2009

Investigating data mining techniques for extracting information from Alzheimer's disease data

Vinh Quoc Dang
Edith Cowan University

Edith Cowan University

Copyright Warning

Investigating data mining techniques for extracting information from Alzheimer's disease data

A dissertation submitted in partial fulfilment of the
requirements for the degree of
Bachelor of Computer Science Honours (Software Engineering)

By: Vinh Quoc Dang

Faculty of Computing, Health and Science
Edith Cowan University
Perth Western Australia

Supervisor: Associate Professor Chiou Peng Lam

Date of submission: 31st October 2009

Abstract

Data mining techniques have been used widely in many areas such as business, science, engineering and *more recently in clinical* medicine. These techniques allow an enormous amount of high dimensional data to be analysed for extraction of interesting information as well as the construction of models for prediction. One of the foci in health related research is Alzheimer's disease which is currently a non-curable disease where diagnosis can only be confirmed after death via an autopsy. Using multi-dimensional data and the applications of data mining techniques, researchers hope to find biomarkers that will diagnose Alzheimer's disease as early as possible. The primary purpose of this research project is to investigate the application of data mining techniques for finding interesting biomarkers from a set of Alzheimer's disease related data. The findings from this project will help to analyse the data more effectively and contribute to methods of providing earlier diagnosis of the disease.

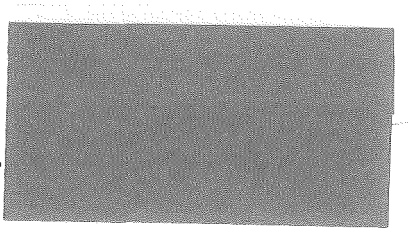
COPYRIGHT AND ACCESS DECLARATION

I certify that this thesis does not, to the best of my knowledge and belief:

(i) incorporate without acknowledgment any material previously submitted for a degree or diploma in any institution of higher education;

(ii) contain any material previously published or written by another person except where due reference is made in the text; or

(iii) contain any defamatory material.

Signed.. 

Dated..31/10/2009.....

USE OF THESIS

The Use of Thesis statement is not included in this version of the thesis.

Acknowledgements

Time flies so quickly since the first day I started my Honours thesis and now I would like to express my thanks and appreciations to all people who helped and inspired me throughout the Honours project.

I would like to thank my supervisor Associate Professor Chiou Peng Lam for sharing her knowledge in the area of Alzheimer's and data mining, and for guiding and supporting me throughout the entire course of the research. I really appreciate the time and energy she has dedicated to teaching and supervising me. Without her support and encouragement, this thesis could not be completed as it should have been.

I sincerely thank Associate Professor Craig Valli, Associate Professor Philip Hingston and Mr. David Cook for providing useful comments in my presentations. Grateful thanks to Dr. Judy Clayden for her comments and her help in my work. Special thanks to Dr. Martin Masek as an Honours coordinator for supporting and motivating me throughout the entire project. I also like to thank all my friends for their friendliness and cheerfulness towards me. A special thank you to my wife and children for their understanding, support and care for me over this time of my study.

Table of Contents

Table of Abbreviations

Abbreviation	Meaning
AD	Alzheimer's disease
ADA	Adaptive discriminant analysis
APP	Amyloid Precursor Protein
BDA	Biased discriminant analysis
BN	Bayesian network
BNCIT	Bayesian network classifier with inverse tree
BSS	Backward sequential selection
CSF	Cerebrospinal fluid
CT	Computerised Tomography
CV	Cross validation
DM	Data mining
EBI	Evolutionary biclustering
EM	Expectation-Maximization
FCM	Fuzzy C-Means
FDR	False discovery rate
FS	Feature selection
FSS	Forward sequential selection
GA	Genetic algorithms
LDA	Linear discriminant analysis
LMT	Logistic model tree
LWL	Locally weighted learning
MCI	Mild cognitive impairment
MLP	Multilayer perceptron
MRI	Magnetic Resonance Imaging
NAD	Non-Alzheimer's disease
NDC	Non-demented control
OD	Other dementia
OND	Other neurological disease

PAM	Prediction Analysis of Microarrays
PET	Positron Emission Tomography
PiB	Pittsburgh compound B
PS-1	Presenilin-1
PS-2	Presenilin-2
RA	Rheumatoid arthritis
SAM	Significance Analysis of Microarrays
SMO	Sequential minimal optimization
WEKA	Waikato Environment for Knowledge Analysis

1 Introduction

Dementia related diseases in general, and Alzheimer's disease (AD) specifically, are currently of great interest amongst various groups of researchers in the health areas. The number of people with AD has increased dramatically in the last century (The Medical News, 2007; Alzheimer's Disease International, 2008), probably owing to an increase in life expectancy as well as to the lack of effective treatments for AD. Patients with AD experience a progressive impairment of cognitive functions. The diagnosis of AD is difficult and is predominantly based on exclusion of other neurological illnesses. One of these difficulties is that symptoms associated with AD appear to develop only after substantial damage has occurred in the brain. Thus, a patient may have the onset of the disease for several years before receiving a diagnosis.

Currently, there is no cure and no early diagnostic tests that are definitive for this disease. The need to detect AD via an "*equivalent pregnancy test*" has been repeatedly stated in the literature (Trojanowskl, 2004, p. 32). The ideal diagnostic test is one that is inexpensive, has a high specificity and can be carried out as easily and accurately as a "*pregnancy test*"; enabling diagnosis as early as possible (Hooper, Lovestone & Sainz-Fuertes, 2008). In addition, such a test should have minimal side effects and methods of obtaining samples should be simple, non-invasive and cost effective. Although at this point in time, treatments only address the symptoms; it is still a matter of urgency that they are initiated as early as possible in the disease process, before neuro-degeneration becomes too severe.

An essential step in the development of approaches for early AD interventions involves the identification of a biomarker or a set of

biomarkers for early clinical diagnosis. The standardised definition of biomarkers is "*a characteristic that is objectively measured and evaluated as an indicator of normal biological processes, pathogenic processes, or pharmacologic responses to a therapeutic intervention*" (EverythingBio, 2001, p. 1).

Advantages associated with such biomarkers include:

- a diagnostic tool that aids clinicians in the early detection of the disease, hopefully before substantial neuro-degeneration has occurred. This implies that treatment and monitoring of AD patients can be provided early in the disease process, preventing devastating damage to the brain. This aspect will become increasingly important when a cure become available.
- ways for measuring progression of the disease and evaluating responses to clinical intervention.

Furthermore, a set of biomarkers should be as small as possible and still have the highest possible diagnostic reliability (i.e. sensitivity and specificity), as a set involving a large number of biomarkers will naturally increase cost and complexity.

Researchers have been developing various molecular tests and techniques (e.g. microarrays, mass spectrometry) to address the need of finding biomarkers and this has resulted in an exponential growth in data acquisition (Gilman, 2006). Traditional statistical approaches are not effective in analysing such data sets with a small number of samples that are characterised by a high-dimensional feature space. A fundamental problem in identifying biomarkers from this data involves how to systematically search for relevant features; to reduce the dimensionality of the data set to a small, yet highly reliable and discriminative subset that is representative. In order to effectively address this problem in a timely manner,

approaches involving data mining (DM) techniques (Witten & Frank, 2005) need to be developed and employed.

Following this direction, this research study will investigate the application of DM techniques to analyse multidimensional AD data, and to evaluate the results from these techniques, both in terms of the performances of the algorithms and the validity of the extracted information.

1.1 The background to the study

According to reports from The Medical News and statistics from Alzheimer's Disease International (2008), Alzheimer's Australia NSW (2009) and CSIRO (2008) the number of people having AD in the world and in Australia has increased considerably. In 1906, the first patient was diagnosed with AD by the physician Alois Alzheimer (Nuzzo, 2007). A century later, about 26 million people have AD in the world, and this number will grow to over 106 million by 2050 (The Medical News, 2007). In Australia, 234,640 people have dementia and it is the fourth highest cause of death in Australia after heart disease, stroke and lung cancer (Alzheimer's Australia NSW, 2009). About 80% of dementia in Australia is AD and the number of people with dementia will be more than 730,000 by 2050 (CSIRO, 2008).

AD is a progressive neurological disease and according to the Alzheimer's Disease Education and Referral (ADEAR) Center (2008), diagnosis of AD can only be confirmed after death by autopsy or by cerebral biopsy. Current diagnostic approaches include cognitive testing, neuropsychological testing, physical testing, analysing answers to questions relating to a patient's memory and medical history, and using advanced technology such as Magnetic

Resonance Imaging (MRI), Positron Emission Tomography (PET) and Computerised Tomograph (CT) scans. Existing biomarkers for AD (Hooper *et al*, 2008) falls mainly into:

- those associated with Cerebrospinal fluid (CSF)
  - total Tau protein measured in CSF
  - Beta-amyloid measured in CSF
  - Neural thread protein/AD7C-NTP measured in CSF and in urine.

  A reduced level of 42-amino-acid beta-amyloid and an increase level in tau protein in CSF are associated with AD patients (Jong, Jansen, Kremer & Verbeek, 2006).

- those associated with genetic traits
  - mutations in amyloid precursor protein (APP), presenilin-1 (PS-1) and presenilin-2 (PS-2) (in the case of Familial Alzheimer's disease – cited in Hooper) genes
  - Presence of Apolipoprotein E ε4 allele **may** lead to development of Sporadic AD

- those associated with neuro-imaging
  - volumetric MRI – brain volume; reduction to total volume and enlargement of the ventricular,
  - PET using Fluoro-deoxy-glucose – measuring a reduction in glucose metabolic rate in various parts of the brain
  - PET using radio-ligands (e.g. Pittsburgh Compound B (PiB) – monitoring amyloid plaques

Currently, there are no established blood-based biomarkers for diagnosing sporadic AD in clinical use.

Limitations of existing biomarkers for Alzheimer's (Hooper *et al*, 2008) are:

- Not absolutely discriminative for diagnosing AD because similar pattern findings can be associated with other types of dementia.
- High costs involved for individual or mass screening.
- In the case of CSF collection, the procedure is invasive and potentially risky.
- Neuro-imaging can be distressing to those with dementia.

Recently, Ray *et al* (2007) have developed a molecular test that involves detecting significant changes in the concentrations of signalling proteins in blood plasma. Using the technique, Significance Analysis of Microarrays (SAM) on the training set, 19 proteins were identified as being expressed significantly different from the others. A separate process, using the technique Prediction Analysis of Microarrays (PAM) on the training data set, was also carried out and they successfully identify a group of 18 signalling proteins (from 120 proteins). These proteins are a subset of the 19 proteins identified from SAM. The 18 protein signature can be used to classify AD samples from non-AD (NAD) samples.

Blood samples, as an alternative source for biomarker assessments, have the advantage of being easily obtained, inexpensive and are a relatively safe procedure unlike a lumbar puncture. Increasingly, blood-based biomarkers are being investigated but none has yet been sufficiently validated for clinical use. Currently, there is no single and simple test that gives a definitive diagnosis for AD. The current *gold standard* associated with diagnosis of AD usually involved a series of tests; involving psycho-metric tests, CSF-based testing and neuro-imaging (Hooper *et al*, 2008). The accuracy of diagnosis using current approaches is up to 90% (Alzheimer

Society, 2005; ADEAR, 2008). A recent Biopharm Reports (2009) identified 60 candidate AD biomarkers - resulting from a significant effort into research for diagnostic tests for AD. This report further stated that while new and effective diagnostic tools and treatments for AD are urgently needed, advances in their development remain elusive.

Mass throughput techniques in molecular biology (e.g. gene expression microarrays and high resolution mass spectrometry) can produce enormous amounts of multi-dimensional data sets about cellular functions. Examples of such data sets include gene arrays with up to 500,000 genes and mass spectrometry data with 300,000 values (Aliferis, Statnikov & Tsamardino, 2006). While the availability of such data sets can aid in the development of techniques/drugs that will improve diagnosis and treatment of diseases, a major challenge involves its analysis to extract useful and meaningful information. High dimensionality and limited sample sizes contribute to problems such as over-fitting and in building predictive models.

The problem with having massive amount of high-dimensional data is the resulting increase in the complexity of analysis, as the structure and relationships amongst the data become a lot harder to understand and analyse (Witten & Frank, 2005). Traditional data analysis tools such as database query and statistics have problems in dealing with data sets that have a large number of variables (i.e. high dimensionality) and non-traditional data types (Gilman, 2006). These characteristics are typically found in biological data sets involving millions of variables (e.g. MRI data (Chen & Herskovits, 2005)). In addition, with the introduction of techniques like microarrays, data are generated on a massive scale, of an order previously thought to be impossible. Examples of biological

databases included GenBank, SWISS-PROT for protein sequences (Bertone & Gerstein, 2001) and ADNI for AD (ADNI, 2007). In order to overcome the problem of finding trends and patterns hidden in enormous amount of high dimensional data, new approaches such as DM can be used (Witten & Frank, 2005) as traditional statistical tools are inadequate. New approaches are required for a comprehensive, systematic and valid analysis – to address, on one hand the large volume of data that are generated and on the other hand, the nature of the data (i.e. high dimensionality and small sample size).

DM automates the process of going through a vast amount of high dimensional data to extract useful and relevant information such as correlations and patterns. DM is different from statistics as it supports the automation of statistical processes involving several techniques (from database technology, machine learning, statistics, etc). A hypothesis is formed and tested against data in statistical inference (i.e. assumption driven) where as DM is data discovery driven -- the hypothesis is automatically generated from the given data sets. It has been traditionally used in many areas such as business, science, engineering and in recent times, increasingly applied in the area of bioinformatics such as microarray analysis for classification, gene selection and clustering (Gilman, 2006). In terms of its use specifically in exploring AD related data, researchers like Ray *et al*, (2007), Chen & Herskovits (2005) and Walker *et al* (2003) have successfully used various approaches involving DM techniques to find interesting patterns. Drawing from these early successes, researchers hope to continue to apply DM techniques for exploration of high dimensional data sets. The aim is to identify other potential biomarkers for the detection of the disease as early as possible as well as to develop strategies to better manage and treat AD patients.

1.2 Significance of study

As mentioned previously, the number of people with AD on a global scale has increased dramatically, and especially in Australia the number of deaths resulting from AD has moved it from the 7th to 4th (Alzheimer's Australia NSW, 2009) leading cause of death. The rate of people contracting the disease is increasing and it is projected that "*by 2050,… 1 in 85 persons worldwide will be living with the disease*" (Brookmeyer, Johnson, Graham & Arrighi, 2007, p. 2). Therefore this research is important and significant because it will contribute to a better understanding of the applications of DM techniques and the effectiveness of its applications in the AD area. The findings from this study will provide insights into ways of improving/refining future analysis.

Finding relevant biomarkers from the AD data set used in this project can contribute to the development of a method for an earlier diagnosis of the disease. An early diagnosis is a critical factor because once people have been diagnosed as "*probable AD*" then their subsequent life expectancy can be as short as 3-4 years (ADEAR center, 2008, p. 3). In addition, the earlier the confirmation of an AD diagnosis, the greater is the benefits in managing its symptoms through the use of pharmaceuticals.

According to Brookmeyer and his co-authors " *If interventions could delay both the disease onset and progression by a modest 1 year, there would be nearly 9.2 million fewer cases of disease in 2050 …"* (Brookmeyer *et al*, 2007, p.4).

Existing AD diagnostic techniques are either highly invasive or expensive and the development of techniques for blood-based

molecular biomarkers is very important as this can provide a simple test, an equivalent "*glucose test commonly used to identify diabetic condition*", for AD. This study is significant as it attempt to identify biomarkers associated with signalling proteins in blood samples via DM techniques. In addition, a diagnostic tool that requires a large number of biomarkers would increase cost as well as complexity. This study will also re-evaluate two existing studies, with the aim of investigating approaches for further reducing the number of biomarkers involved and yet, still maintained its classification capability.

1.3 The purpose of the study

The aims of the project are

- to investigate and to evaluate a number of DM techniques and software tools for analysing AD related data to find biomarkers that can be incorporated in a diagnostic tool for AD.
- to apply a number of DM techniques on an AD related data set and to extract relevant patterns.
- to evaluate the validity of the extracted information from the DM perspectives.
- to carry out an in-depth study of two existing studies (Ray *et al*, 2007; Ravetti & Moscato, 2008) and to explore the possibility of improving the existing results.
- to investigate the applications of different feature selection (FS) techniques and their impact on the subsequent classification accuracy.

1.4 Research questions

1.4.1 The main question that this research needs to answer is how can DM techniques be used effectively to analyse AD data for extraction of relevant information?

In order to answer the main research question, two sub-questions need to be answered as follows:

1.4.1.1 Can the experiment results associated with two case studies (listed below) be re-produced using the information provided in their respective papers, and can we further improve upon these results?

- Classification and prediction of clinical Alzheimer's diagnosis based on plasma signaling proteins (Ray *et al*, 2007).

- Identification of a 5 protein biomarker molecular signature for predicting AD (Ravetti & Moscato, 2008).

1.4.1.2 What is the impact on the accuracy of the classification models generated using features obtained from different FS methods and used on the same dataset?

1.5 Contributions of this study

The contributions of this study are:

- An approach for systematic exploration of using DM techniques for mining interesting information from high dimensional AD data.
- Carried out an in-depth study of Ray *et al* experiments involving SAM and PAM, providing more details and a finding of a 17 protein biomarker signature that can be used to produce a classifier with similar classification accuracy.
- Carried out an in-depth study of the Ravetti and Moscato's experiments and subsequently obtained a 4 protein biomarker signature. The 4 proteins is a subset of the 18 protein signature from the Ray *et al* experiments. 23 Waikato Environment for Knowledge Analysis (WEKA) classifiers were trained using this 4 protein biomarker signature and the overall classification results on the test data were similar to the results of the 5 protein biomarker signature in Ravetti and Moscato's experiments. The finding is significant as having a smaller signature may lower the cost and complexity in making a diagnosis.
- Produced initial findings associated with the use of different FS techniques and its impact on the accuracy of the resulting classifiers.

1.6 Structure of the thesis

This thesis consists of 6 chapters.

Chapter 1 provides a brief snapshot of the current status of AD; number of people having the disease is increasing dramatically and there is no early, simple and definitive test to diagnose it so far, only invasive and expensive methods are available for identifying people inflicted with the disease. The chapter also provides insights to the problems of high throughput technologies which generate tremendous amount of data, requiring new analytical approaches. DM techniques for analysing these data effectively is one possible solution that has been proposed. The significance and purpose of this study as well as research questions to be address are also clearly stated in this chapter.

Chapter 2 describes a literature review, providing information about DM techniques and tools with a focus on those that have been previously used in bioinformatics. The chapter also examines existing work that have employed DM techniques for extracting information from health-related data in general and AD related data specifically. The remaining sections of this chapter detailed concepts related to FS.

Chapter 3: This chapter provides general information of the data set and research method used for this study. The data format of training and test data sets were described in details. Limitations of the study were also stated in this chapter.

Chapter 4: This chapter described an investigation to address the first aim of this study. Detailed descriptions associated with the in-depth study of the work of Ray *et al* and Ravetti and Moscato's experiments are provided here. The chapter also provides information related to the experiment procedures and analysis in obtaining the 4 protein biomarker signature.

Chapter 5: This chapter detailed experimentations associated with the investigations of using different WEKA FS methods for generating relevant biomarker signatures. The aim is to investigate the impact of using different FS methods on the accuracy of the resulting classifiers that were trained using these sets of features signatures. The analysis is based on evaluating the performance of the resulting classifiers on the test sets. The chapter provides information of the experiment to be carried out as well as the results of these experiments.

Chapter 6: This chapter is used to outline the conclusions of this thesis, highlight things that have been achieved, and discuss future work.

1.7 Definitions of terms (Witten & Frank, 2005 and Tan, Steinbach & Kumar, 2006)

- Alzheimer's disease (AD): a type of progressive non-curable neurological disease.
- Amyloid precursor protein (APP): *"is found in many tissues and organs, including the brain and spinal cord"* (Genetics Home References, 2008, p. 2).
- Association: in DM, it is a method of finding relationships between attributes in databases.

- Biclustering: is a DM algorithm which performs clustering on both dimensions of data (rows and columns) at the same time.
- Classification: in DM, it is the task of assigning attributes to a predefined class.
- Classification model: is known as a target function. It is used for predicting or describing a class.
- Clustering analysis: in DM, it is a technique which groups data with similar features into clusters.
- CT: Computerised Tomography is "*a diagnostic procedure that uses special X-ray equipment to create cross-sectional pictures of your body*" (Medline Plus, 2009, p. 1). It is used to produce images of the internal body.
- Data: raw facts, numbers, letters and outputs from devices are collected and used for producing information.
- Database: storage of data where data are stored and related in specific structures in a computer.
- Data mining (DM): process of retrieving useful information from a large amount of data.
- Dementia: a progressive impairment of cognitive functions.
- MRI: Magnetic Resonance Imaging - a technique that is used mainly in the health area to scan patients for an analysis view of tissues.
- MCI: Mild Cognitive Impairment – an early stage of AD.
- Microarray: is an array that consists of many tiny dots of DNA, protein or tissues arranged in order on a small piece of glass.
- NDC: Non-demented control, used as controllers for people without dementia disease.
- PAM: Prediction Analysis of Microarrays – a software program that is used to classify microarray data.

- PET: Positron Emission Tomography is a "*nuclear scanning uses radioactive substances to see structures and functions inside your body"* (Medline Plus, 2009, p. 1).
- Presenilin-1 (PS-1): in human, it is a protein that is encoded by the PS1 gene.
- Presenilin-2 (PS-2): in human, it is a protein that is encoded by the PS2 gene.
- SAM: Significance Analysis of Microarrays – a software program that is used to analyse significant differences in expressions of genes or signalling proteins.
- Supervised learning: a type of DM techniques that uses training data to train the algorithm and to generate a classification model.
- Test data: data that are used to test an algorithm for accuracy.
- Training data: data that are used to train a supervised learning algorithm.
- Unsupervised learning: a type of DM techniques that does not need to use training data to train the algorithm. No prediction involved.
- Variable: an attribute or a feature of data
- WEKA: Waikato Environment for Knowledge Analysis – is a DM software program to analyse data.

