table G.6: One-vs-rest AUC for the See5 KDD Cup 1999 models*

| **See5 KDD Cup 1999 models: TPRATE, FPRATE, AUC and Mean AUC** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Mean values for thresholds | | | | | | | |
| | | $\lambda_1 = 0.5$ | | $\lambda_2 = 0.75$ | | $\lambda_3 = 1.0$ | | | |
| Model | Positive class | FP1 | TP1 | FP2 | TP2 | FP3 | TP3 | *AUC* | $AUC_{above}$ |
| single See5 | NORM | 0.22 | 0.86 | 0.22 | 0.86 | 0.02 | 0.63 | 0.88 | 0.38 |
| | R2L | 0.02 | 0.38 | 0.02 | 0.38 | 0.00 | 0.12 | 0.68 | 0.18 |
| | DOS | 0.02 | 0.82 | 0.02 | 0.82 | 0.02 | 0.82 | 0.90 | 0.40 |
| | PROBE | 0.04 | 0.36 | 0.04 | 0.36 | 0.02 | 0.36 | 0.67 | 0.17 |
| | U2R | 0.16 | 0.77 | 0.16 | 0.77 | 0.00 | 0.00 | 0.81 | 0.31 |
| | | | | | | **Mean:** | | **0.79** | **0.29** |
| OVA unboosted See5 | NORM | 0.11 | 0.98 | 0.11 | 0.98 | 0.10 | 0.98 | 0.94 | 0.44 |
| | R2L | 0.09 | 0.34 | 0.09 | 0.34 | 0.06 | 0.04 | 0.62 | 0.12 |
| | DOS | 0.00 | 0.50 | 0.00 | 0.50 | 0.00 | 0.01 | 0.75 | 0.25 |
| | PROBE | 0.10 | 0.88 | 0.10 | 0.88 | 0.10 | 0.88 | 0.89 | 0.39 |
| | U2R | 0.01 | 0.46 | 0.01 | 0.46 | 0.00 | 0.00 | 0.73 | 0.23 |
| | | | | | | **Mean:** | | **0.79** | **0.29** |
| OVA boosted See5 | NORM | 0.24 | 0.99 | 0.24 | 0.99 | 0.15 | 0.93 | 0.91 | 0.41 |
| | R2L | 0.02 | 0.24 | 0.02 | 0.24 | 0.00 | 0.01 | 0.61 | 0.11 |
| | DOS | 0.06 | 0.56 | 0.06 | 0.56 | 0.01 | 0.56 | 0.77 | 0.27 |
| | PROBE | 0.08 | 0.89 | 0.08 | 0.89 | 0.08 | 0.89 | 0.91 | 0.41 |
| | U2R | 0.01 | 0.40 | 0.01 | 0.40 | 0.00 | 0.00 | 0.69 | 0.19 |
| | | | | | | **Mean:** | | **0.78** | **0.28** |
| pVn See5 | NORM | 0.20 | 0.98 | 0.20 | 0.98 | 0.07 | 0.41 | 0.90 | 0.40 |
| | R2L | 0.02 | 0.54 | 0.02 | 0.54 | 0.01 | 0.22 | 0.76 | 0.26 |
| | DOS | 0.00 | 0.68 | 0.00 | 0.68 | 0.00 | 0.44 | 0.84 | 0.34 |
| | PROBE | 0.03 | 0.97 | 0.03 | 0.97 | 0.01 | 0.97 | 0.98 | 0.48 |
| | U2R | 0.01 | 0.77 | 0.01 | 0.71 | 0.00 | 0.43 | 0.88 | 0.38 |
| | | | | | | **Mean:** | | **0.87** | **0.37** |