1.8 Summary

This chapter provides the background and highlights some important aspects that lead to the use of DM techniques as necessary tools to analyse AD data. In the next chapter, DM tools in bioinformatics, classification and clustering techniques, as well as FS techniques are described.

2. Review of the literature

This chapter briefly describes DM techniques that are applicable to the project and briefly reviews existing work involving the use of DM techniques in extracting interesting information from health-related data in general and AD data specifically. Typical characteristics associated with datasets in the biomedical area are very high dimensional relative to sample size, posing a challenge in terms of the application of traditional statistical approaches.

Many DM techniques are available and in general, they are categorised in three types of DM techniques: supervised learning, unsupervised learning and association analysis. In this study, only supervised and unsupervised are discussed because they are used to analyse the AD data.

2.1 Data mining techniques

2.1.1. Supervised learning analysis

2.1.1.1. Classification

Classification analysis is an example of supervised learning. Using a set of input data as examples, the algorithm learns a mapping between the attributes and the designated class label assigned by the user. This mapping function is known as the classification model and can be used for predictive modeling and descriptive modeling.

Consider the vertebrate data set below:

Table 2-1: The Vertebrate data set, adapted from Tan, Steinbach & Kumar (2006, p.106)

Name	Body temperature	Skin cover	Gives birth	Aquatic creature	Aerial creature	Has legs	Hibernates	Class label
human	warm blooded	hair	yes	no	no	yes	no	mammal
python	cold blooded	scales	No	no	no	no	yes	reptile
salmon	cold blooded	scales	No	yes	no	no	no	fish
bat	warm blooded	hair	yes	no	yes	yes	yes	mammal
: : :	: : :	: : :	: : :	: : :	: : :	: : :	: : :	: : :
turtle	Cold blooded	scales	no	semi	no	yes	no	reptile
cat	warm blooded	fur	yes	no	no	yes	no	mammal

The vertebrate data set consists of names of animals with their attributes and designated class labels as shown in Table 2.1. The data types associated with the attributes could be discrete or continuous. However, the class label must be a discrete variable. This data set is considered as a training data set. The classification algorithm uses the training data set to learn the association between each animal, with its attributes and the class to which it belongs to. The classification model is then generated and subsequently used on a test data set (data that has not been used previously in the training) to evaluate the accuracy of the model before putting it in use to predict or to describe other unknown

creatures (Tan *et al*, 2006). Table 2-2 shows an example of a test data.

Table 2-2: The Vertebrate test data, extracted from Tan *et al* (2006, p.106)

Name	Body temperature	Skin cover	Gives birth	Aquatic creature	Aerial creature	Has legs	Hibernates	Class label
gila monster	cold blooded	scale	no	no	no	yes	yes	?

The above table shows the features of a gila monster. Its class is normally assigned a class label prior to prediction by the classifier. However in the table above, it is labelled with a question mark (?) with the purpose of indicating that the class label will be predicted by the classifier. Therefore the classification model needs to analyse the features of a gila monster, and based on what has been learned from the training data set, the classification model predicts the class of the creature to which the gila monster belongs. The performance of the model is measured by the accuracy of its prediction. The less error and the higher the accuracy rate in prediction, the better and more reliable, is the model.

The process of supervised learning associated with classification analysis may be illustrated by the following diagram:

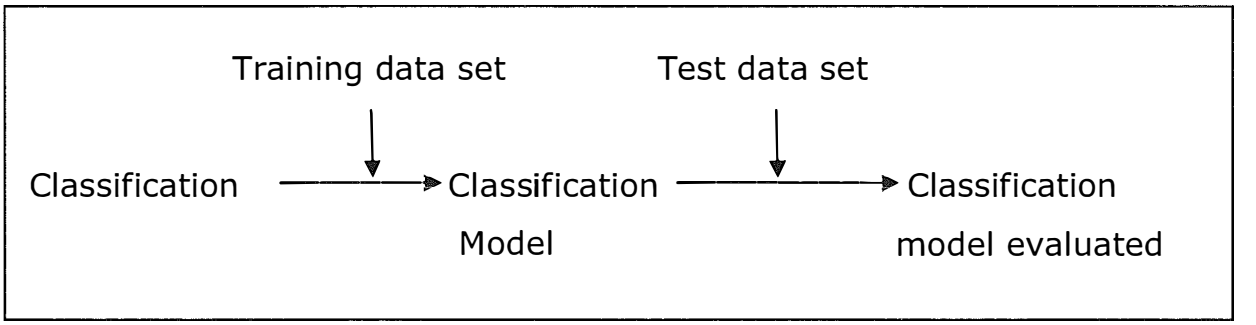


Figure 2-1: The process of supervised learning in classification analysis

When a classification model has been built, it needs to be evaluated to see whether it performs accurately and gives a correct classification result for a specific unseen data point. According to Tan *et al* (2006) this can be done by using a confusion matrix table to record the number of correct and incorrect outcomes and then calculate the accuracy and error rate by the following formulae:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

$$\text{Error rate} = \frac{\text{Number of wrong predictions}}{\text{Total number of predictions}}$$

Classification methods are most appropriate for problems where the data sets are associated with binary or nominal data types. They do not perform so well when ordinal data types or relationships such as

"*superclass-subclass*" (Tan *et al*, 2006, p. 107) differentiation are involved.

Decision trees, Artificial Neural Networks (ANN) and Bayesian classifiers are examples of classification techniques.

2.1.1.2. Problems with Classification techniques

According to Tan *et al* (2006), two problems are associated with classification.

- A problem associated with classification techniques is missing values that happen when a certain data point collected is missing from a data set. The problem causes the algorithm to be confused when deciding which way to go if a node has a missing value. In order to overcome this problem, many solutions have been proposed in the literature. Example solutions assume the missing value to be just another possible value of the node or to omit the record with the missing value.

- Another problem associated with classification techniques is model overfitting which happens when the classification has a low training error rate and a high test error rate, while model underfitting has a high rate for both types of errors. Overfitting is caused by mislabelled data (noise) or lack of training records. Two approaches used to overcome overfitting are pre-pruning (forward pruning) and post-pruning (backward pruning) (Han & Kamber, 2006; Tan *et al*, 2006; Witten & Frank, 2005)

    - With the pre-pruning approach, the tree is stopped early before reaching its full development. This would reduce the number of sub-trees created thus making the tree less

complex and hence not causing the problem of overfitting the training data.

- With the post-pruning approach, the tree is first created completely and then is trimmed from bottom-up by using a class label node to replace a sub-tree. Figure 2-2 shows the sub-tree of node A3 in a fully unpruned tree which was then pruned and replaced by Class B, with the assumption that of the class B is the most common class for this sub-tree (Han & Kamber, 2006).

The Post-pruning approach gives better results than the pre-pruning approach because the trimming of the tree is based on a fully created tree. In building a fully created tree, all possible combinations of features are constructed as part of the tree. However, this extra computation in building a complete tree could be wasted especially when infeasible subtrees are pruned (Witten & Frank, 2005; Tan *et al*, 2006).

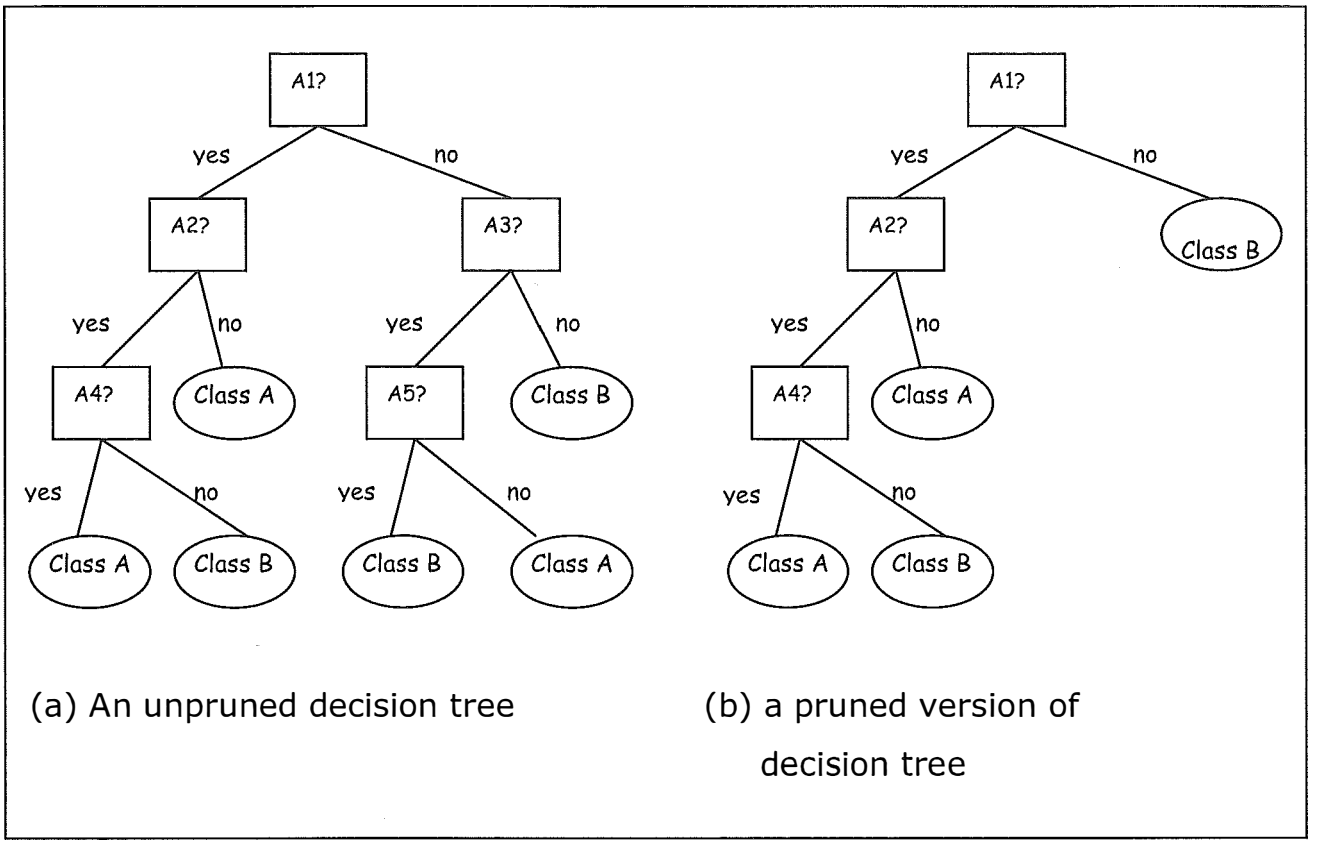


Figure 2-2: An unpruned and pruned decision tree, extracted from Han & Kamber (2006, p. 305)

2.1.1.3. Evaluation of a classifier

According to Han and Kamber (2006) and Tan *et al* (2006), the performance of a classifier is evaluated by using the holdout, random subsampling, cross-validation and bootstrap techniques.

- Holdout technique: the data set is divided into training data and test data. The training data is used to build a classification model and the test data is used to evaluate the accuracy of the classification model.
- Random subsampling: is based on the holdout technique but the process is repeated a number of times. The accuracy of the classifier is then based on the average value of the iterations.

- Cross-validation: the data set is divided into a number (k) of subsets. For each run one subset is used as test data and the others for training data. This procedure is repeated a number (k) of times so that every subset has been used as test data once. The total error is calculated by adding up all errors generated by all the runs.
- Bootstrap: "*In the approach, the training records are sampled with replacement; i.e., a record already chosen for training is put back into the original pool of records so that it is equally likely to be redrawn.*" (Tan *et al*, 2006, p.137)

2.1.1.4. Comparison of classifiers

According to Han and Kamber (2006), the comparison of the classifiers is based on their accuracy, speed, robustness, scalability and interpretability.

- Accuracy is measured based on the level of correction of a classifier for predicting a given unknown record data. The accuracy measures can be carried out by using different techniques such as cross validation and bootstrapping as described in the section 2.1.1.3.
- Speed is measured based on the time required by a classifier to generate the result. This involves the complexity of the computational algorithm.
- Robustness is measured based on how well a classifier can predict in the cases where a given data set is noisy or have missing values.
- Scalability is measured based on how well a classifier is able to deal with a large data set.
- Interpretability is measured based on "*the level of understanding and insight that is provided by the classifier or*

predictor" (Han & Kamber, 2006, p. 291). – how easy it is to understand the results returned by the classifier.

2.1.2. Unsupervised learning analysis

Clustering analysis is an example of unsupervised learning (Groth, 2000) as the training data is not labelled. The technique does not involve training and clustering depends upon whether the data are related or unrelated to each other. Clustering analysis is a method or process of separating data in a data set into different groups based on some measures of similarities of their features or functions. Each cluster contains data points that are similar to each other on the basis of a defined measure. Commonly used measures of similarity are distance measures such as Euclidean distance, Hamming distance and supremum distances (Tan *et al*, 2006). The ideal result from clustering analysis is to have groups that are completely different to one another. The more different the groups, the easier it is to distinguish between them, the better the understanding of the relationships between data and the stronger the conclusions that may be drawn from this process. Clustering analysis can be used in many areas such as biology (eg. groups of genes with similar functions), medicine (eg. different types of diseases), climate, information retrieval and business (Tan *et al*, 2006).

Categories of clustering algorithms include: Exclusive Clustering, Hierarchical Clustering, Overlapping Clustering and Probabilistic Clustering (Matteucci, n.d.).

2.1.2.1. Exclusive Clustering

In exclusive clustering, a data point can only belong to one cluster, as shown in Figure 2.3. The following diagram illustrates an example of the results from a clustering analysis technique used in business:

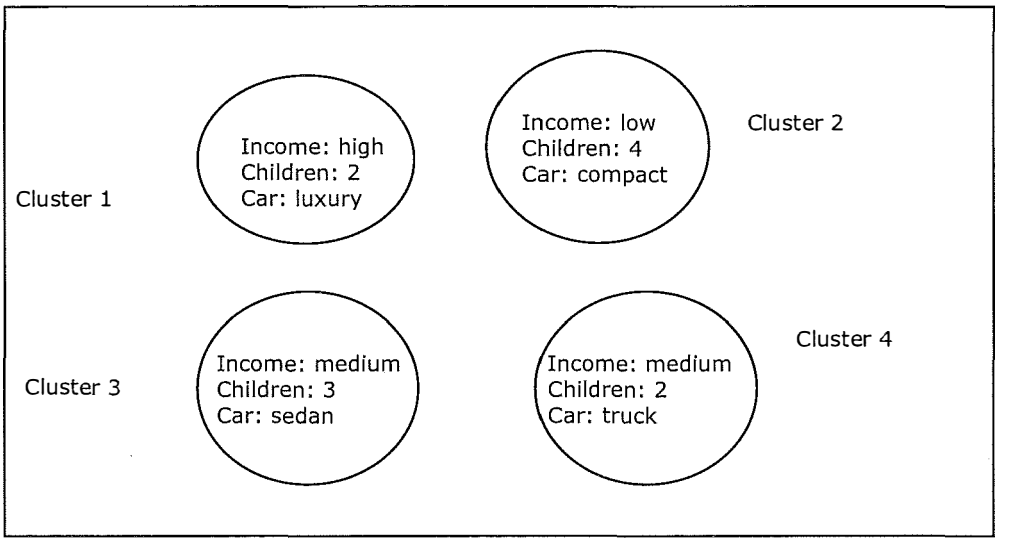


Figure 2-3: Customer are clustered into four segments, adapted from Groth (2000, p. 25)

An example of this type of clustering algorithm is the K-means algorithm. According to Tan *et al* (2006), the K-means algorithm requires a number of initial target clusters centroids to be specified at the start of the algorithm and then each data point is grouped into one of the clusters defined by each of these centroids. The location associated with each centroid is then recalculated. The process of grouping data points into the clusters continues until all the data points are grouped.

According to Tan *et al* (2006), K-Means clustering has the following advantages and disadvantages:

Advantages:

- Simple, efficient, widely used in many applications and works well with many different data types.

Disadvantages:

- Clustering data with different sizes and densities can cause the K-means problems in finding sub-clusters.
- Clustering data with outliers can cause the K-means problems in specifying cluster centroids and obtaining desired clusters.

2.1.2.2. Hierarchical Clustering

The data is normally clustered as a set of nested cluster in a tree structure. Except for the leaf nodes, each node is formed from merging its children. According to Tan *et al* (2006), hierarchical clustering works by calculating the distance between 2 clusters and then joining the 2 nearest clusters together to form a new cluster. The algorithm then recalculates the distance between the newly form cluster and the original clusters. The process of joining and re-calculating continues until there is only one cluster left over.

According to Tan *et al* (2006), hierarchical clustering has advantages and disadvantages:

Advantages:

- It is useful for hierarchical applications.
- Clusters generated are more reliable and correct.

Disadvantages:

- Once clusters have been combined together, that action cannot be reversed and may cause data to become noisy and extra but unnecessary dimensions of data to be created.
- The algorithm is computationally more expensive and a higher memory requirement.

2.1.2.3. Overlapping Clustering

A data point may belong to more than one cluster as fuzzy sets are used to determine the clusters. Example of an algorithm in this category is Fuzzy C-means clustering (Matteucci, n.d.).

The Fuzzy C-Means (FCM) clustering algorithm has advantages and disadvantages as follows:

Advantages:

"*… membership function and an object can belong to several clusteres [sic] at the same time but with different degrees. This is a useful feature for a facility location problem.*" (Zalik, 2006, p. 1)

Disadvantages (Amiri, 2003):

- Computations take more time to perform.
- Easily effected by noise

2.1.2.4. Probabilistic Clustering

Clusters are generated using a probabilistic approach via algorithms such as the Mixture of Gaussian technique (Matteucci, n.d.). According to Borman (2004, 2009) and Dellaert (2002), the

Expectation-Maximization (EM) algorithm is commonly used in this type of model-based clustering. This algorithm has two steps: Expectation step (E-Step) and Maximization step (M-Step).

> "In the Expectation or *E-step*, *the missing data are estimated given the observed data and current estimate of the model parameters*. … I*n the M-Step*, *the likelihood function is maximized under the assumption that the missing data are known*. *The estimate of the missing data from the E-step are used in lieu of the actual missing data*." (Borman, 2004, 2009, p. 5)

Advantages:

- Be able to identifies various sizes and shapes of clusters;
- "*provides a disciplined way of eliminating some of the complexity associated with data*" (Tan *et al*, 2006, p. 445).

Disadvantages:

- Slow, not handle well the data with a small number of samples but having a large number of features;
- Not easy to deal with noise (Tan *et al*, 2006).
- Problems with constraints where 2 clusters have the same equal probability but one belongs to both clusters and the other belongs to one cluster only, subsequently reducing the accuracy of the classification (Bodyanskiy, n.d.)

2.1.2.5. Evaluation of Clusters

A key concept in the development of a clustering algorithm is the measure of similarity, commonly it is some form of a distance measure between data points (Matteucci, n.d.). After a data set has been clustered, results of the clusters need to be evaluated or

measured in order to find out whether the outcome derived from the data analysis is correct and valid, and that the employed technique is reliable and suitable for clustering a particular data set. According to Tan *et al* (2006), the three categories of clustering validation are unsupervised measures, supervised measures and relative measures.

- Unsupervised measures are based on cohesion and separation of clusters. Cluster cohesion measures the "closeness" relationship between data within a cluster. Cluster separation measures the degree of separation of the data between clusters.

The following diagrams illustrate a cluster cohesion and separation:

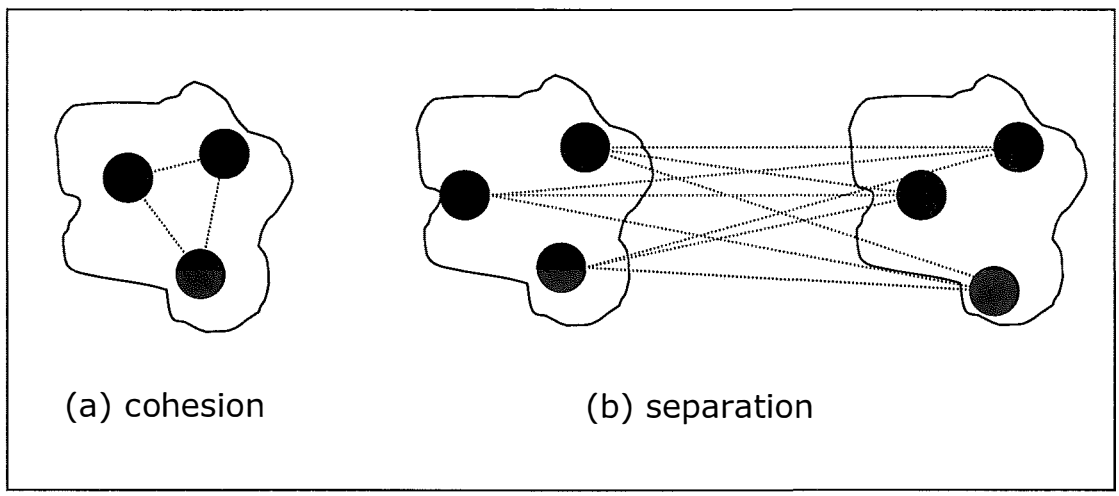


Figure 2-4: graph based view of cluster cohesion and separation, extracted from Tan *et al* (2006, p. 402)

- Supervised measures are based on
    - classification-oriented measures (with different types of measures such as entropy, purity, precision, recall and F-measure) to measure whether the class to be predicted matches up to the training class in the data set.

- Similarity-oriented measures to compare the ideal cluster similarity matrix and the ideal class similarity matrix.

- Relative measures can be either unsupervised or supervised measures and they are used for comparison of different clusters.

2.1.2.6. Problems with clustering

According to Matteucci (n.d.), some problems associated with clustering techniques are:

- *"Current clustering techniques do not address all the requirements adequately (and concurrently);*
- *Dealing with large number of dimensions and large number of data items can be problematic because of time complexity;*
- *The effectiveness of the method depends on the definition of "distance" (for distance based clustering);*
- *If an obvious distance measure doesn't exist we must "define" it, which is not always easy, especially in multi-dimensional spaces;*
- *The result of the clustering algorithm (that in many cases can be arbitrary itself) can be interpreted in different ways."* (Matteucci, n.d, p. 5)

2.2 Data mining tools in bioinformatics

High throughput technologies in bioinformatics such as microarrays, MRI and PET scanning have been developed recently. These technologies have been used to generate a large amount of data with high dimensions and complexities. This challenges researchers and scientists in the area of bioinformatics as to how to make good use of these data and how to analyse them effectively. They require analytical techniques and tools that are capable of handling data sets which have only a small number of samples and yet at the same time, is of very high dimensionality (e.g. gene arrays with up to 500,000 genes and mass spectrometry data with 300,000 values (Aliferis, Statnikov & Tsamardino, 2006)). Many DM software programs have been developed and used widely in many areas to analyse and extract useful information from the data. Recently, more attention has been given to using DM in the area of health and AD, with the aim of analysing the data to find biomarkers that can be used in diagnosing the disease as early as possible, so that better management treatments can be provided to patients. Table 2-3 provides a summary of some of the many DM software packages that are currently used for mining biomedical data.

Table 2-3: DM software programs for mining medical data, adapted from Levy, Statnikov & Aliferis (2005, p. 6-7); Chu, Narashihan, Tibshirani & Tusher (2001, p. 14); WEKA 3.6.1 (2009); Tanagra 1.4.31 (2003)

Name	***Supervised Classification***	***Cross-validation for performance estimation***
Avadis Prophetic	*Decision trees* *Neural networks* *Support vector machines (SVM)*	*yes*
BRB ArrayTools	*Nearest centroid* *K-Nearest Neighbor* *SVM* *Compound Covariate Predictor* *Diagonal Linear Discriminant*	*yes*
caGEDA	*Nearest Neighbor methods* *Naïve Bayes classifier*	*Yes*
GeneCluster 2	*Weighted voting* *K-Nearest Neighbor*	*yes*
GenePattern	*Weighting voting*	*yes*
GeneMaths XT	*Neural network* *K-Nearest neighbours* *SVM*	*yes*
Genesis	*SVM*	*No*
GeneSpring	*K-Nearest neighbours* *SVM*	*Yes*
PAM	*Nearest shrunken centroids*	*yes*
SAM	Unsupervised classification: K-Nearest neighbours	Yes

Tanagra	C.45 Multi-layer perception Logistic regression Naïve Bayes classifier Meta-learning techniques (Boosting, Bagging)	Yes
WEKA Explorer	*K-Nearest neighbours* *Decision trees* *Rule sets* *Bayesian classifiers* *SVM* *Multi-layer perceptron* *Linear regression* *Logistic regression* *Meta-learning techniques* *(Boosting, Bagging)* *Decorate* *Multi-class classifiers*	*yes*

The following sections describe DM software and techniques used in this study.

2.2.1 Significance Analysis of Microarrays (SAM) program

SAM is an unsupervised learning analysis program that can be used to find significant difference in gene expressions. SAM technique was initially proposed by Tusher, Tibshirani and Chu (2001), and the SAM software program was created by Narasimhan and Tibshirani and distributed by Stanford University (Chu *et al*, 2001);). SAM is a MS-Excel plugin and it is a free licenced program.

According to SAM user guide and technical document (Chu *et al*, 2001), input and output data formats for SAM are as follows:

2.2.1.1 Input data

- The first row of the dataset contains response measurements, one per column, starting at column 3.

- The remaining rows of the dataset contain gene expression measurements one line per gene with the following format:
    - Column 1 for gene Name
    - Column 2 for gene ID
    - Remaining column for gene expression measurements in numeric values.

2.2.1.2 Output data

- SAM plot shows positive significant genes (red colour) and negative significant genes (green colour) on the plot.

- SAM output lists all positive and negative significant genes in the following format:
  Row number, gene name, gene ID, SAM score, denominator ($S + S_0$), q value and local False Discovery Rate (FDR)

2.2.1.3 Algorithms

SAM uses the following algorithm to find genes with significant differences in expression.

1. *Calculate a statistic score (d):*

$$d_i = \frac{r_i}{s_i + s_0}$$

i=1,2..,p
r_i is a score
s_i is a standard deviation
s_0 is exchangeability factor

2. *Calculate the order of score(d): $d1 \leq d2 \leq dp$*
3. *Take B set of permutations of the response value y_i.*
   For each permutation b
      Calculate d_i^{*b}
      Calculate order of $d_1^{*b} \leq d_2^{*b} \ldots \leq d_p^{*b}$
   Estimate the order of score(d) $d_i = (1/B) \sum_b d_i^{*b}$
4. *Plot d_i values against the $\overline{d_i}$*
5. *For a fixed threshold Δ, start at the origin*
   Move up to the right
      Find first $i=i_1$, where $d_i - \overline{d_i} > \Delta$
      Label all genes past i_1 as significant positive genes
   Move down to the left
      Find first $i = i_2$ where $\overline{d_i} - d_i > \Delta$
      Label all genes past i_2 as significant negative genes
   For each Δ
      Compute cut_{up} Δ
      Compute cut_{low} Δ
6. *For all Δ values*
   Compute the total number of significant genes

Compute median number of falsely called genes

7. *Estimate the proportion of unaffected genes*
8. *List significant genes with specified Δ*
9. *Compute False Discovery Rate (FDR)*
10. *Compute q-value*
11. *Compute local FDR*

Algorithm 2-1: SAM algorithm for finding significant genes in a data set, adapted from Chu *et al* (2001, p. 27-29) and Wie (n.d., p.2)

SAM handles missing data from a data set by using a k-Nearest Neighbor algorithm. The following are steps for this algorithm:

1. *For each gene i having at least one missing value*
   a. *Let S_i be the samples for which gene i has no missing values.*
   b. *Find k nearest neighbor to gene, using only sample S_i to compute the Euclidean distance. When computing the Euclidean distance, other genes may have missing values for some of sample S_i; the distance is averaged over the non-missing entries in each comparison*
   c. *Impute the missing sample values in gene I, using the averages of the non-missing entries for the corresponding sample from the k nearest neighbors.*
2. *If a gene still has missing values after the above steps, impute the missing values using the average (non-missing) expression for that gene.*

Algorithm 2-2: k-nearest neighbour algorithm, extracted from Chu *et al* (2001.), p14-15)

2.2.2 Prediction Analysis of Microarrays (PAM) program

PAM is a supervised learning analysis program that was created by Tibshirani, Hastie, Narasimhan and Chu (2002) at Stanford University. PAM can be used for classification, survival analysis and regression tasks, on gene expression data using the nearest shrunken centroid algorithm. PAM is a MS-Excel plugin and it is a free licenced program. According to PAM user guide and manual (Hastie, Narasimhan, Tibshirani & Chu, 2002), input and output data formats for PAM are as follows:

2.2.2.1 Input Data

Input data format for PAM is similar to SAM but PAM can be used to handle a standard classification problem, survival analysis and regression. Therefore input data format for each type of classifications are different as follows:

- Classification:
  - The first three rows of the dataset consist of class labels (required), sample and batch labels (optional).
  - The remaining rows of the dataset contain gene expression measurements one line per gene with the following format:
    - Column 1 for gene Name
    - Column 2 for gene ID
    - Remaining column for gene expression measurements in numeric values.

- Survival analysis and regression:
  Data format is similar to classification. But for survival analysis, class labels are replaced by Survival time and Censoring status label, and for regression, class labels are replaced by Outcome labels.

- PAM handles missing data from a dataset by using a k-Nearest Neighbor algorithm. The algorithm is the same as the one used in SAM.

2.2.2.2 Output Data

- PAM plots: plots for training errors, test errors, cross validation (CV), CV probabilities, shrunken centroids and test probabilities.

- PAM output: a list for a subset of significant genes in the following format:
  Gene Name (column 1), ID (column 2) and following columns for centroid scores of all the classes specified in the data set.

- Prediction output: a confusion matrix that lists all actual and predicted class labels with their prediction probabilities.

2.2.2.3 Algorithms

PAM uses the nearest shrunken centroid as a technique for classification. The nearest shrunken centroid algorithm is described as follows:

1. *Let x_{ij} be expression for genes i=1..p and samples j=1..n*
2. *Let C_k be indices of n_k samples in class k*
3. *Calculate the mean expression value in class k for gene i*
4. *Calculate i^{th} component of the overall centroid for class k*

   $x_i = \sum^{n}_{j=1} x_{ij}/n$
5. *Let* $d_{ik} = (\overline{x_{ik}} - \overline{x_i})/s_i$
6. *Shrink each d_{ik} towards 0:*

   Apply K-fold cross validation to select Δ

   Calculate shrinkage by soft thresholding

   $D'_{ik} = \text{sign}\,(d_{ik})(|d_{ik}| - \Delta)_+$ where + means positive part
7. *Calculate new shrunken centroid*

   $\overline{x}'_{ik} = \overline{x}_i + s_i d'_{ik}$

Algorithm 2.3: PAM algorithm, adapted from Prediction Analysis of Microarrays user guide and manual (Hastie *et al* (2002), p. 33-37); Supervised learning (Hastie & Tibshirani, 2002, p. 32-33))

2.2.3 WEKA program

According to KDNuggets (2006), 28 DM software programs were used in 2005 and one of them was the WEKA program, which was ranked 11th. WEKA has been developed and maintained since 1994 by the WEKA team (Markov & Russell, 2006). WEKA consists of learning algorithms that can be used to analyse data sets and to predict new classes (Witten & Frank, 2005). It provides three types of graphical user interfaces (refer to APPENDIX B for screen shots of WEKA interfaces)

- Explorer interface for accessing menu options and form filling.
- Experimenter interface for automating the process of comparing different learning algorithms on data sets with different parameter settings.
- Knowledge flow interface for configuring data sources and learning algorithms.

WEKA is a free DM program with many features (Markov & Russell, 2006)

- many different algorithms are provided (e.g. logistics, decision tree J48, Bayes Net, Random Forests);
- many different FS methods (e.g. best first, greedy, Forward sequential selection, backward sequential selection, wrappers, classifiers);
- free open source;
- platform independent;
- easy to use

The following section describes briefly classifiers from WEKA that are used in this study.

WEKA Classifiers

23 WEKA classifiers have been used in this research study. Each of these is described briefly in this section and papers detailing more information relating to each of them can be found in APPENDIX C. These classifiers are categorised into 6 groups as shown in Table 2-4.

Table 2-4: 23 WEKA classifiers

Type of classifiers	Classifiers
Function	SMO
	Simple Logistic
	Logistic
	Multilayer Perceptron
Bayes	Bayes Net
	Naïve Bayes
	Naïve Bayes Simple
	Naïve Bayes Updatable
Lazy (Instance based learning)	IB1
	IBk
	KStar
	LWL
Meta	AdaBoost
	Classification Via Regression
	Decorate
	Multiclass classifiers
	Random Committee
	Ordinal classifier
Rules	PART
Tree	J48
	NBTree
	LMT
	Random Forest

2.2.3.1. Function Classifiers

SMO

Sequential Minimal Optimization (SMO) was created by John Platt, 1999. SMO algorithm is used to solve large mathematical programming optimization problems which are broken down into a number of smaller problems and subsequently are solved systematically (An & Slezak, 2008). SMO replaces all missing values, converts nominal attributes to binary attributes and normalizes all attributes automatically as a default option (Witten & Frank, 2005; WEKA 3.6.1, 2009). Advantages of SMO are: less computational time used to solve problems, only standard memory size is required to handle very large Support Vector Machine (SVM) training problems and are more tolerant to noise (Microsoft.com, 2009).

Logistic

This algorithm uses a two-class logistic regression model. As defined by An and Slezak (2008, p. 68): "*Logistic regression is a regression model for predicting the value of binomially distributed response variable Y = {C1, C2}*". The logistic with pairwise classification is used for predicting the estimates of probabilities for multi-class problems (Witten & Frank, 2005). The pairwise classification is "*a class binarization procedure that converts a multi-class problem into a series of two-class problems, one problem for each pair of classes*" (Park & Furnkranz, n.d. p. 1).

It is used to build logistic regression models, using LogitBoost. LogitBoost uses a simple regression procedure as a base learner for performing classification for multi-class problems, and selects attributes automatically based on the number of iteration to run

using cross validation. This algorithm can handle noise very well. (Witten & Frank, 2005; Cai, Feng, Lu & Chou, 2005).

Multilayer Perceptron

> *"An MLP is a network of simple neurons called perceptrons. The basic concept of a single perceptron was introduced by Rosenblatt in 1958. The perceptron computes a single output from multiple real-valued inputs by forming a linear combination according to its input weights and then possibly putting the output through some nonlinear activation function."* (Honkela, 2001, p. 1)

According to Seung (2002) multilayer perceptron uses backpropagation as an algorithm to train and classify the neural network. The backpropagation algorithm consists 2 phases:

- Forward phase for transforming input vectors into output vectors and computing errors between the expected and actual outputs.
- Backward phase for propagating errors and updating the results from both forward and backward phases.

2.2.3.2. Bayes Classifiers

Bayes Network

According to Tan *et al* (2006, p. 176), the network provides "*a graphical representation of the probabilistic relationship among a set of random variables*". It consists of a directed acyclic graph and a probability table for each node, where each node represents a random variable. The network makes classification by creating the

structure of the graph and calculating the probability values in the table for each node.

According to Groth (2000), some advantages and disadvantages of using Bayesian networks as follows:

Advantages:

- Easy to understand the way of the network works.
- The network is capable of classifying the data very well.

Disadvantages:

The prediction is bias toward outcomes that have high probabilities.

Naïve Bayes

According to Witten and Frank (2005) and Mitchell (2005), Naïve Bayes is based on Bayes's rule for probability which is, if there is a hypothesis H and observed data E based on H, then the probability of H holds over E will be Pr[H|E] = (Pr[E|H] Pr[H]) / Pr[E] with the assumption that the attributes are independent. Naïve Bayes uses density estimators to learn the mapping of attributes to the probability.

Naïve Bayes handles missing values very well because if there is a missing value, the algorithm does not count it and the probability will be calculated by using actual values which have been counted rather than on the total values.

Naïve Bayes Simple

This classifier uses a simple Naïve Bayes classifier to model numeric attributes with the implementation of a normal distribution (Witten & Frank, 2005; WEKA 3.6.1, 2009)

Naïve Bayes Updateable

This is an updateable version of Naïve Bayes but it does not handle the discretization process in the same way as Naïve Bayes; it only uses a density estimator for the mapping of attributes to the probability (Witten & Frank, 2005; WEKA 3.6.1, 2009).

2.2.3.3. Lazy Classifiers

IB1

IB1 is a simple instance based learner that stores samples of the training set and uses Euclidean distance measures to work out which sample is the closest to the unknown sample of the test set. As a result, the first sample (if two or more samples have the same minimal value) of training set with a smallest distance is selected and its class is then used to predict the class sample of the test set. IB1 is simple, effective but slow (Witten & Frank, 2005).

IBk

IBk is similar to IB1, using the same Euclidean distance measures but the number of nearest neighbors (k) can be specified at the time of running the classifier. If two or more neighbors are associated with a prediction, IBk converts the distance between

neighbors into weightings and then makes the decision (Witten & Frank, 2005).

KStar

KStar also uses the nearest neighbor method to predict the class of an unknown sample based on the samples learned in the training set, but it uses entropy based distance measures (WEKA 3.6.1, 2009) instead of Euclidean.

According to Cleary and Trigg (1995, p. 4), the entropy approach is based on information theory and "*the intuition is that the distance between instances be defined as the complexity of transforming one instance into another*". The use of an entropy based distance measure has some advantages such as handling nominal and numeric attributes, and missing values consistently.

LWL

LWL is locally weighted learning classifier. It is also a type of instance based learner algorithm, thus the classifier is constructed based on the weighting of samples. A classifier (e.g. J48 classifier) needs to be specified at the time of running the LWL (Witten & Frank, 2005).

2.2.3.4. Meta classifiers

AdaBoostM1

AdaBoostM1 is a special classifier designed only for nominal class prediction. The way it works is that all the samples in the training data are given the same weighting at start. They are then re-

assigned weights based on the results of the classification after each iteration in the training. As a result, some samples might have more weighting than others and some samples might have less. The process continues and at the end of each iteration, it reveals how often samples have been classified incorrectly. Only samples with correct classification will have their weights updated. All samples are nomalised and the weight is re-calculated for each sample. The weights of all the classes are summed up and the class prediction is based on the class with a highest total weight (Witten & Frank, 2005)

Classification Via Regression

As indicated by its name, this technique uses regression methods for classification.

Decorate

According to Melville (2004), the decorate algorithm is designed for generating diverse ensembles by using artificial training data. At the end of each iteration, an ensemble is created, artificial training data are generated, and the algorithm learns a new classifier and adds it to its current ensemble. Both original and artificial data are used to train the classifiers. The class labels for the artificial data are selected so they are totally different from the current ensemble's prediction. The process continues until the size of the target committee is obtained or the number of iterations is completed. The aim here is to produce a larger ensemble as it produces a more accurate model. However, the disadvantages are higher model complexity and training time (Witten & Frank, 2005).

Multiclass Classifier

This algorithm uses a two class prediction to classify the datasets with more than two classes. In order to do that the algorithm uses two strategies for classification:

- Voting is done by using a classifier to vote for a class and label an object to the class with the highest number of votes.
- Combination of probability of classifier estimates, "*the classifiers output an estimated class probability and assign a test object to the classifiers with maximal classification output*" (Tax & Duin, 2002, p. 1).

Random Committee

The random committee algorithm constructs an ensemble of base classifiers in a random fashion by using a different seed number every time a base classifier is created. It then averages the predictions from all these classifiers to determine a final prediction (WEKA 3.6.1, 2009; Witten & Frank, 2005).

Ordinal Class Classifier

This is a Meta classifier that supports the application of standard classification algorithms for ordinal class problems (WEKA 3.6.1, 2009).

2.2.3.5. Rules classifiers

PART

According to Witten and Frank (2005), the PART algorithm builds a partial decision tree to obtain a rule instead of building a complete tree. A split and conquer approach is used to build a rule and to eliminate the sample which has been covered, the cycle continues for the rest of the samples until all samples are removed. In order to make a rule, a temporary tree is created based on the current set of samples, the leaf node with the most number of samples is considered as a rule. The temporary tree is then deleted.

2.2.3.6. Tree classifiers

J48

J.48 is an updated version of C4.5. C4.5 is an extended version of the Iterative Dichotomiser (ID3) and is an algorithm for decision tree classifications (Han *et al*, 2006; Pujari, 2001). "*C4.5 is the most popular in the machine learning community*" (Salzberg, 1994, p. 1). According to Kohavi and Quinlan (1999), C4.5 constructs decision trees based on training data sets. The way that C4.5 works is to look at all attributes and applies normalized information gain measures to each attribute to work out which one has the highest normalized information gain. An attribute with the highest normalized information gain is then selected. The algorithm uses the selected attribute to make a decision to split a node and continues to work in that way on the sublist of the node to create child nodes. A leaf node is a class and a test node is a node that consists of 2 or more child nodes, each with a subtree (Kohavi &

Quinlan, 1999). When comparing C4.5 with its earlier version ID3, C4.5 has a number of improvements such as

- Working with data that consist of both continuous and discrete attributes.
- Handling data with missing attribute values.
- Pruning trees after they have been created.

LMT

LMT is an algorithm that constructs logistic model trees; a type of "*classification trees with logistic regression functions at its leaves*". (WEKA 3.6.1, 2009). The logistic model tree to be built is based on a normal decision tree. The logistic regression model is built for each node, the trees are pruned based on certain conditions and all the logistic regression models are combined to make a single model (Landwehr, Hall & Frank, 2003). LMT handles binary and multiclass target variables. In terms of attributes it is able to deal with missing values as well as numeric and nominal data types.

NBTree

NBTree is a hybrid of a decision tree and Naïve Bayes classifiers. The Naïve Bayes classifiers are created at the leaves of the decision tree. When a decision tree is created, the algorithm performs a cross validation to determine whether a node should be split further or it should be terminated as a leaf node comprising of a Naïve Bayes classifier (Witten & Frank , 2005).

Random Forest

Random Forest is an algorithm for decision tree classification techniques. The algorithm constructs a forest of "random trees", hence the name. According to Tan *et al* (2006), this algorithm consists of three steps:

- First step is to create random vectors (data points) by randomizing the training data (Figure 2-5, step 1).
- Second step is to use these random vectors to train multiple decision trees (Figure 2-5, step 2).
- Third step is to combine the predictions of all trained decision trees to form a predictive model tree (Figure 2.5, step 3).