*Table G.7: One-vs-rest AUC for the See5 Wine quality models*

| See5 Wine quality white: TPRATE, FPRATE,auc and MEAN AUC | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Mean values for thresholds | | | | | | | |
| | | $\lambda_1 = 0.5$ | | $\lambda_2 = 0.75$ | | $\lambda_3 = 1.0$ | | | |
| model | positive Class | FP1 | TP1 | FP2 | TP2 | FP3 | TP3 | *AUC* | $AUC_{above}$ |
| single See5 | 4 | 0.04 | 0.26 | 0.04 | 0.26 | 0.00 | 0.01 | 0.61 | 0.11 |
| | 5 | 0.33 | 0.70 | 0.03 | 0.05 | 0.00 | 0.00 | 0.68 | 0.18 |
| | 6 | 0.18 | 0.28 | 0.02 | 0.04 | 0.00 | 0.00 | 0.55 | 0.05 |
| | 7 | 0.19 | 0.48 | 0.05 | 0.14 | 0.00 | 0.00 | 0.64 | 0.14 |
| | 8 | 0.01 | 0.08 | 0.00 | 0.08 | 0.00 | 0.00 | 0.54 | 0.04 |
| | | | | | | **Mean:** | | **0.60** | **0.10** |
| un-boosted OVA See5 | 4 | 0.09 | 0.40 | 0.09 | 0.40 | 0.01 | 0.09 | 0.66 | 0.16 |
| | 5 | 0.30 | 0.65 | 0.30 | 0.65 | 0.02 | 0.01 | 0.67 | 0.17 |
| | **6** | 0.12 | 0.13 | 0.10 | 0.13 | 0.01 | 0.00 | 0.51 | 0.01 |
| | 7 | 0.25 | 0.43 | 0.24 | 0.43 | 0.00 | 0.00 | 0.59 | 0.09 |
| | 8 | 0.03 | 0.13 | 0.03 | 0.13 | 0.00 | 0.00 | 0.55 | 0.05 |
| | | | | | | **Mean:** | | **0.60** | **0.10** |
| boosted OVA See5 | 4 | 0.09 | 0.40 | 0.09 | 0.40 | 0.01 | 0.09 | 0.66 | 0.16 |
| | 5 | 0.33 | 0.68 | 0.31 | 0.68 | 0.02 | 0.01 | 0.68 | 0.18 |
| | 6 | 0.07 | 0.10 | 0.01 | 0.02 | 0.00 | 0.00 | 0.51 | 0.01 |
| | 7 | 0.26 | 0.46 | 0.24 | 0.45 | 0.00 | 0.00 | 0.60 | 0.10 |
| | **8** | 0.03 | 0.14 | 0.03 | 0.13 | 0.00 | 0.00 | 0.56 | 0.06 |
| | | | | | | **Mean:** | | **0.60** | **0.10** |
| pVn See5 | 4 | 0.06 | 0.34 | 0.06 | 0.34 | 0.01 | 0.09 | 0.64 | 0.14 |
| | 5 | 0.19 | 0.55 | 0.14 | 0.48 | 0.00 | 0.00 | 0.69 | 0.19 |
| | 6 | 0.19 | 0.41 | 0.12 | 0.27 | 0.01 | 0.02 | 0.61 | 0.11 |
| | 7 | 0.29 | 0.58 | 0.25 | 0.56 | 0.03 | 0.06 | 0.66 | 0.16 |
| | 8 | 0.02 | 0.14 | 0.02 | 0.14 | 0.01 | 0.00 | 0.56 | 0.06 |
| | | | | | | **Mean:** | | **0.63** | **0.13** |

# Appendix H

# Using statistical and database software to implement dataset selection methods

Recommendations for using database and statistical software for the implementation of dataset selection methods proposed in this thesis were given in chapter 10. Tables H.1 and H.2 provide detailed suggestions for feature selection, training instance selection and model aggregation.

*TableH.1: Suggestions for feature selection using statistical software*

| Feature selection activity | Step for activity | Implementation |
|---|---|---|
| Feature ranking | Generation of probe variables | SPSS, SAS or MS Excel |
| | Sampling | SPSS or SAS |
| | Binarisation of qualitative features and class variable | SPSS, SAS or MS Excel |
| | Measurement of class-feature and feature-feature correlations | Bivariate correlation matrix for quantitative variables |
| | | Pearson's chi-square, SU coefficient, phi and Cramer's V statistics |
| | Computation of mean and 95% CIs of means for correlations | SPSS |
| | Ranking and feature elimination using probes | SPSS or MS Excel |
| Feature subset search | Search for best subset | Specialised code e.g. C++ code |

*Table H.2: Suggestions for OVA and pVn modeling using statistical software*

| Activity | Implementation |
|---|---|
| Sampling for training set to create single model | SPSS or SAS |
| Creation of single model and confusion matrix | SPSS or SAS |
| Dataset partitioning | SPSS, SAS or SQL |
| Sampling from partitions to obtain boosted samples for base model creation | SPSS, SAS |
| Creation of base models | SPSS, SAS or other modelling software |
| Model aggregation | SPSS, SAS or MS Excel<br>or Specialised code e.g. C++ code |

# Appendix I

# Publications and conference presentations

LUTU, P. E. N. & ENGELBRECHT, A. P. (2006) A Comparative Study of Sample Selection methods for Classification. *South African Computer Journal,* 36, 69-85.

LUTU, P. E. N. & ENGELBRECHT, A. P. (2008) A decision rule-based method for feature selection in predictive data mining. Presentation at*: The 18th Triennial Conference of the International Federation of Operational Research Societies (IFORS 2008), Sandton, Johannesburg, July 2008*.

LUTU, P. E. N. & ENGELBRECHT, A. P. (2010) A decision rule-based method for feature selection in predictive data mining. *Expert Systems with Applications,* 37, 602-609.

LUTU, P. E. N. & ENGELBRECHT, A. P. (2010) Using OVA modeling to improve classification performance for large datasets. Submitted to the Journal of Expert Systems With Applications (ESWA).

LUTU, P. E. N. & ENGELBRECHT, A. P. (2010) An algorithm for combining K-Nearest Neighbour base model predictions. Submitted to the Journal of Expert Systems With Applications (ESWA).