The steps of the Random forest algorithm are illustrated in the following diagram:

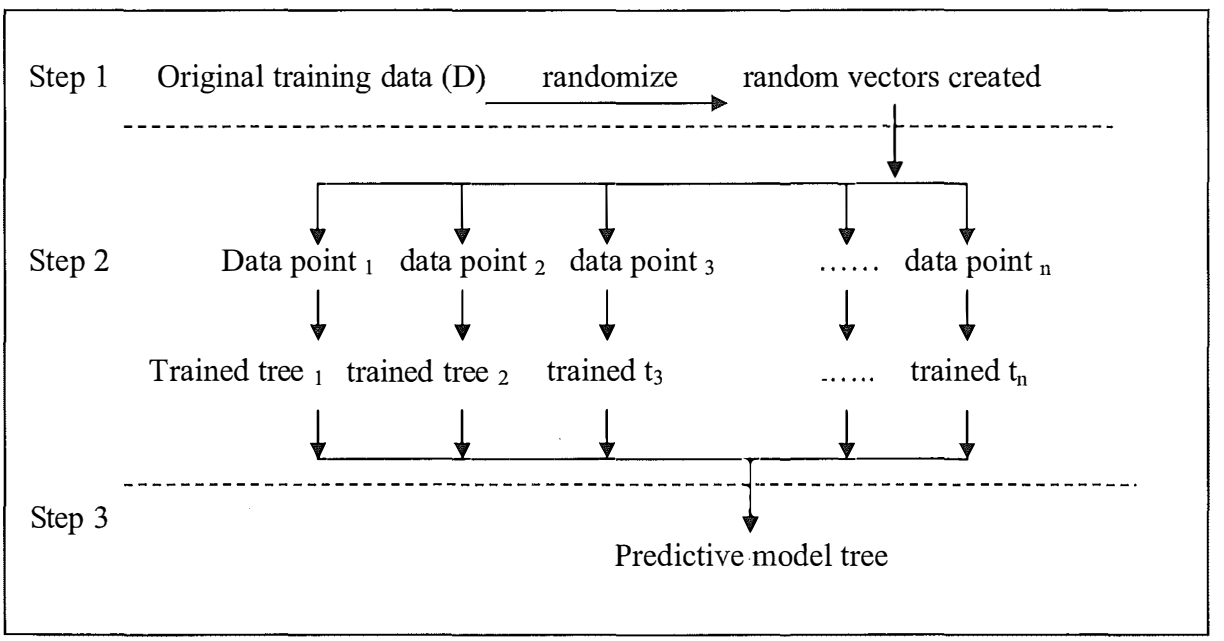


Figure 2-5: Random Forests, adapted from Tan *et al*, 2006, p. 215)

According to Breiman and Cutler (n.d.), Random forests have the following features:

- *"It does not overfit*.
- *It is unexcelled in accuracy among current algorithms*.
- *It runs efficiently on large data bases*.
- *It can handle thousands of input variables without variable deletion*.
- *It gives estimates of what variables are important in the classification." (*Breiman & Cutler (n.d., p. 3)),

2.3 Data mining approaches used to analyse data in health and AD area.

DM techniques have been used to analyse data in health areas such as bioinformatics and AD.

2.3.1. General health areas

- DM has been used in bioinformatics and pharmaceutical industries to identify normal and abnormal structural patterns of genes. The gene structures are analysed to discover any related diseases and then the findings can be used for developing medicines for treatments (Rudjer Boskovic Institute, 2001; Christen, 2005)

- DM techniques have also been used to analyse data in the metabolomic field. Classification techniques are used to generate models from metabolic data. The models are then used to diagnose patients with breast cancer (Kim, Park & Lee, 2007).

- DM techniques using ID3 and Naïve Bayesian classifiers were implemented in approaches for diagnosing the localization of primary tumors, prognostics of breast cancer recurrence, Thyroid diseases and Rheumatology (Kononenko, 1993)

- According to Ng and Pei (2007), DM has been used to analyse a large amount of data generated through high-throughput biotechnologies e.g. DNA microarrays. DM techniques have been used to:

  - analyse gene expressions for determining a certain disease condition, for example, lung cancer,
  - predict the response of the patient to a certain treatment, for example, cancer chemotherapy treatment,
  - predict how well patients recover from a certain medical intervention, for example, transplant operation,
  - understand a disease mechanism, for example, how a gene is expressed in a certain disease.

2.3.2. Alzheimer's disease areas

At present, one of the existing biomarkers for diagnosing AD is the use of tau protein and β-amyloid peptide concentration in cerebrospinal fluid (CSF) (Park, Li & Kricka, 2006; Hoffman & Froemke, 2009; McCorquodale & Myers, 2008). By identifying the concentration level of these peptides in CSF, people can be diagnosed for AD with the accuracy of 83% (ScienceDaily, 2009). Magnetic Resonance Imaging (MRI) and Positron Emission Tomography (PET) scanning are also used to scan the brain of potential patients with AD to identify the presence of β-amyloid

deposits in the brain (Hoffman & Froemke, 2009; McCorquodale & Myers, 2008). People with AD can be diagnosed with 100% accuracy with MRI (Lee-Frye, 2009), *"MRI reflects clinically defined disease stage better than the CSF biomarkers tested"* (Vemuri *et al*, 2009, p. 4). However these methods will only show that people have AD already and that neuro-degeneration has already occurred. In addition, these methods also have other limitations; collecting CSF is highly invasive and potentially dangerous, MRI and PET scans are very expensive and time consuming and may be distressful to people with dementia. Equipment associated with MRI/PET is very expensive and require highly skilled personnel to operate them – thus limiting access to areas that have the means to support them. Thus the search is on for a diagnostic tool that's inexpensive, has a high specificity and can be carried out as easily and accurately as a "*pregnancy test*", thus enabling diagnosis as early as possible.

A number of existing approaches have used DM techniques in terms of analysing AD data. The following sections will discuss three approaches in detail: analysing Magnetic Resonance Images (MRI) of Alzheimer's brain (Chen & Herskovits, 2005), diagnosing AD based on Plasma signalling protein (Ray *et al*, 2007) and identification of a 5 protein signature (Ravetti & Moscato, 2008).

2.3.2.1. Analysing Magnetic Resonance Imaging brain approach (Chen & Herskovits, 2005).

This approach involved using DM techniques in the process of detecting differences in the brain's structure and functions of AD people as compared with NAD people. This was done by analysing their MRI images from scans (Nuzzo, 2007).

Chen and Herskovits (2005) proposed a Bayesian Network Classifier with Inverse Tree structure (BNCIT) to work with MRI images. According to Chen and Herskovits, the BNCIT method was based on a Bayesian network (BN) but, having an inbuilt variable subset selection. With this feature, BNCIT overcomes problems with BN for handling high dimensional data in MRI images with millions of variables.

The procedure involved with BNCIT is as follows:

- obtain a training data set for a subject with MRI images;
- apply BNCIT to the training data set to create the classifier;
- predict AD for a new subject using his/her MRI images and the classifier.

Results achieved by BNCIT were more accurate than other methods such as decision tree and Naïve Bayes.

2.3.2.2. Classification and prediction of clinical Alzheimer's diagnosis based on plasma signalling proteins approach (Ray e*t al*, 2007)

Ray *et al (2007)* hypothesised that there would be characteristic changes in the signalling proteins during the process of developing AD. Ray *et al* proposed that if these changes can be detected then the disease can be detected based on the recognition of these changes.

With the hypothesis above, Ray *et al* applied DM techniques via SAM and PAM on an AD data set consisting of 259 plasma samples. They first successfully identified 19 proteins with highly significant gene expressions by using SAM; and subsequently an 18 protein biomarker signature was discovered by using PAM. The 18 protein

signature was evaluated by using the AD test set with the classification accuracy of 90% for AD and 88% for non-demented control (NDC). The biomarker was then tested on the mild cognitive impairment (MCI) test set to see how well the classifier can predict the AD from MCI data. Based on the results, the approach predicted, with a classification accuracy of 91% for MCI, patients who subsequently developed AD. This is significant because if MCI patients that can potentially developed AD can be identified early in the process, then steps can be put in place for their treatment/management. The dilemma here is that there is currently no cure for AD.

The 18 protein signature is quite significant in terms of *"finding a superior molecular test for an earlier diagnosis of Alzheimer's disease (AD)"* (Ravetti & Moscato, 2008, p. 1). The findings from this paper contributed significantly to AD area in terms of advancements in the development of possible diagnostic tools for AD.

2.3.2.3. Identification of a 5-protein biomarker for predicting AD (Ravetti & Moscato, 2008)

Subsequent to the publication of Ray *et al* (2007)'s experiments, which discovered the 18 protein signature, Ravetti and Moscato (2008), in light of recognising the importance of molecular biomarker signatures, carried out experiments involving biomarker signatures of 10, 6 and 5 proteins. Utilising the Ray *et al's* data sets, Ravetti and Moscato used an integrative data analysis method as their methodology to carry out their experiments. Their method comprised of 4 steps: abundance quantization, FS, literature analysis and classification selection.

As a result of the first 2 steps, together with literature analysis, the number of proteins selected out of the original 120 was 10, 6 and 5. DM programs PAM and WEKA, with 23 classifiers, were selected to test the accuracy of the classifiers trained using these protein signatures.

According to Ravetti and Moscato, the experiments have achieved several important findings: A 5 protein signature (a subset of the Ray *et al's* 18 protein signature) has the same overall classification performance as the 18. The 5 protein signature gives a smaller average prediction error when testing on 23 WEKA classifiers with the accuracy of 96% on AD. With some classifiers such as Simple Logistic and Logistic the accuracy is 100% for AD and 92% for NDC. Their findings are very significant in terms of reducing the number of proteins in biomarker signature, while the overall classification performance is still very robust. The 5 protein biomarker would be a simpler diagnostic technique for diagnosing AD at the early stages whilst subsequently reducing time and cost.

2.4 High dimensional data reduction approaches

As mentioned previously, data sets in bioinformatics are characterised by small sample size and very high dimensionality, and thus approaches for systematic selection of relevant features are important. A large number of attributes in a data set can create problems for clustering techniques, especially with a gene data set, which consists of a large number of variables, compared with the number of records (Bontempi, 2007). In addressing this problem, Bontempi proposed a blocking strategy. The key point of the proposed strategy is to increase the number of conditions applied to FS, hence reducing the dimensions of data. The results of their experiments (Bontempi, 2007) show that, with the blocking

strategy algorithms, the average accuracy of classification is better than the algorithm without the blocking strategy.

Qi Tan *et al* (2008) asserted that Linear Discriminant Analysis (LDA) and Biased Discriminant Analysis (BDA) techniques have been used in microarray classification to address the problem of high dimensional data. Qi Tan *et al* (2008) proposed a new technique to reduce high dimensional data by combining LDA and BDA, known subsequently as Adaptive Discriminant Analysis (ADA). Results from this study shows that ADA is more efficient than LDA and BDA (Qi Tan *et al*, 2008).

Biclustering algorithms have also been employed to find more subsets in a data set. The algorithm performs selections on both dimensions of the data matrix table at the same time. It applies local conditions to the data in the same cluster, resulting in more subsets being found, thus leading also to more patterns being found in the subsets (Madeira & Oliveira, 2004). Subsequently, Aguilar-Ruiz and Divina (2005) reported that the Evolutionary Biclustering (EBI) approach is more efficient than the traditional biclustering algorithms. The key to this new approach was to find the biggest size clusters with lowest average residues. It gave better results in terms of finding more types of genes and overcame the overlapping problem in traditional clustering techniques. However, according to Christinat, Wachmann and Zhang (2008) there is a problem with biclustering algorithms when discrete data are involved. Information may be lost in the process of discretization. In order to address this problem, the combination of both algorithms on discrete and continuous data needs to be carried out one after another, using the result from discrete data as well as the input source for the algorithm on continuous data. The combination approach is a more effective way of clustering gene data (Christinat *et al*, 2008).

2.5 Feature selection techniques

Another approach to address the dimensionality problem is to apply FS techniques. This is a process of selecting an optimal (of minimal size) subset of features based on criteria defined by Dash and Liu (1997, p. 2): "*classification accuracy does not significantly decrease*"

Many FS techniques have been developed and applied in a variety of fields such as DM and bioinformatics (Saeys, Inza & Larranaga, 2007; Portinale & Saitta, 2007). In general these techniques fall into three categories: filter methods, wrapper methods and embedded methods.

Filter methods are performed prior to the use of a learning algorithm. They select relevant attributes by looking at the data and giving scores to attributes. Attributes with low scores are removed from the list of attribute selection, as a result, only a number of high scoring attributes are retained and considered as relevant attributes. The learning algorithm is then trained, based on the relevant attribute subset.

Wrapper methods are different from the filter methods described above. Instead of finding a relevant attribute subset by an independent process, the wrapper method has a search induction algorithm as part of the subset attribute selection algorithm.

The following diagram illustrates the wrapper method with "black box" in detail:

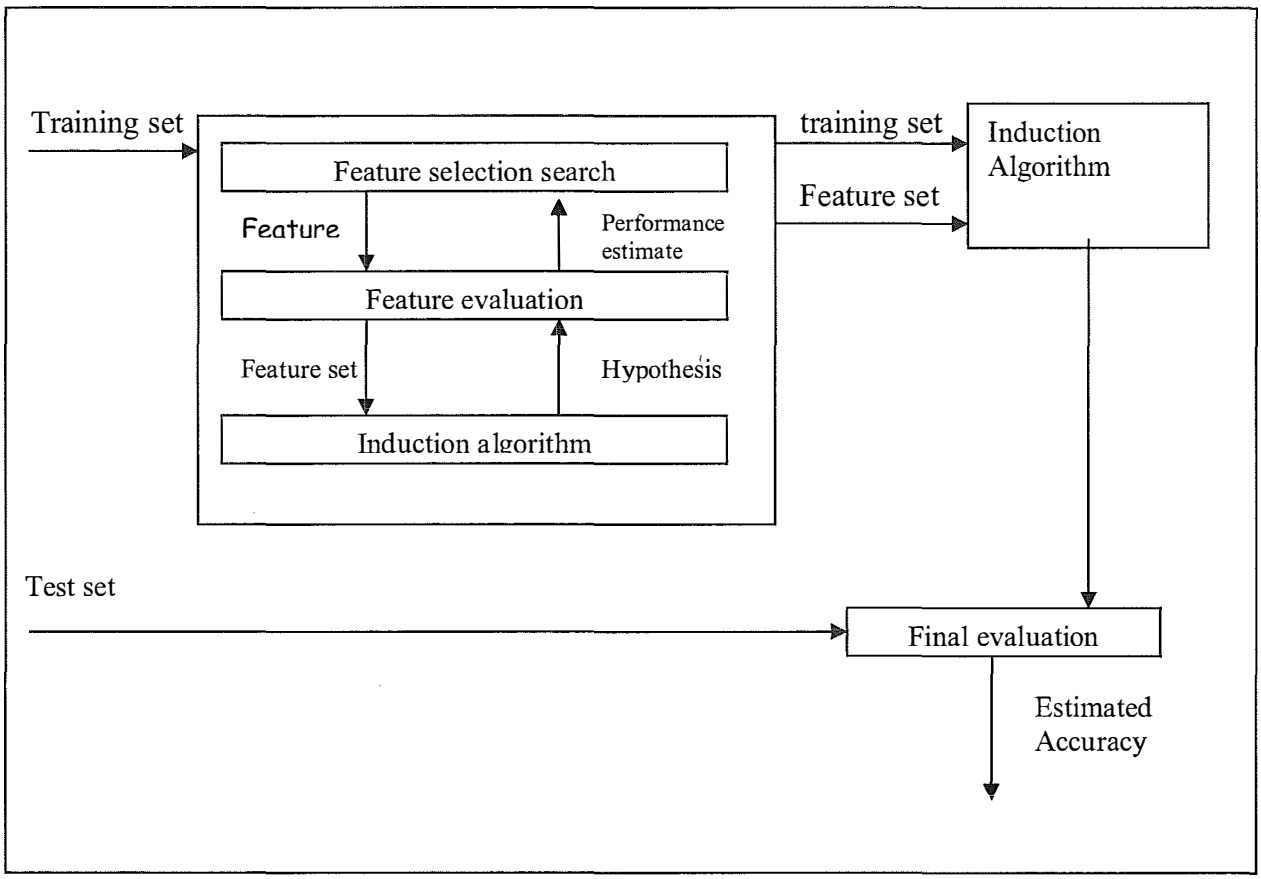


Figure 2-6: wrapper method algorithm to feature subset selection with black box, extracted from Kohavi & John (1996, p.27).

As in the diagram above, according to Kohavi and John, the induction algorithm is considered as a "black box" which consists of FS, feature evaluation and induction algorithm itself. The way the method works is that the "black box" runs on the training data set, as a result, a subset of highest ranking features is created, and is used as a final feature subset for the induction algorithm to perform on the training set. The final evaluation is carried out by using the test set to estimate the accuracy of the classifier.

Embedded methods are methods that have a FS algorithm built into classifiers, so that the search for relevant attributes can be done within the classifier itself on the data set.

According to Saeys *et al* (2007), advantages and disadvantages associated with each of these three methods are detailed in the following table:

Table 2-5: Advantages and disadvantages of filter, wrapper and embedded methods, adapted from Saeys *et al* (2007, p. 2508)

Method	Advantages	Disadvantages
Filter	Simple and fast computation, scalable, independent of classification algorithm	No interaction with classifiers, ignores feature dependencies
Wrapper	Interaction with the classifier, considering feature dependencies	Risk of overfitting, computational cost is higher than filter method
Embedded	Interaction with the classifier, computational cost is less than wrapper method	Classifier dependent selection

2.5.1 Overfitting, Accuracy, computational cost and time

High dimensional data are normally associated with a large number of attributes or features in bioinformatics (Yu *et al*, 2004). Irrelevant attributes can lead to problems in generating classifiers: overfitting, accuracy, time and computational cost (Deng & Moore, 1998; Yang & Honavar, 2007; Saeys *et al*, 2007).

In terms of overfitting, they can cause the classifiers to be confused in the classification process. Some irrelevant attributes might be taken into account for prediction and as a result, the classification may be determined by those irrelevant attributes, leading to poor classification results.

In terms of computational cost to perform classifications, irrelevant features cause the classifiers to spend more time to compute the prediction. Irrelevant attributes would increase the size of search space that cause the algorithm to take more time to explore all the possible combinations of all the features selected (Yang & Honavar, 1997). This can be illustrated by the formula to calculate a single prediction: $O(m^3 + m^2 \log N$ where N is the number of data points and m is the number of attributes used in the classification process (Deng & Moore, 1998).

Therefore, with relevant FS, the performance of learning algorithms is improved on a given data set (Portinale & Saitta, 2007).

2.5.2 Generic Steps in a Feature selection method

According to Dash and Liu (1997), FS techniques include the following steps:

- Find a feature generation scheme for obtaining candidate subsets. It involves a search strategy.
- Use an evaluation measure to measure the candidate subset.
- Pick a stopping criterion for determining the end point to stop the process of selecting the subset.

The following diagram illustrates the above steps:

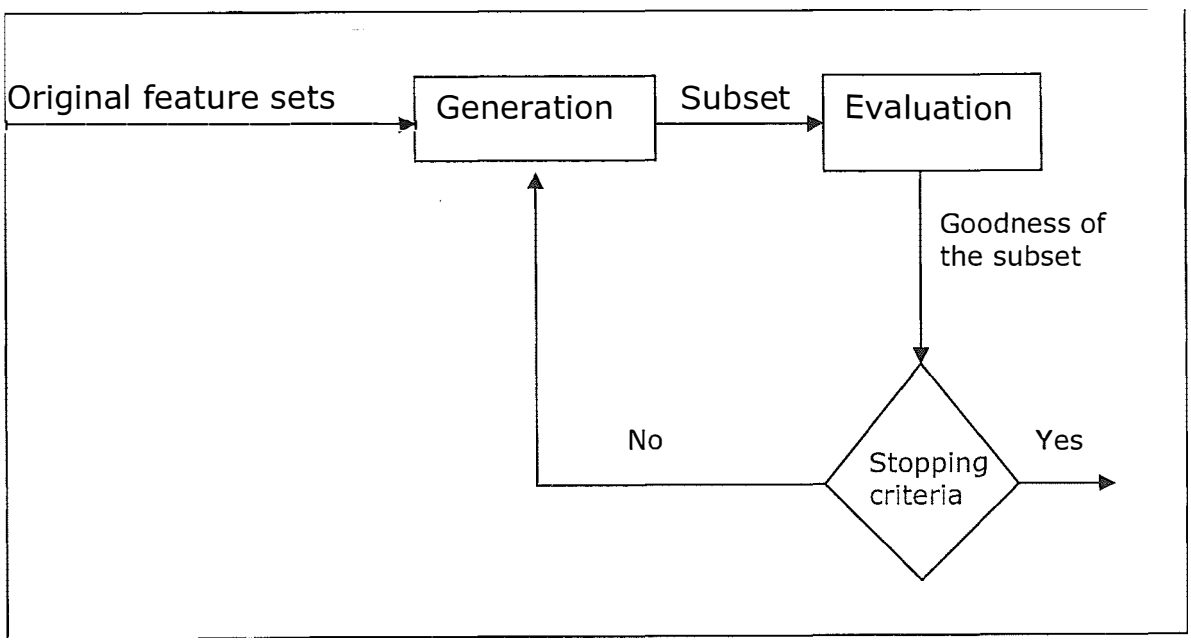


Figure 2-7: FS process, adapted from Dash and Liu (1997)

2.5.2.1 Generation procedure

This procedure uses a type of search strategies to generate candidate subsets of features. Attribute evaluation methods and search strategies available in WEKA 3.6.1 are listed in Table 2-6 and 2-7 respectively. Please refer to Witten and Frank (2005, p. 420-425) and WEKA 3.6.1 software package for more information. Some search strategies used in this study are briefly described as follows:

- **Forward sequential selection (FSS)**

  FSS starts with an empty subset, examines all the attributes provided in terms of their performance, one feature at the time and then selects the best one to add to the list of the subset. This procedure stops when there are no more "best attributes" to be added to the subset.

- **Backward sequential selection (BSS)**

  BSS is the opposite of FSS. BSS starts with the whole set of attributes and subsequently removes the worst one at a time and, eventually, a smaller subset of best attributes is created.

- **Greedy search**

  According to Kohavi and John (1996), greedy search is also called hill-climbing search or steepest ascent and it is the simplest search technique. It can search forward or backward through attribute subsets. The number of attributes to be obtained at the end of the search can be specified at the start of the search.

- **Ranker**

  Ranker is a ranking scheme . It sorts and ranks attributes. Ranker can "*perform attribute selection by removing the lower-ranking ones*" (Witten & Frank, 2005, p. 425). The number of attributes to be obtained at the end of the ranking can be specified at the start of the ranking scheme.

- **Race search**

  According to Witten and Frank (2005, 424), the search "*calculates the cross-validation error of completing attribute subsets using race search*." The race search can search forward, backward, schemata and rank racing. Attributes selected are ranked in a list. The search keeps doing that until all attributes are selected. The number of attributes to be obtained at the end of the search can be specified at the start of the search.

2.5.2.2 Evaluation measures

According to Dash and Liu (1997), evaluation measures are as follows:

- Distance measures: for 2 classes, select a feature X which has a greater different between the 2 class conditional probabilities than feature Y

- Information measures: a feature X is selected based on the information gain from feature X is greater that feature Y.

- Dependence measures: select feature X which has a correlation with class C higher than feature Y which has a correlation with class C.
- Consistency measures: *"find out the minimum sized subset that satisfies the acceptable inconsistency rate, that is usually set by the user."* Dash and Liu (1997, p. 6)

- Classifier error rate measures: use a classifier as a function evaluation to evaluate features selected. The classifier also is used to classify the class based on these features.

As shown in Table 2-6, there are a number of attribute evaluation methods in WEKA. They can be selected with the search methods (Table 2-7) in different combinations.

2.5.2.3 Stopping criteria

Stopping criteria are based on the generation procedure such as a number of features required, a number of iterations reached or based on an evaluation function such as when an optimal subset is found or no better subset are generated.

Table 2-6: Attribute evaluation methods, adapted from Witten and Frank, (2005, p. 421), and WEKA 3.6.1 (2009)

Name	*function*
CfsSubsetEval	*Consider the predictive value of each attribute individually, along with the degree of redundancy among them*
ChiSquaredAtributeEval	*Compute the chi –square statistic of each attribute with respect to the class*
ClassifierSubsetEval	*Use a classifier to evaluate attribute set*
ConsistencySubsetEval	*Project training set onto attribute set and measure consistency in class values*
FilteredSubsetEval	*Run an arbitrary subset evaluator on data that has been passed through an arbitrary filter.*
GainRatioAtributeEval	*Evaluate attribute based on gain ratio*
InfoGainAtributeEval	*Evaluate attribute based on information gain*
OneAtributeEval	*Use OneR's methodology to evaluate attributes*
PrincipalComponents	*Perform principal components analysis and transformation*
ReliefAtributeEval	*Instance-based attribute evaluator*
SVMAtributeEval	*Use a linear support vector machine to determine the value of attributes*
symmetricalUncer-AtributeEval	*Evaluate attribute based on symmetric uncertainty*
WrapperSubsetEval	*Use a classifier plus cross-validation*

Table 2-7: Search methods, adapted from Witten & Frank, (2005, p. 421) and WEKA 3.6.1 (2009)

Name	*Function*
BestFirst	*Greedy hill-climbing with backtracking*
ExhaustiveSearch	*Search exhaustively*
GeneticSearch	*Search use a simple genetic algorithm*
GreedyStepwise	*Greedy hill-climbing without backtracking; optionally generate ranked list of attributes*
LinearForwardSelection	*Extension of BestFirst*
RaceSearch	*Use race search methodology*
RandomSearch	*Search randomly*
Ranker	*is a ranking scheme, rank individual attributes (not subsets) according to their evaluation*
RankSearch	*Sort the attributes and rank promising subsets using an attribute subset evaluator*
ScatterSearch	*Performs an scatte search through the space of attribute subsets*

2.6 Summary

This chapter has described the DM techniques with different classification and clustering techniques, and DM software packages (SAM, PAM and WEKA) in bioinformatics. The chapter also described DM approaches in health area in general and in Alzheimer disease, specifically. Previous case studies of MRI approach (Chen & Herskovits, 2005), plasma signalling protein approach (Ray *et al*, 2007) and the identification of the 5 protein signature approach

(Ravetti & Moscato, 2008) have also been described. High dimensional data reduction approaches with FS techniques have been discussed in this chapter as well.

In the next chapter, research approach and data sets used in the study are described in details.

3. Research Approach.

This chapter describes the approach and the data sets used in the study. Section 3.1 detailed the research method, namely the engineering method; while section 3.2 and 3.3 describe the data sets used throughout the study. Lastly, limitations of the study are stated in section 3.4.

3.1 Research Method in this study

Figure 3-1 shows the steps in the Engineering method that was employed in this study.

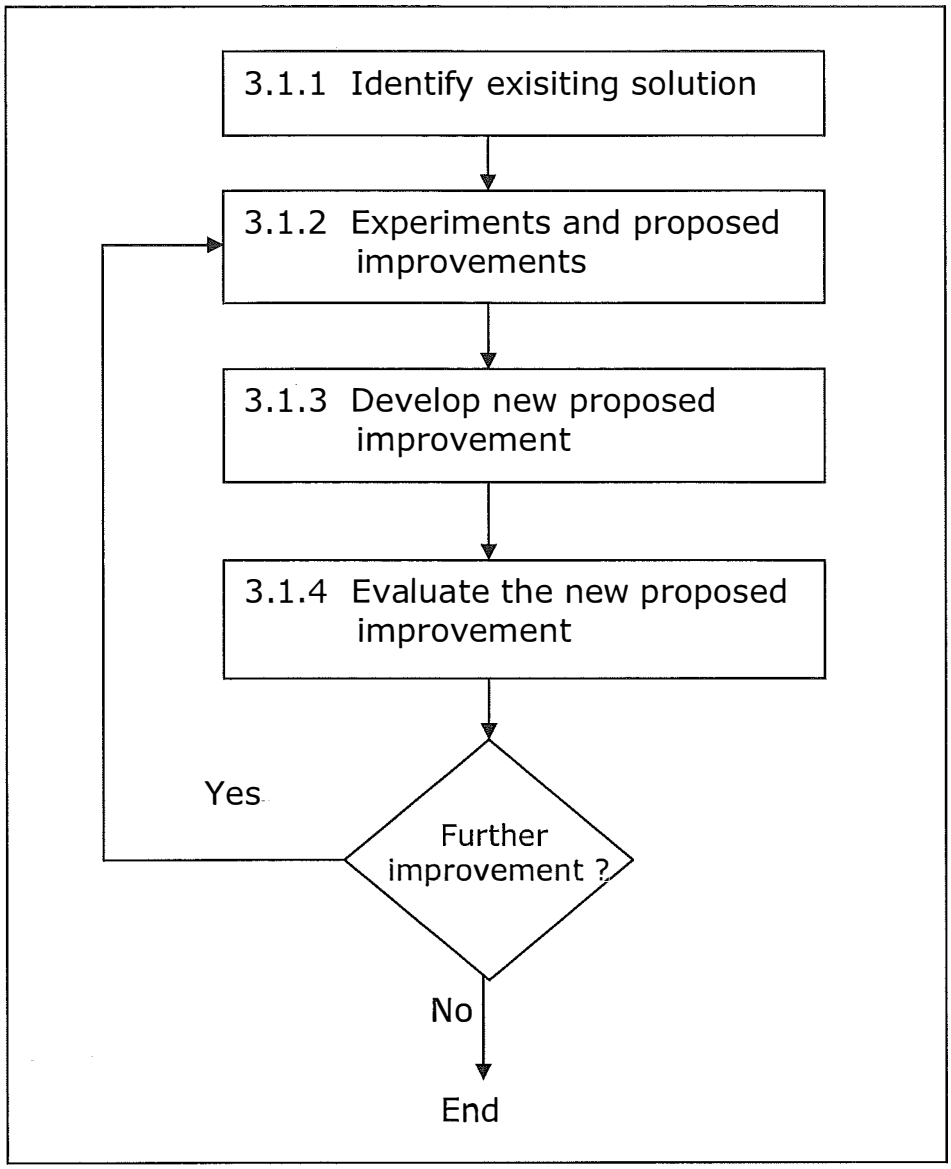


Figure 3-1: Engineering method used in this study adapted from Adrion (1993)

3.1.1. Identify existing solutions

DM tools in bioinformatics: SAM, PAM and WEKA programs are some of the many DM techniques that have been used previously by researchers. These algorithms and FS methods were investigated in terms of their algorithms, performances, application areas, strengths and weaknesses in the first step of this study. The two previous case studies (described in sections 2.3.2.2 and 2.3.2.3)

were then examined thoroughly for their methods, procedures, materials involved, results and limitations as they form the basis of existing solutions --- the starting point for this study.

3.1.2. Experiments and proposed improvements.

As a proof of concept, this study attempts to carry out the same steps using the information provided in these two previous case studies to re-produce their associated results. Based on analysis of these results, a systematic approach was used in further explorations of the data; with the aim of identifying other interesting protein signatures. Experiments with various FS strategies were also carried out to examine the impact of different FS methods returning different subsets of selected features and how that then affects the accuracy of classification models trained using these corresponding set of features.

3.1.3. Develop the new proposed improvements.

Once new protein signatures have been identified, PAM and WEKA with 23 classifiers (Section 2.2.3, Table 2-4) will be used to investigate the performances of classification models trained using these selected signatures. The 23 classifiers used are the same as Ravetti and Moscato's, because one of the aims here is to be able to carry out a comparison of the obtained results to that of the Ravetti Moscato study. As the protein signatures from this study are slightly different to those in Ravetti and Moscato, it is important to use the same 23 classifiers to ensure that any differences in the classification results are due to the protein signatures. In terms of specifically addressing the research question 1.4.1.2, WEKA FS methods were also used to select different feature subsets which were subsequently evaluated via a WEKA classifier.

3.1.4. Evaluate the new proposed improvements

Based on the analysis of the results from the previous step (3.1.3), a subset of tasks associated steps 3.1.2 and 3.1.3 in Figure 3-1, were repeated for refinements of the results of the DM until no further improvements were observed

3.2 Alzheimer's disease data sets used in the study

The AD data set used in this study was obtained from Ray *et al* (2007). The following section describes them in detail:

The AD data set has a total of 259 samples with 120 known signaling proteins in MS-Excel format. The plasma samples associated with these data points were collected from several academic centres specialising in neurological or neurodegenerative diseases (Ray *et al*, 2007). For full details of how the data were processed please refer to the information found on the Nature website listed below:

http://www.nature.com/nm/journal/v13/n11/extref/nm1653-S1.pdf

The data set is divided into a number of subsets as shown in Table 3-1.

Table 3-1: Description of subsets of the Ray *et al* data

Clinical diagnosis	Number
Alzheimer disease (AD)	85
Non-demented control (NDC)	79
	Training set
	AD: 43
	NDC : 40
	Test set
	AD: 42
	NDC: 39
Other dementia (OD)	11
Mild cognitive Impairment (MCI)	47
	MCI -> AD: 22
	MCI -> OD: 8
	MCI -> MCI: 17
Other neurological disease (OND)	21
Rheumatoid arthritis (RA)	16

Blood plasma from 259 individuals that were clinically diagnosed (via a Mini-mental state Examination) with pre-symptomatic to late stage AD were used in the Ray *et al* study. There were 120 signalling proteins associated with each sample. 85 of these samples are in the AD group and 79 into the NDC group. These two groups of samples were sub-divided equally into AD training set (43) and AD test set (40), NDC training set (40) and NDC test set (39). Thus the training set consists of 83 (43 AD and 40 NDC.) data points

In terms of separate test sets, there are two used in this study: one consisting of 42 AD, 39 NDC and 11 OD; making a total of 92 data points and the other consisting of 47 cases of MCI. An additional set, consisting of 21 OND and 16 RA were used by Ray *et al* to compare with AD data for distinguishing pattern of signaling protein expression in AD, via the use of the clustering package CLUTO 2.1.1. This subset was not used in this study.

The training set was used to generate the classification model. The AD and MCI test sets were used to evaluate the accuracy of classifiers for AD predictability. More specific details as follows:

- The AD test set was used with the aim of evaluating the prediction of AD against NDC.

- The MCI test set was used with the aim of evaluating the ability of the classifier to predict AD from MCI data.

These data sets described above were originally used in the experiments carried out by Ray *et al* (2007) and subsequently in their analysis by Ravetti and Moscato (2008)

For the full details of the 120 proteins, please refer to the information found on the Nature website listed below:

http://www.nature.com/nm/journal/v13/n11/extref/nm1653-S1.pdf

3.3 Data preparation

All data sets had the same format as samples were arranged in columns and proteins were arranged in rows. Although the data sets did not require any data pre-processing, the format of the data needs to be modified in order to work with the different programs SAM, PAM and WEKA as they use different input formats.

3.3.1. SAM data format

The following screen shots of the original training data sets show their format:

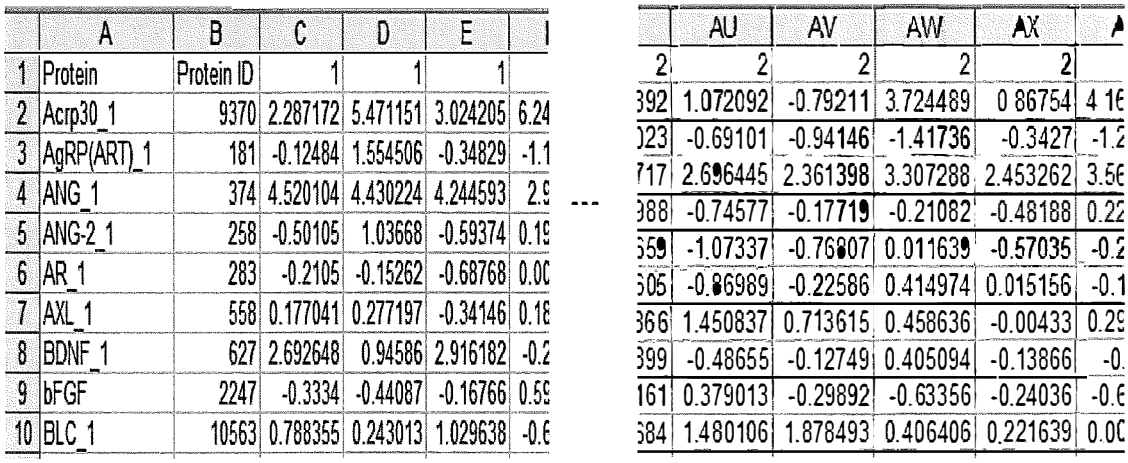
	A	J	K	L	
1	CLASS	AD	AD	AD	AD
2	ANG_1	4.404305	4.521486	2.911354	3.8
3	BDNF_1	0.397225	0.276861	-0.09201	1.6
4	BLC_1	-0.24781	-0.56939	-0.75756	0.
5	BMP-4_1	1.267004	0.14041	-0.04376	0.4
6	BMP-6_1	0.447816	-0.50919	0.068396	0.
7	CK b8-1_1	0.178844	-0.1398	-0.18074	-(
8	CNTF_1	0.151601	-0.35846	-0.22141	-0.
9	EGF_1	-0.3705	0.268294	-0.32451	-0.
10	Eotaxin_1	-1.36075	-0.61562	-0.89623	-0.

	AT	AU	AV	AW	
	NDC	NDC	NDC	NDC	ND
'17	2.696445	2.361398	3.307288	2.453262	3.5
166	1.450837	0.713615	0.458636	-0.00433	0.2
61	0.379013	-0.29892	-0.63356	-0.24036	-0
184	1.480106	1.878493	0.406406	0.221639	0.1
27	1.163023	1.092004	-0.32308	-0.26795	0.1
116	0.16363	0.486715	-0.6176	-0.37735	-0
03	0.632882	1.330187	-0.17693	-0.03449	-0
'45	1.520563	1.778444	1.163192	0.7447	1.2
161	1.207607	2.410608	2.104258	-0.50613	-0

Figure 3-2: Original training data format.

The class labels (row 1) AD and NDC were changed to 1 and 2 respectively because the SAM program requires the class labels to be in numeric format. Protein IDs were entered as an extra column immediately after the protein column so that the actual data started at column 3 as required by SAM (refer to 2.2.2.1 for input data format)

The whole training data set was sorted by proteins. The following screen shots show the new format of the training data set:



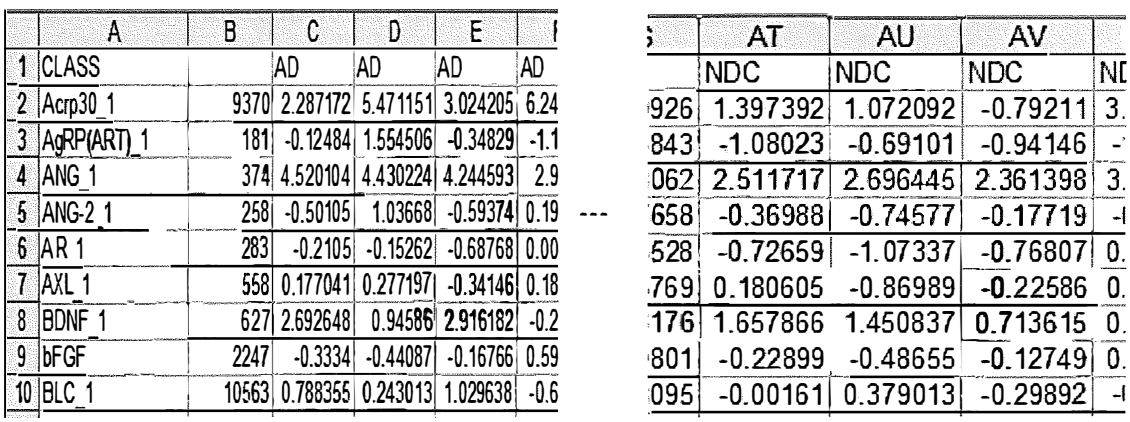
	A	B	C	D	E	
1	Protein	Protein ID	1	1	1	
2	Acrp30_1	9370	2.287172	5.471151	3.024205	6.24
3	AgRP(ART)_1	181	-0.12484	1.554506	-0.34829	-1.1
4	ANG_1	374	4.520104	4.430224	4.244593	2.9
5	ANG-2_1	258	-0.50105	1.03668	-0.59374	0.19
6	AR_1	283	-0.2105	-0.15262	-0.68768	0.00
7	AXL_1	558	0.177041	0.277197	-0.34146	0.18
8	BDNF_1	627	2.692648	0.94586	2.916182	-0.2
9	bFGF	2247	-0.3334	-0.44087	-0.16766	0.59
10	BLC_1	10563	0.788355	0.243013	1.029638	-0.6

	AU	AV	AW	AX	
	2	2	2	2	
392	1.072092	-0.79211	3.724489	0 86754	4 16
023	-0.69101	-0.94146	-1.41736	-0.3427	-1.2
717	2.696445	2.361398	3.307288	2.453262	3.56
988	-0.74577	-0.17719	-0.21082	-0.48188	0.22
659	-1.07337	-0.76807	0.011639	-0.57035	-0.2
605	-0.86989	-0.22586	0.414974	0.015156	-0.1
366	1.450837	0.713615	0.458636	-0.00433	0.29
399	-0.48655	-0.12749	0.405094	-0.13866	-0.
161	0.379013	-0.29892	-0.63356	-0.24036	-0.6
584	1.480106	1.878493	0.406406	0.221639	0.00

Figure 3-3: New training data format for SAM.

3.3.2. PAM data format

The data format used for PAM was the same format as SAM's format, but the PAM program allows the class labels to be in both numeric and alphabetic format, therefore there was no need to change the original class labels (AD and NDC) to numeric values. In fact, labelling the class with AD and NDC are much better as they are more meaningful and easier for identifying the classes. The following screen shots show the new format of the training data set and test data sets.



	A	B	C	D	E	
1	CLASS		AD	AD	AD	AD
2	Acrp30_1	9370	2.287172	5.471151	3.024205	6.24
3	AgRP(ART)_1	181	-0.12484	1.554506	-0.34829	-1.1
4	ANG_1	374	4.520104	4.430224	4.244593	2.9
5	ANG-2_1	258	-0.50105	1.03668	-0.59374	0.19
6	AR_1	283	-0.2105	-0.15262	-0.68768	0.00
7	AXL_1	558	0.177041	0.277197	-0.34146	0.18
8	BDNF_1	627	2.692648	0.94586	2.916182	-0.2
9	bFGF	2247	-0.3334	-0.44087	-0.16766	0.59
10	BLC_1	10563	0.788355	0.243013	1.029638	-0.6

	AT	AU	AV	
	NDC	NDC	NDC	NI
926	1.397392	1.072092	-0.79211	3.
843	-1.08023	-0.69101	-0.94146	-
062	2.511717	2.696445	2.361398	3.
658	-0.36988	-0.74577	-0.17719	-I
528	-0.72659	-1.07337	-0.76807	0.
769	0.180605	-0.86989	-0.22586	0.
176	1.657866	1.450837	0.713615	0.
801	-0.22899	-0.48655	-0.12749	0.
095	-0.00161	0.379013	-0.29892	-I

Figure 3-4: New training and test data format for PAM.

3.3.3. WEKA data format

The original data supplied with rows for proteins (features) and columns for samples. This format works fine with SAM and PAM, but it is not suitable for WEKA because WEKA requires the data in the format of rows for samples and columns for features, and the class labels must be in the last column. Therefore the data need to be converted to the WEKA required format. The following are screen shots of the new data format.

	A	B	C
1	ANG_1	BDNF_1	BLC_1
2	4.520104	2.692648	0.788355
3	4.430224	0.94586	0.243013
4	4.244593	2.916182	1.029638
5	2.90625	-0.26235	-0.65143
6	3.106781	0.084827	-0.53014
7	1.966611	-0.02477	-0.4665
8	3.670303	0.691213	-0.20747
9	3.365818	-0.30619	-0.59858
10	4.404305	0.397225	-0.24781

.....

DN	DO	DP	DQ
uPAR_1	VEGF-B_1	VEGF-D_1	CLASS
-0.25394	-0.05796	-0.3151	AD
-0.16785	-0.06098	-0.73985	AD
-0.78449	-0.73179	-0.43054	AD
0.485075	-0.09547	0.601015	AD
0.642871	0.389534	0.298311	AD
0.49224	0.441505	0.429353	AD
-0.33735	-0.48366	-1.05555	AD
0.035654	0.887269	0.779165	AD
-0.25675	-0.76956	-0.16107	AD

Figure 3-5: The new data format for WEKA

3.4 Limitation of using the data set

The limitation of this study is that only one data set from Ray *et al* (2007) was used but other studies such as Ravetti and Moscato (2008) have also just used this data set in their experiments. Owing to the fact that AD data collection is often a tedious (many diagnostic tests) and expensive process, there are not many existing data sets of this nature in the public domain. In addition given that the research is still in its infancy in terms of identifying blood plasma-based biomarkers, the Ray *et al* data set is the most easily available one at this point. Often, the data collection are

funded by commercial companies and the collected data are considered as having commercial potential, therefore they are not made available for public use.

3.5 Summary

Research method used in this study and the details of data sets were described. The investigations in chapter 4 and 5 employ the research method and use the data set in the experiments.

4. In-depth study of two existing approaches and mining for the interesting biomarkers

This chapter addresses the first aim of this study, which is to carry out an in-depth study of Ray *et al (2007)* and Ravetti and Moscato's (2008) experiments and to investigate if any further improvements can be made in terms of the results. A secondary aim here was to see whether the information provided in the Ray *et al* and Ravetti and Moscato's papers is sufficient in terms of being able to repeat their experimentations and results. The investigations carried out by both Ray *et al* and Ravetti and Moscato have focused on the classification and prediction of clinical diagnosis of AD based on plasma signaling proteins.

4.1 Classification and prediction of clinical Alzheimer's diagnosis based on plasma signaling proteins (Ray *et al*, 2007).

Ray *et al* used the DM programs Significance Analysis of Microarrays (SAM) and Prediction Analysis of Microarrays (PAM) on the data sets described in section, 3.2 and 3.3.

Ray *et al* first used an unsupervised learning analysis program, SAM to identify 19 proteins in the training set with significant differences in the expression of the signaling protein for AD and NDC samples. A supervised learning classification program, PAM was next applied to the training set to discover the 18 protein signature. The resulting classifier associated with this signature was used to classify the test data.

Their general approach is illustrated in Figure 4-1:

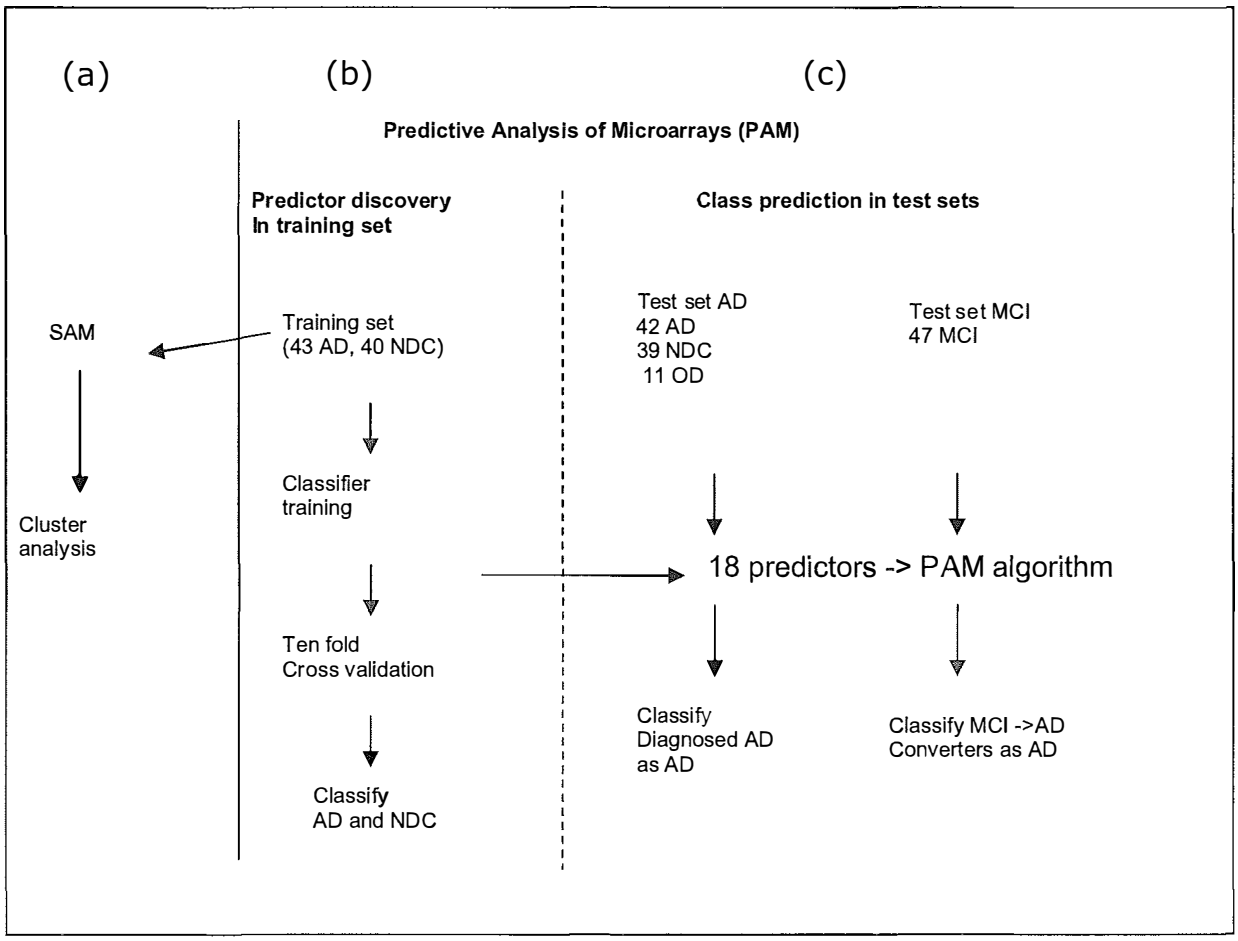


Figure 4-1: Prediction Analysis of Microarrays, extracted from Ray *et al* (2007, p.2)

As shown in Figure 4-1(a), SAM was applied to the training data set to find signalling protein with significant differences in concentrations followed by the clustering program CLUTO 2.1.1 which was used to cluster the training data in to AD and NDC groups. In Figure 4-1(b), PAM was employed to analyse the training data set and to obtain the classification model of 18 protein signature which was then used with 10 fold cross validation to classify AD and NDC. The classification model was evaluated by using the test AD set to classify AD from NDC and subsequently also on the MCI test set to see the classifier can predict AD from MCI (Figure 4-1(c)).

From these experiments, Ray *et al* reported that they achieved an overall 89 % accuracy with 90 % positive (sensitivity) for AD and 88% negative (specificity) for NDC samples. .

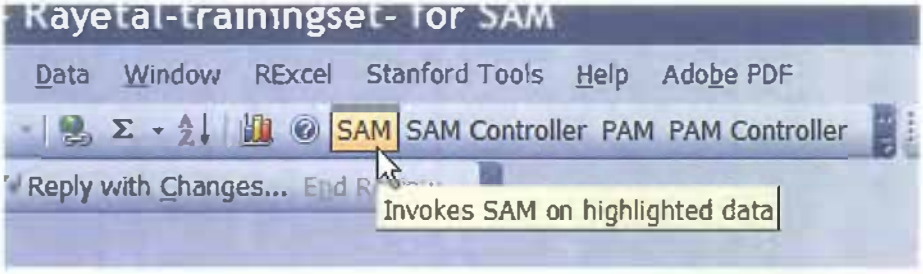
4.1.1. Implementation of the Ray *et al* study

4.1.1.1 Using SAM

Data set used in this study was described in section 3.2

Steps carried out in this study to recreate the original analysis using SAM:

- Run MS Excel program
- Load the training data set
- Select entire data including protein ID and protein name
- Activate the SAM program from MS Excel by clicking on the SAM add-in component as shown in Figure 4-2.



D	E	F	G	H	I	J
1	1	1	1	1	1	1
5.471151	3.024205	6.241606	5.965083	7.308389	0.398244	0.327759
1.554506	-0.34829	-1.15565	-1.08015	-1.14101	0.262278	-0.84829
4.430224	4.244593	2.90625	3.106781	1.966611	3.670303	3.365818
1.03668	-0.59374	0.192356	-0.06693	0.123216	0.02508	0.275702
-0.15262	-0.68768	0.007707	-0.18406	-0.42221	0.157029	-0.32835
0.277197	-0.34146	0.186427	0.152715	-0.20732	0.063176	0.289334
0.94586	2.916182	-0.26235	0.084827	-0.02477	0.691213	-0.30619
-0.44087	-0.16766	0.599456	0.921695	0.421405	0.24559	1.252732
0.243013	1.029638	-0.65143	-0.53014	-0.4665	-0.20747	-0.59858

Figure 4-2: Screen shot of activating SAM plugin component

- Select unlogged option and leave other options as default and click OK as shown in Figure 4-3.

- Experiment with different threshold values.

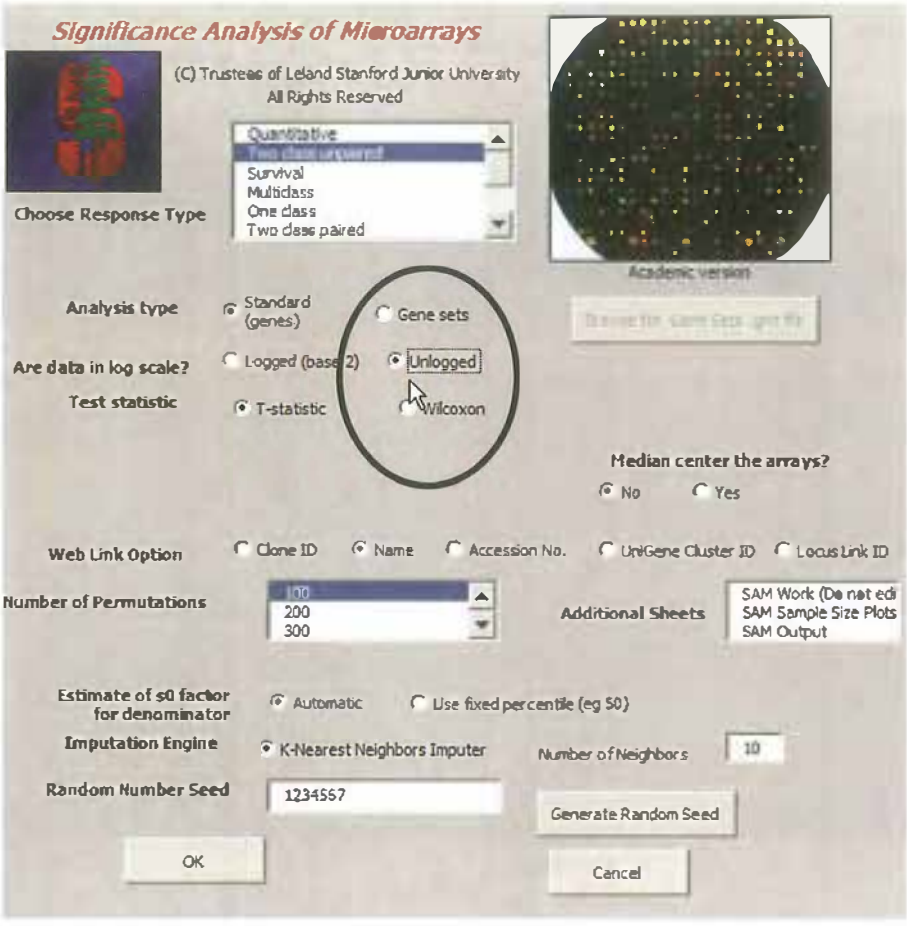


Figure 4-3: Screen shot of selecting unlogged option.

Results:

After some experimentation with the threshold values, a threshold value of 0.56 provided 19 proteins that has significant changes in their concentrations. The SAM plot of this experiment is shown as Figure 4-4.

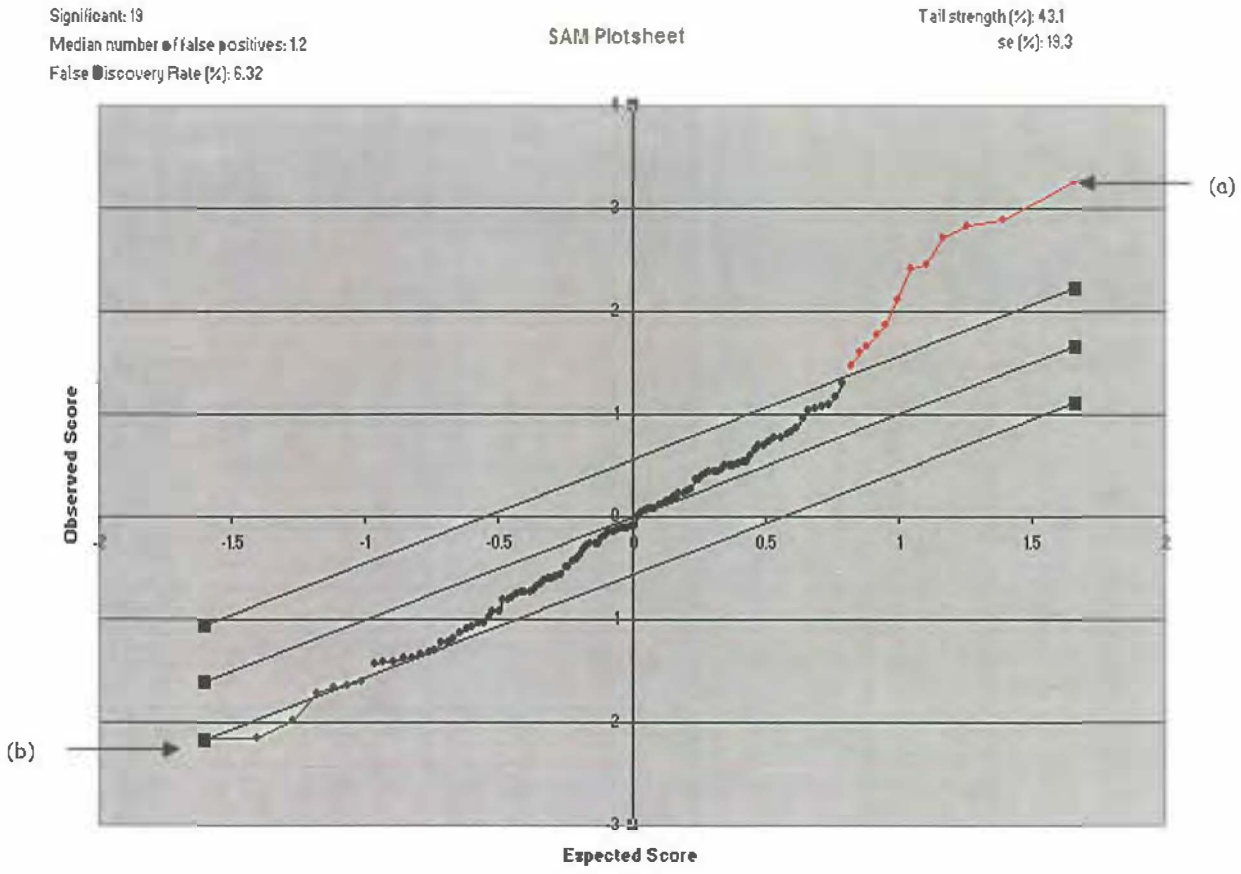


Figure 4-4: SAM plot sheet for 19 proteins significant differences in expression

As seen in the plot diagram above, all the dots in top right which fall outside the diagonal line (a) are positively expressed proteins (12) with significant differences in concentrations compared with others. All the dots in the bottom left, outside the diagonal line (b), indicate the negatively expressed proteins (7), with significant differences in concentrations.

The 19 proteins identified were the same as those identified by the Ray *et al* paper. These proteins are listed in Table 4-1.

Table 4-1: 19 proteins obtained via the analysis using SAM

12 positively expressed proteins
PDGF-BB_1
IL-1a_1
RANTES_1
TNF-a_1
EGF_1
M-CSF_1
IL-3_1
GCSF_1
GDNF_1
MIP-1d_1
MCP-3_1
MDC_1
7 negatively expressed proteins
IL-11_1
ICAM-1_1
ANG-2_1
TRAIL R4_1
IL-8_1
PARC_1
IGFBP-6_1

4.1.1.2 Using PAM

Classification on the training data set.

Steps carried out in this study:

- Run MS Excel program
- Select the entire data set.
- Activate PAM from MS Excel by clicking on the PAM plugin component, similar to how to activate SAM (Figure 4-2).
- Enter 1 for Class label in selection row to indicate that the class label starts at row 1. Enter 2 in the "Expression data starts in selection row" to indicate the data start at row 2, and accept other default options, as indicated in the screen shot below:

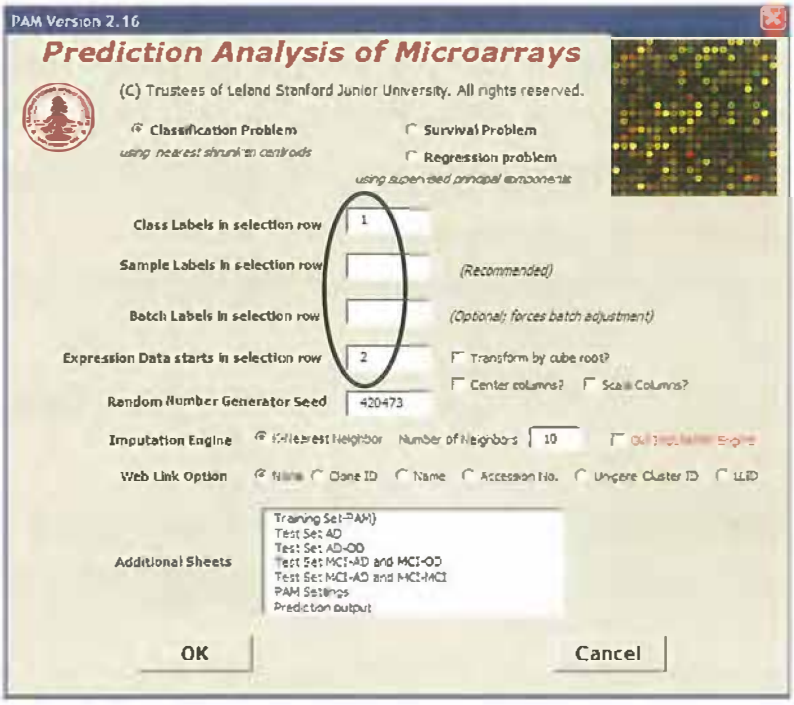


Figure 4-5: PAM classification entry screen dialog

- Press OK to proceed.

- Press Train button to proceed with the training as shown in Figure 4-6
- Experiments with different threshold values.

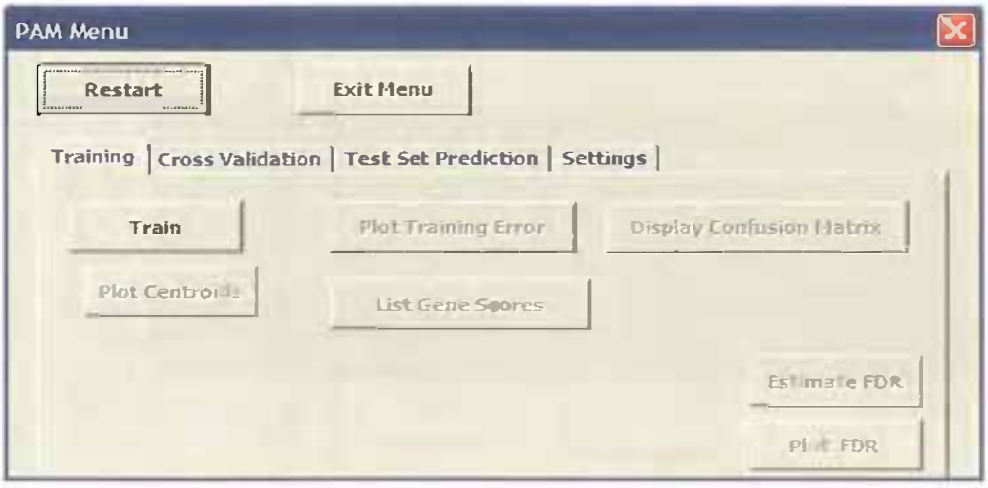


Figure 4-6: PAM training screen with menu options

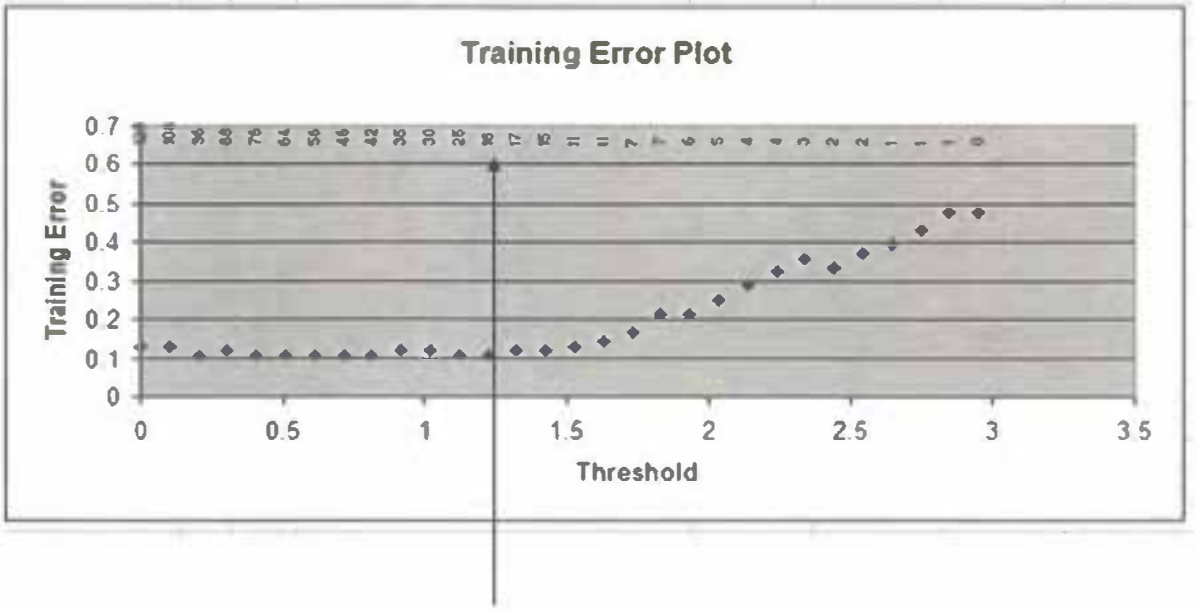
The number of specific proteins for a biomarker signature returned by PAM depends on the centroid threshold value. Decreasing the threshold resulted in increasing the number of proteins in a biomarker signature and vice versa. The ideal threshold value to be selected is to obtain a minimum number of proteins with the least number of classification errors generated from the confusion matrix.

Results:

With a threshold value of 1.22, the number of proteins obtained was 18. The PAM plot of this experiment is shown as Figure 4-7.

Training Confusion Matrix (Threshold=1.22)

True\Predicte	AD	NDC	Class Error rate
AD	41	2	0.046511628
NDC	7	33	0.175



Threshold = 1.22 with 18 proteins

Figure 4-7: Results of PAM classification on the training data set.

As shown in the results in Figure 4-7, the training error plot shows that with the threshold value of 1.22, 18 proteins were selected out of 120 proteins with a least number of training errors for AD and NAD.

The 18 proteins identified here were the same as those identified in the Ray *et al* paper. These 18 proteins is a subset of the 19 proteins obtained via SAM. These proteins are listed in Table 4-2.

Table 4-2: 18 protein signature from PAM with threshold of 1.22

Proteins
PDGF-BB_1
IL-1a_1
RANTES_1
TNF-a_1
EGF_1
M-CSF_1
IL-3_1
GCSF_1
GDNF_1
MIP-1d_1
MCP-3_1
IL-11_1
ICAM-1_1
ANG-2_1
TRAIL R4_1
IL-8_1
PARC_1
IGFBP-6_1

The training confusion matrix associated with the training is shown in Figure 4-7 as demonstrated the results obtained in this study is the same as those of Ray *et al*.

		AD	NAD
Clinical Diagnosis		43	40
Ray *et al*	AD	41	7
	NAD	2	33
This study	AD	41	7
	NAD	2	33

Figure 4-8: Classification results (on training data): from this study and Ray *et al* with 18 protein signature.

As shown in Figure 4-8, the classifier generated in this study can predict correctly 41 AD out of 43 and 33 NAD out of 40. These results are the same the Ray *et al*'s results.

Evaluating the classification model using the test data set.

Classification model obtained via PAM is evaluated using 2 test data sets (described in section 3.2). This model is based on a 18 protein signature.

- The aim of the first test is to see if the classifier can distinguish AD from NDC.

Data set used in this study:

AD test data set.

Steps carried out in this study:

- Activate the Test Set Prediction option as shown in Figure 4-9

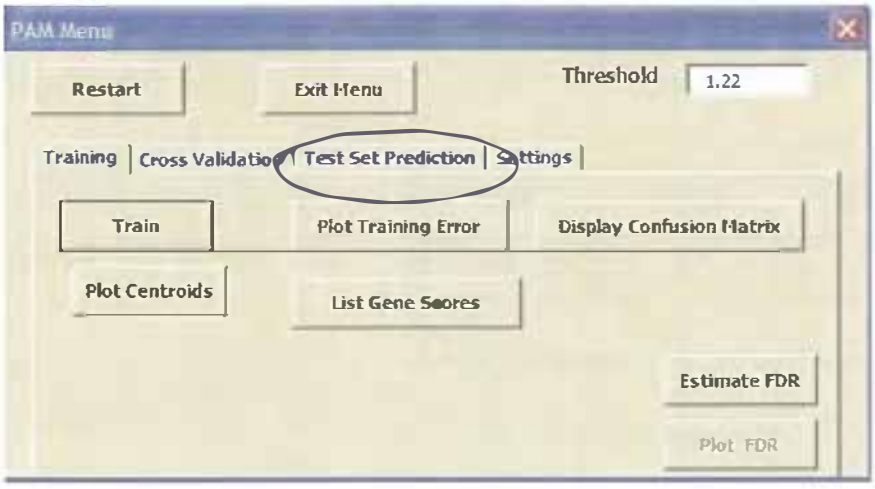


Figure 4-9: PAM dialog training menu screen with test set prediction option

- Specify the AD test data set and a range of data to test as shown in Figure 4-10.

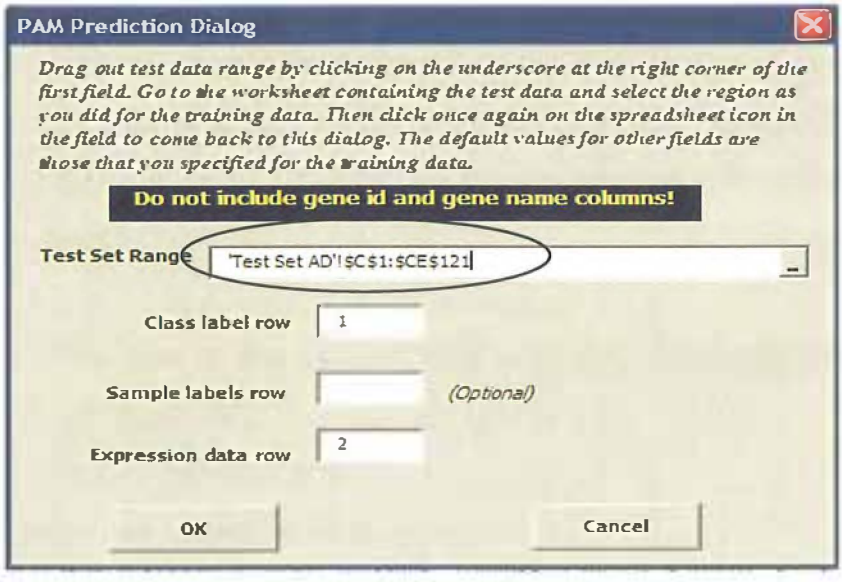


Figure 4-10: Input dialog screen to specify the test data

Results:

According to the test results from this study, the classifications for AD, NDC and OD with the threshold value of 1.22, the results are similar with the results stated in the paper for AD test data sets as Figure 4-11.

		AD	NAD	OD
Clinical Diagnosis		42	39	11
Ray *et al*	AD	38	5	1
	NAD	4	34	10
This study	AD	39	5	1
	NAD	3	34	10

Figure 4-11: AD test set classification results of study experiment For 18 protein signature compared with Ray *et al* and clinical diagnosis.

As results shown in Figure 4-11, the classifier generated in this study can predict correctly 39 AD out of 42, 34 NAD out of 39 and 10 OD out of 11. These results are almost the same the Ray *et al*'s results.

- The aim of the second test is to see if the classifier can predict AD from MCI.

Data set used in this study:

MCI test data set.

Steps carried out were similar to that of the first test.

Results:

According to the test results from this study, the classifications for AD, OD and MCI with the threshold value of 1.22, the results are similar with the results stated in the paper for MCI test data sets as Figure 4-12.

		AD	OD	MCI
Clinical Diagnosis		22	8	17
Ray et al	AD	20	0	7
	NAD	2	8	10
This study	AD	19	0	8
	NAD	3	8	9

Figure 4-12: MCI test set classification results for 18 protein signature compared with Ray *et al* and clinical diagnosis.

As shown in Figure 4-12, the classifier generated in this study can predict correctly 19 AD out of 22, 100% OD and 9 MCI out of 11. These results are similar to Ray *et al*'s results.

4.1.2. Further experimentation

Further experimentation with PAM was carried out in this study to see if we can find a biomarker signature with less than 18 proteins and still similar classification results.

Steps carried in this experiment are similar to those previously described for using PAM.

Results:

Experiments with different threshold values found that, with a threshold value of 1.29, the number of proteins obtained were 17 as shown in Table 4-3.

Table 4-3: 17 protein signature from study experiment

Ray *et al* (18 protein signature)	Study experiment (17 protein signature)
PDGF-BB_1	PDGF-BB_1
RANTES_1	RANTES_1
IL-1a_1	IL-1a_1
TNF-a_1	TNF-a_1
EGF_1	EGF_1
M-CSF_1	M-CSF_1
ICAM-1_1	ICAM-1_1
IL-11_1	IL-11_1
IL-3_1	IL-3_1
GCSF_1	GCSF_1
ANG-2_1	ANG-2_1
PARC_1	PARC_1
GDNF_1	GDNF_1
TRAIL R4_1	TRAIL R4_1
IL-8_1	IL-8_1
MIP-1d_1	MIP-1d_1
IGFBP-6_1	IGFBP-6_1
MCP-3_1	

As results shown in Table 4-3, the new 17 protein signature is a subset of the 18 proteins from Ray *et al*. The protein MCP-3_1 is not included in the 17 protein signature.
The results of training classification are listed in Figure 4-13.

		AD	NAD
Clinical Diagnosis		43	40
Ray et al	AD	41	7
	NAD	2	33
This study with 17 proteins	AD	41	8
	NAD	2	32

Figure 4-13: Learning classification results from study experiment with the 17 protein signature.

As results shown in Figure 4-13, the classifier generated in this study can predict correctly 41 AD out of 43 and 32 NAD out of 40. These results are similar the Ray *et al*'s results.

The classification model of the 17 protein signature was evaluated with the test AD data set and the classification results are the same results as shown in Figure 4-11.

The classification model of the 17 protein signature generated was tested with MCI test data set and the classification results are the same results as shown in Figure 4-12.

Following information provided in the Ray *et al* paper, the study is able to reproduce similar results stated in the paper. The finding of the 17 protein signature was a subset of 18 protein signature. It also gives classification results as good as the 18 protein signature on the test data.

The next experiment to be carried out is the Ravetti and Moscato (2008) experiment.

4.1.2 Identification of a 5-protein signature for predicting AD (Ravetti & Moscato, 2008)

This investigation used the Ray *et al* (2007) data sets. In terms of DM software, PAM and 23 WEKA classifiers were also used.

According to Ravetti and Moscato (2008), the identification of a protein molecular signature is significant and important in terms of providing the accurate form of diagnosis to identify the disease as early as possible. They identified a 5 protein signature, namely, IL-1$\propto$, IL-3, EGF, TNF-$\propto$ and G-CSF. This is a subset of the Ray *et al*'s 18 protein signature.

The method used in their experiment consists of 4 steps which were applied on the training data set:

1. Abundance quantization
2. Feature selection
3. Literature analysis
4. Classification analysis using 23 WEKA classifiers.

In the first 2 steps, Fayyad and Irani's algorithm (Fayyad & Irani, 1993) was used to quantize and select attributes. The Ravetti and Moscato study experimented with 18, 10, 6 and 5 protein signatures. The protein associated with these protein signatures are shown in the Table 4-4.

Table 4-4: 18, 10, 6 and 5 protein signatures

protein name	18 protein signature	10 protein signature	6 protein signature	5 protein signature
ANG 2	√			
EGF	√	√	√	√
G-CSF	√	√	√	√
GDNF	√			
ICAM 1	√			
IL-∝	√	√	√	√
IL-3	√	√	√	√
IL-6		√	√	
IL-8	√			
IL-11	√	√		
IGFBP-6	√			
MCP-3	√	√		
M-CSF	√			
MIP-1d	√	√		
PARC	√			
PDGF-BB	√	√		
RANTES	√			
TNF-∝	√	√	√	√
TRAIL R4				

As shown in the Table 4-4, the smaller signature is a subset of the bigger signature.

Classifiers were then obtained by training with each of these signatures. The test sets were used to evaluate the influence of the different biomarker signatures using 24 different classifiers (PAM

program and 23 WEKA classifiers). In terms of parameters associated with training involving WEKA classifiers, default settings were used. The results obtained by Ravetti and Moscato (2008) are shown in Table 4-5.

Table 4-5: Classification results average over 24 classifiers for each biomarker signature, adapted from Ravetti and Moscato (2008).

Protein signature	AD (64)	NAD (75)
18	86%	80%
10	88%	76%
6	87%	79%
5	88%	79%

The results in Table 4-5 were based on the AD test data set and the MCI test data set. The AD data set (64) consist of 42 AD from the AD test data set and 22 of those diagnosed with AD from MCI test set. In terms of the NAD test set (75), it consists of 50 NDC (39 NAD + 11 OD) from AD test data set and 25 NAD (8 OD + 17 MCI) from the MCI test data set. The percentages for each biomarker signature shown in this table were calculated over the 64 AD and 75 NAD.

4.1.3 Implementation of Ravetti and Moscato's experiment in this study.

According to M. Ravetti (personal communication, August 18, 2009), the first 2 steps used in their study were implemented by using a commercial software, Mixed Integer Problem (CPLEX), which implemented Fayyad and Irani's algorithm. This commercial software is not available for use in this study, thus the decision was made to use WEKA's implementation of Fayyad and Irani's algorithm instead.

Steps carried out in this study:

- In conjunction with Fayyad and Irani, various FS strategies from WEKA were then used to obtain biomarker signatures of size of 18, 10, 6 and 5. Table 4-6 shows the signatures obtained from this step.

Table 4-6: WEKA FS of 18, 10, 6 and 5 protein signatures.

18 protein signature	**10 protein signature**	**6 protein signature**	**5 protein signature**
IL-1a_1	IL-1a_1	IL-1a_1	BMP-6_1
PDGF-BB_1	IL-3_1	IL-3_1	IL-1a_1
Fractalkine_1	PDGF-BB_1	BMP-6_1	IL-3_1
IL-3_1	IGFBP-2_1	GCSF_1	TNF-a_1
ANG_1	ANG-2_1	TNF-a_1	GCSF_1
sTNF RI_1	GM-CSF_1	ANG_1	
BDNF_1	I-309_1		
BLC_1	IGF-1_1		

BMP-4_1	IL-15_1		
BMP-6_1	IL-2_1		
CNTF_1			
EGF_1			
Eotaxin_1			
Eotaxin-2_1			
Eotaxin-3_1			
FGF-7_1			
Fit-3 Ligand_1			
GCP-2_1			

- Each signature is used for training with the 23 WEKA classifiers and this is followed by the evaluation of the classifiers using the test sets.

Results:

The classification results from classifiers trained using the 18, 10, 6 and 5 protein signatures are summarised in Tables 4-7, 4-8, 4-9 and 4-10.

Table 4-7: Report of the results of the 23 classifiers when using 18 protein signature from WEKA FS.

		Overall (AD +MCI)		Test Set AD		Test Set MCI	
	Total	**AD errors**	**NAD errors**	**AD errors**	**NAD errors**	**AD errors**	**NAD errors**
Dataset size	**139**	**64**	**75**	**42**	**50**	22	25
Classifier							
SMO	29	9	20	4	8	5	12
Simple Logistic	23	6	17	2	5	4	12
Logistic	37	12	25	8	10	4	15
Multilayer Perceptron	33.6	12.7	20.9	7.6	8.3	5.1	12.6
Bayes Net	28	6	22	2	9	4	13
Naïve Bayes	36	7	29	5	15	2	14
Naïve Bayes Simple	36	7	29	5	15	2	14
Naïve Bayes Up	36	7	29	5	15	2	14
IB1	35	15	20	7	8	8	12
Ibk	35	15	20	7	8	8	12
Kstar	37	7	30	4	13	3	17
LWL	28	15	13	5	3	10	10
AdaBoost	35	12	23	7	10	5	13
ClassViaRegression	24	9	15	3	4	6	11
Decorate	32.8	12.6	20.2	5.9	9.3	6.7	10.9
Multiclass Classifier	37	12	25	8	10	4	15
Random Committee	29.6	8.4	21.2	3.1	8.7	5.3	12.5
j48	28	12	16	4	7	8	9
LMT	23	6	17	2	5	4	12
NBTree	28	10	18	3	8	7	10
Part	26	17	9	10	1	7	8
Random Forest	30.6	8.6	22	3.3	9.8	5.3	12.2
Ordinal Classifier	28	12	16	4	7	8	9
Average	31.11	10.36	20.75	5.00	8.57	5.37	12.18
Agreement(%)	78%	84%	72%	88%	83%	76%	51%

As shown in Table 4-7, firstly, the error results of each classifier were calculated separately for the "Test set AD" and the "test set MCI" (third and fourth column of table). Within each of these categories, there are two sub-groups: "AD errors" and "NAD errors". For example, in terms of SMO, from a total of 42, there are 4 classification errors in the subgroup of "AD errors" in the column for "Test set AD". Next, these individual errors were summed up to make up the total errors (i.e. the column with heading "Overall (AD + MCI)". Again, using the row associated with SMO as an example, the AD errors under the column "Overall (AD + MCI)" is 9 (comprising of 4 from "Test Set AD" and 5 from "Test Set MCI").

Finally the total error of AD and NAD were summed up together to make the grand total error for each classifier (e.g. for SMO – the value of 29 is made up of 20 and 9 in the "Overall (AD + MCI)" column.

The average error in the "Total" column was calculated by summing up all the total errors of 23 classifiers in the column "Total" (from SMO down to Ordinal Classifier) to get the grand total error, and then divide the grand total error over 23 (a total number of classifiers) to get the average error of 23 classifiers (e.g. for average "31.11" (Table 4-7) - sum up the total errors of 23 classifiers from SMO down to Ordinal classifier, which gives the grand total of 715.6 and then divide 715.6 over 23 to get 31.11)

The agreement % in the "Total" column was calculated by dividing the average error of 23 classifiers over the total samples (139) and then multiply 100 to obtain overall error percentage. Finally subtract the overall error percentage from 100 to get an agreement % (e.g. for agreement % "78%" (Table 4-7) - divide average error "31.11" over 139, which gives 0.22 and then multiply 0.22 to 100 to obtain the overall error percentage 22%. Finally subtract 22% from 100 to get the agreement % of 78%). These calculations were also similarly applied in Tables 4-8, 4-9, 4-10 and 4-14.

PAM classifier was not included in this study because PAM did not select the same biomarker signatures of 10, 6 and 5 as WEKA did. The idea of using the same biomarker signatures is to train and test all the 23 classifiers so the results obtained are more stable and less likely to be biased towards any specific classifiers.

Table 4-8: Report of the results of the 23 classifiers when using 10 protein signature from WEKA FS.

		Overall (AD +MCI)		Test Set AD		Test Set MCI	
	Total	**AD errors**	**NAD errors**	**AD errors**	**NAD errors**	**AD errors**	**NAD errors**
Dataset size	**139**	**64**	**75**	**42**	**50**	22	25
Classifier							
SMO	22	3	19	1	6	2	13
Simple Logistic	23	6	17	2	5	4	12
Logistic	20	5	15	1	4	4	11
Multilayer Perceptron	22	8.5	13.5	3.1	3.4	5.4	10.1
Bayes Net	29	16	13	6	3	10	10
Naïve Bayes	31	8	23	5	8	3	15
Naïve Bayes Simple	32	8	24	5	9	3	15
Naïve Bayes Up	31	8	23	5	8	3	15
IB1	32	9	23	5	9	4	14
Ibk	32	9	23	5	9	4	14
Kstar	44	9	35	6	21	3	14
LWL	29	16	13	6	3	10	10
AdaBoost	29	7	22	5	10	2	12
ClassViaRegression	24	11	13	4	3	7	10
Decorate	25.9	10.7	15.2	4	4.7	6.7	10.5
Multiclass Classifier	20	5	15	1	4	4	11
Random Committee	25.5	10.6	14.9	3.7	5.8	6.9	9.1
j48	23	12	11	4	2	8	9
LMT	23	6	17	2	5	4	12
NBTree	24	10	14	3	4	7	10
Part	25	14	11	7	2	7	9
Random Forest	27.1	10.8	16.3	4.6	6.3	6.2	10
Ordinal Classifier	23	12	11	4	2	8	9
Average	26.80	9.33	17.47	4.02	5.97	5.31	11.51
Agreement(%)	81%	85%	77%	90%	88%	76%	54%

Table 4-9: Report of the results of the 23 classifiers when using 6 protein signature from WEKA FS.

		Overall (AD +MCI)		Exp - Test Set AD		Exp - Test Set MCI	
	Total	AD errors	NAD errors	AD errors	NAD errors	AD errors	NAD errors
Dataset size	**139**	**64**	**75**	**42**	**50**	22	25
Classifier							
SMO	25	12	13	3	4	9	9
Simple Logistic	28	6	22	2	9	4	13
Logistic	29	6	23	2	9	4	14
Multilayer Perceptron	27.5	5.8	21.7	0.7	8.6	5.1	13.1
Bayes Net	26	8	18	1	5	7	13
Naïve Bayes	24	8	16	2	4	6	12
Naïve Bayes Simple	25	8	17	2	5	6	12
Naïve Bayes Up	24	8	16	2	4	6	12
IB1	33	15	18	10	6	5	12
Ibk	33	15	18	10	6	5	12
Kstar	28	7	21	4	8	3	13
LWL	29	16	13	6	3	10	10
AdaBoost	39	8	31	2	14	6	17
ClassViaRegression	22	10	12	2	2	8	10
Decorate	38.6	16.9	21.7	8.2	10.3	8.7	11.4
Multiclass Classifier	29	6	23	2	9	4	14
Random Committee	32.6	12.3	20.3	4.8	9.1	7.5	11.2
j48	42	17	25	8	14	9	11
LMT	28	6	22	2	9	4	13
NBTree	32	12	20	4	7	8	13
Part	40	19	21	10	10	9	11
Random Forest	30.5	11.4	19.1	4	8.1	7.4	11
Ordinal Classifier	42	17	25	8	14	9	11
Average	**30.75**	**10.89**	**19.86**	**4.33**	**7.74**	**6.55**	**12.12**
Agreement(%)	78%	83%	74%	90%	85%	70%	52%

Table 4-10: Report of the results of the 23 classifiers when using 5 protein signature from WEKA FS.

		Overall (AD +MCI)		Test Set AD		Test Set MCI	
		AD errors	NAD errors	AD errors	NAD errors	AD errors	NAD errors
Dataset size	**139**	**64**	**75**	**42**	**50**	**22**	**25**
Classifier							
SMO	25	12	13	2	2	10	11
Simple Logistic	21	4	17	0	5	4	12
Logistic	22	2	20	0	6	2	14
Multilayer Perceptron	26.7	4.5	22.2	0.5	9.8	4	12.4
Bayes Net	26	8	18	1	5	7	13
Naïve Bayes	25	8	17	1	5	7	12
Naïve Bayes Simple	24	7	17	1	5	6	12
Naïve Bayes Up	25	8	17	1	5	7	12
IB1	31	10	21	6	7	4	14
Ibk	29	10	19	6	7	4	12
Kstar	28	5	23	1	11	4	12
LWL	29	16	13	6	3	10	10
AdaBoost	38	6	32	3	15	3	17
ClassViaRegression	25	8	17	1	7	7	10
Decorate	34	12.3	21.7	3.5	10.9	8.8	10.8
Multiclass Classifier	22	2	20	0	6	2	14
Random Committee	34.4	12.3	22.1	4.3	11.3	8	10.8
j48	36	11	25	2	14	9	11
LMT	21	4	17	0	5	4	12
NBTree	32	12	20	4	7	8	13
Part	31	13	18	4	7	9	11
Random Forest	32.5	12	20.5	4.3	9.8	7.7	10.7
Ordinal Classifier	36	11	25	2	14	9	11
Average	**28.42**	**8.61**	**19.80**	**2.33**	**7.73**	**6.28**	**12.07**
Agreement(%)	80%	87%	74%	94%	85%	71%	52%

Table 4-7, 4-8, 4-9 and 4-10, show the breakdown in terms of the classification results of the test data sets which is associated with each signature and within that, each classifier. With the non-deterministic classifiers: Multilayer Perceptron, Decorate, Random Committee and Random Forest, the results are associated with 10 runs, each with a different seed and then obtained an average from the results of the 10 runs.

Table 4-11 shows the overall results of 23 WEKA classifiers for each biomarker signature. It also shows a comparison to the results the Ravetti and Moscato (i.e. values listed in the column heading "paper"). For example, in the case of SMO in the case of the 18 biomarker signature, the value obtained in the experiments carried out in this study is 29 while Ravetti and Moscato have a value of 20.

Table 4-11: Overall errors for each biomarker signature over the test sets (139 samples) compared with the results from the paper.

	18 proteins		10 proteins		6 proteins		5 proteins	
Classifers	**paper**	**Re-run**	**Paper**	**Re-run**	**Paper**	**Re-run**	**Paper**	**Re-run**
SMO	20	29	23	22	20	25	19	25
Simple Logistic	25	23	25	23	18	28	18	21
Logistic	27	37	24	20	21	29	20	22
Multilayer Perceptron	21.7	33.6	21.8	22	25.6	27.5	21.6	26.7
Bayes Net	27	28	28	29	22	26	21	26
Naïve Bayes	23	36	30	31	23	24	19	25
Naïve Bayes Simple	23	36	31	32	24	25	20	24
Naïve Bayes Up	23	36	30	31	23	24	19	25
IB1	21	35	28	32	33	33	30	31
Ibk	21	35	28	32	33	33	30	29
Kstar	28	37	41	44	33	28	26	28
LWL	28	28	28	29	29	29	29	29
AdaBoost	23	35	31	29	27	39	31	38
ClassViaRegression	28	24	23.4	24	23	22	24	25
Decorate	23.1	32.8	28.3	25.9	24.7	38.6	21.8	34
Multiclass Classifier	27	37	24	20	21	29	20	22
Random Committee	26.1	29.6	26.3	25.5	26.6	32.6	26.1	34.4
j48	24	28	24	23	24	42	24	36
LMT	25	23	25	23	18	28	18	21
NBTree	26	28	23	24	21	32	21	32
Part	25	26	30	25	27	40	27	31
Random Forest	24.3	30.6	24.3	27.1	25.6	30.5	26.2	32.5
Ordinal Classifier	24	28	24	23	24	42	24	36
Average	**24.487**	**31.113**	**27.004**	**26.804**	**24.630**	**30.748**	**23.291**	**28.417**
Agreement(%)	**82%**	**78%**	**81%**	**81%**	**82%**	**78%**	**83%**	**80%**

The overall "average errors" and agreement percentages of the 23 classifiers for each protein signature, which was selected from WEKA FS, are similar (but not exactly the same) to those in the original paper. These variations are due to the fact that this study is not using the same implementations of the FS technique (CPLEX vs.

implementation in WEKA) and thus there will be differences in the sets of selected features used for training the classifiers. This finding also re-enforces the belief that different FS techniques will likely impact the final classification results as this case only involved a different implementation of the same algorithm.

4.2 Exploration for obtaining biomarker signature with a size less than 5

Further experiment was carried out in this study to see if we can find a biomarker signature with less than 5 proteins and still can be used to train a classifier that produces similar classification results as the 5 biomarker signature. Using a systematic approach we analysed the 5 protein signature from this study and compared it to that obtained by Ravetti and Moscato. As shown in (Table 4-12), there are 4 common proteins between the two sets. Only one protein differs - BMP-6 instead of EGF.

Table 4-12: Five protein signature from paper and WEKA FS

Proteins (paper)	**Proteins (WEKA)**
EGF	BMP-6
G-CSF	G-CSF
IL-1a_1	IL-1a_1
IL-3	IL-3
TNF-a_1	TNF-a_1

The decision then was to use a 4 protein signature (shown in Table 4-13) for further investigations.

Table 4-13: Four protein signature.

Proteins (analysis)
G-CSF
IL-1a_1
IL-3
TNF-a_1

The new 4 protein signature was used to train 23 classifiers with the training data set. The classification model from each classifier was evaluated with the AD test set and then tested with the MCI test set, in a similar manner to the Ravetti and Moscato study. The results associated with these classifiers are shown in Tables 4-14 and 4-15.

Table 4-14: Report of the classification results of the 23 classifiers trained using the proposed 4 protein signature.

		Overall (AD +MCI)		Test set AD		Test Set MCI	
	Total	AD errors	NAD errors	AD errors	NAD errors	AD errors	NAD errors
Dataset size	**139**	**64**	**75**	**42**	**50**	**22**	**25**
Classifier							
SMO	20	5	15	1	3	4	12
Simple Logistic	22	4	18	0	6	4	12
Logistic	22	4	18	0	6	4	12
Multilayer Perceptron	27.2	3.6	23.6	0.4	9.8	3.2	13.8
Bayes Net	24	11	13	1	3	10	10
Naïve Bayes	21	3	18	1	5	2	13
Naïve Bayes Simple	21	3	18	1	5	2	13
Naïve Bayes Up	21	3	18	1	5	2	13
IB1	34	12	22	6	14	6	8
Ibk	34	12	22	6	14	6	8
Kstar	26	7	19	2	10	5	9
LWL	28	15	13	6	3	9	10
AdaBoost	28	15	13	6	3	9	10
ClassViaRegression	25	8	17	1	7	7	10
Decorate	25.7	11.9	13.8	4.1	4.3	7.8	9.5
Multiclass Classifier	22	4	18	0	6	4	12
Random Committee	39.3	12	27.3	3.1	15.7	8.9	11.6
j48	25	10	15	3	5	7	10
LMT	25	9	16	0	6	9	10
NBTree	24	11	13	2	3	9	10
Part	27	15	12	10	2	5	10
Random Forest	34.2	13.5	20.7	4.2	10.1	9.3	10.6
Ordinal Classifier	25	10	15	3	5	7	10
Average	**26.10**	**8.78**	**17.32**	**2.69**	**6.56**	**6.10**	**10.76**
Agreement(%)	81%	86%	77%	94%	87%	72%	57%

Table 4-15: Overall errors for the 4 protein signature over the test sets (139 samples) compared with the results of the 5 protein signature from the Ravetti and Moscato paper.

	5 proteins		4 proteins
Classifers	**Paper**	**Re-run**	**Re-run**
SMO	19	25	20
Simple Logistic	18	21	22
Logistic	20	22	22
Multilayer Perceptron	21.6	26.7	27.2
Bayes Net	21	26	24
Naïve Bayes	19	25	21
Naïve Bayes Simple	20	24	21
Naïve Bayes Up	19	25	21
IB1	30	31	34
Ibk	30	29	34
Kstar	26	28	26
LWL	29	29	28
AdaBoost	31	38	28
ClassViaRegression	24	25	25
Decorate	21.8	34	25.7
Multiclass Classifier	20	22	22
Random Committee	26.1	34.4	39.3
j48	24	36	25
LMT	18	21	25
NBTree	21	32	24
Part	27	31	27
Random Forest	26.2	32.5	34.2
Ordinal Classifier	24	36	25
Average	**23.291**	**28.417**	**26.104**
Agreement(%)	**83%**	**80%**	**81%**

The table 4-15 compares the classification results on the test data set by using the classifiers from the 5 protein signature against the classifiers from the 4 protein signatures. The results show that although only a 4 proteins signature was used, the performance of the classifiers generally improved in their prediction of AD and NAD for both test sets, as compared with those associated with the previous 5 protein signature from WEKA selection, and are similar to those in the Ravetti and Moscato paper.

4.3 Summary

This chapter has described the steps involved in carrying out an in-depth study of the works of Ray *et al* (2007), and Ravetti and Moscato (2008). By using the information provided in their respective papers, the investigation involved: firstly to re-produce the associated experimental results and secondly, to explore if any improvements can be achieved.

With Ray *et al* (2007) study, it was very easy to repeat their experiments and to reproduce their results. A small improvement of finding a 17 protein signature which is a subset of the 18 protein signature and its classification results on the test sets are similar to those associated with 18 protein signature.

With Ravetti and Moscato (2008) study, it was not so easy to reproduce the same experimental steps and results because a different software implementation was used in the Ravetti and Moscato's steps for FS and that software was not available to use in this study. In addition, there were insufficient details in some areas of the paper and various assumptions had to be made in this study. In terms of results, similar classification results were obtained for protein signature of 18, 10, 6 and 5. Further improvement of finding of a 4 protein signature which is a subset of the 5 protein signature and the results obtained are similar to those of the 18 and 5 protein signatures.

In the next chapter, the investigations associated with the second aim of this study will be described.

5. The exploration of feature selection techniques on the accuracy of the classifiers

Different FS techniques are likely to return different subsets of features because of the different types of algorithms incorporated in them. The aim of the study is to investigate which FS techniques will return subsets that are used for training classifiers, which are able to differentiate AD from NDC or predict AD from MCI with high sensitivity and specificity. For the basis of comparison, the 18 proteins signature from Ray *et al* (2007) will be used in this study. The details of the investigation of the various WEKA FS techniques used in this study and the results of the investigation are described in the following sections.

5.1 Feature selection Analysis

Steps carried out in this study:

- Run different FS techniques to obtain 10 subsets of features, each consisting of 18 proteins. Analyse and compare them with the Ray *et al*'s 18 proteins signature (Table 5-2)
- Use WEKA classifier J4.8 and the 10 subsets of features to generate the corresponding classifiers. Use these classifiers to evaluate the test data set.
- Analyse the results.

5.1.1 Feature selection techniques used to obtain subsets of features

Data set used in this study: AD training data set.

Steps carried out in this study:

Step 1:

Load the AD training data set as shown in Figure 5-1.

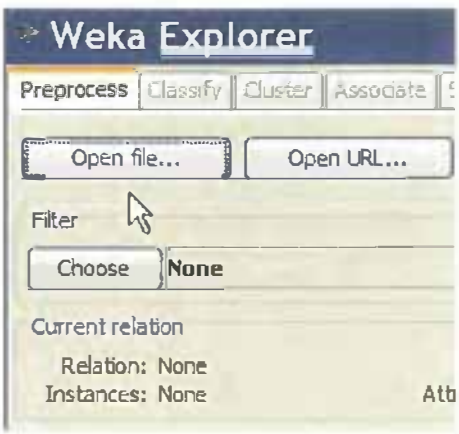


Figure 5-1: Screen shot of open file interface in WEKA Explorer.

Step 2:

Select "Select Attributes" option from menu as shown in Figure 5-2.

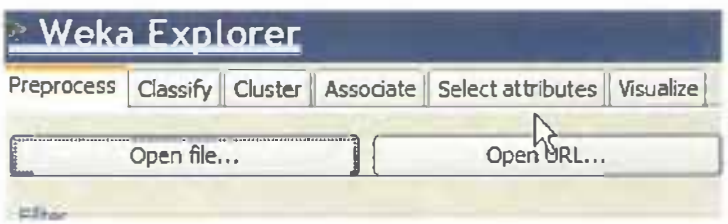


Figure 5-2: Screen shot of select attributes option.

Step 3:

Select various FS strategies from WEKA to generate 10 proteins signatures of size of 18 shown in Figure 5-3.

Figure 5-3: Screen shot of attribute selection list

Table 5-1 outlines the ten FS strategies, consisting of different combinations of search methods with feature subset evaluators. Brief descriptions of the techniques in both categories, search methods and feature subset evaluators, are found in Section 2.5.2.1, Table 2-5 and Table 2-6.

Table 5-1: WEKA FS strategies applied in this study

FS strategies	**Search method**	**Feature subset evaluator**
FS 1	Ranker	Filtered attribute evaluator
FS 2	Greedy Stepwise	Consistency subset evaluator
FS 3	Ranker	Symmetrical uncertainty attribute evaluator
FS 4	Race search	Classifier subset evaluator (logistic classifier)
FS 5	Greedy stepwise	Filtered attribute evaluator
FS 6	Ranker	Gain ratio feature evaluator
FS 7	Greedy stepwise	Classifier subset evaluator (logistic classifier)
FS 8	Ranker	OneR feature evaluator
FS 9	Ranker	Relief attribute evaluator
FS 10	Greedy stepwise	Wrapper subset evaluator (J48 classifier)

Step 4: Analyse the 10 FS of 18 proteins signature against the 18 proteins signature in the obtained in the Ray *et al* paper.

Results:

The results of subsets generated and the analysis are shown in Table 5-2.

Table 5-2: Comparison of the 10 proteins signature of size 18 with Ray *et al's*

Protein Name	Ray *et al*	FS1	FS2	FS3	FS4	FS5	FS6	FS7	FS8	FS9	FS 10
ANG-2_1	√								√		
EGF_1	√	√	√	√		√	√	√		√	
GCSF_1	√	√	√	√	√	√	√			√	
GDNF_1	√							√		√	√
ICAM-1_1	√										
IGFBP-6_1	√										
IL-1a_1	√	√	√	√	√	√	√	√	√	√	√
IL-3_1	√	√	√	√		√	√	√		√	√
IL-8_1	√										
IL-11_1	√	√	√	√	√	√	√	√	√		
MCP-3_1	√	√	√	√		√	√		√	√	
M-CSF_1	√	√		√	√				√	√	
MIP-1d_1	√	√	√	√		√	√		√		
PARC_1	√										
PDGF-BB_1	√	√		√		√	√	√		√	√
RANTES_1	√	√		√	√	√	√		√	√	
TNF-a_1	√	√	√	√		√	√	√		√	
TRAIL R4_1	√								√		
IL6_1		√		√		√	√			√	
BMP-6_1		√	√	√		√	√				√
BDNF_1		√	√		√	√		√		√	√
BLC_1		√	√			√		√			√
MCP-4_1		√		√			√				
BMP-4_1		√	√	√		√	√				√
LIGHT_1		√									
ANG_1			√			√		√			

CK b8-1_1			√	√		√	√				
CNTF_1			√			√		√			√
Eotaxin_1			√		√			√			√
Eotaxin-2_1			√							√	√
Eotaxin-3_1			√						√		
MDC_1				√			√				
LEPTIN(OB)_1				√			√		√	√	
NT-3_1					√					√	√
Lymphotactin_1					√						
TGF-b_1					√						
IL-1b_1					√						
sTNF RI_1					√						
GITR_1					√						
IL-1R4					√						
IL-15_1					√						
IGF-1 SR					√						
IGF-1_1					√						√
PIGF_1					√						
IL-12p70_1								√			
GCP-2_1								√			√
IL-7_1								√			
FGF-6_1								√			
Fit-3 Ligand_1								√			
TIMP-1_1								√			
MIG_1									√		
uPAR_1									√		
IL-1ra_1									√		
IL-10_1									√	√	

ArRP(ART)_1									√		
HCC-4_1									√		
FGF-9_1									√		
TECK_1									√		
PARC										√	
IGFBP-1_1										√	
I-309_1											√
IGFBP-2_1											√
MCP-2_1											√
GM-CSF_1											√

As shown in Table 5-2 different FS strategies returned different subsets of proteins. Refer to APPENDIX A for detailed information associated with the outputs from the 10 WEKA FS techniques.

The first column of Table 5-2 lists the name signalling proteins, the second column relates to the 18 proteins stated in the Ray *et al* paper, and columns 3 (FS1) to 12 (FS10) relate to the subsets of proteins selected from the application of WEKA FS techniques in this study. For the full details of the 120 proteins, please refer to the information found on the Nature website listed below:

http://www.nature.com/nm/journal/v13/n11/extref/nm1653-S1.pdf

5.1.2 Evaluate the 10 subsets of 18 proteins signature using J.48

Data set used to evaluate: AD test data set and MCI test data set.
Data set used to train each classifier: FS1 to FS10 as shown in Table 5-2.

Steps carried out to evaluate 10 FS selected with the AD test data set:

Step 1:
Load the first 18 protein signature (FS1) selected from FS techniques as shown in Figure 5-1. In this case FS1 is loaded.

Step 2:
Load the test AD data set as shown in Figure 5-4.

Figure 5-4: Screen shot of open test file interface in WEKA Explorer.

Step 3:

Select - J48 as a classifier to evaluate the proteins signatures shown in Figure 5-5. .

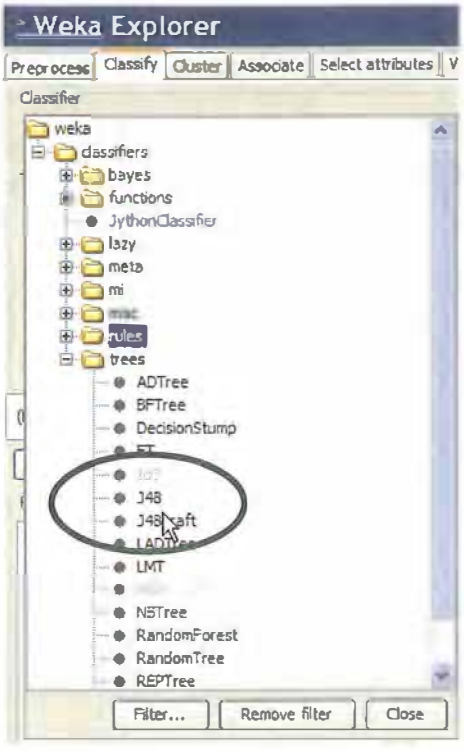


Figure 5-5: Screen shot of selecting classifier J48 interface in WEKA Explorer.

Step 4: Train the first protein signature by selecting "train" option from WEKA Explore classification interface and "start" button as shown in Figure 5-6.

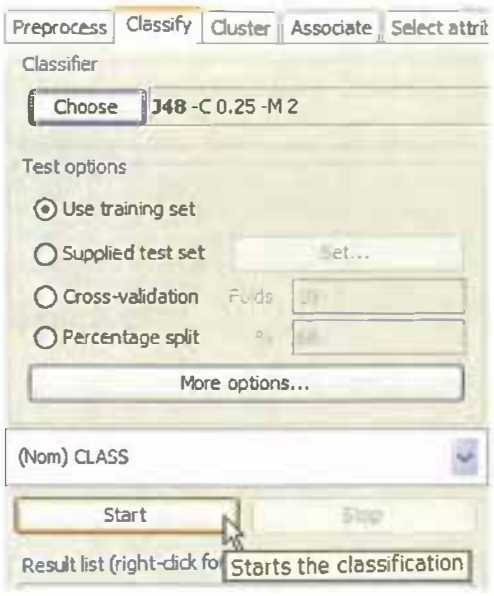


Figure 5-6: Screen shot of open test file interface in WEKA Explorer.

Step 5: Perform 10 fold cross validation by selecting "cross validation" option from WEKA Explore classification interface as shown in Figure 5-7.

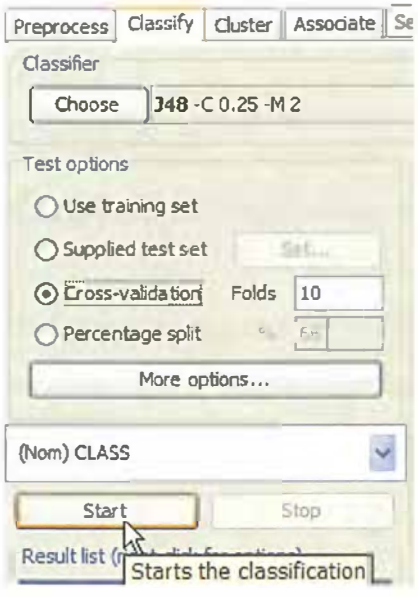


Figure 5-7: Screen shot of 10 cross validation interface in WEKA Explorer.

Step 6: Perform the test on the test AD data set by selecting "supply test set" radio button and "start" button to commence the test.

Results:

Confusion matrix was generated to show a number of prediction errors for AD and NDC as shown in Figure 5-8

```
=== Confusion Matrix ===

  a  b   <-- classified as
 37  5 |  a = AD
  2 48 |  b = NDC
```

Figure 5-8: Screen shot of confusion matrix results

Results from the confusion matrix above show 5 AD errors (5 AD were predicted as NDC) and 2 NDC errors (2 NDC were predicted as AD)

Repeat step 1, and step 3 to step 6 for evaluating the accuracy of classifiers trained using FS2 to FS10.

In terms of testing the MCI test data set, all the steps are the same as described above for evaluating with the AD test data set, the ONLY difference is in step 2: load the test MCI data set instead.

5.1.3 Results:

The results of the analysis for using the 10 sets of 18 proteins signatures are shown in table 5-3.

Table 5-3: Classification results over 10 FS of 18 protein signature with the WEKA classifier J48.

	Test set AD		Test set MCI		Overall (AD+MCI)		Total (AD+MCI)
	42	50	22	25	64	75	139
	AD errors	NAD errors	AD errors	NAD errors	AD errors	NAD errors	Total errors
FS 1	5	2	9	8	14	10	**24**
FS 2	9	18	8	14	17	32	**49**
FS 3	5	7	7	8	12	15	**27**
FS 4	9	13	8	10	17	23	**40**
FS 5	9	7	6	9	15	16	**31**
FS 6	5	7	7	8	12	15	**27**
FS 7	3	2	8	9	11	11	**22**
FS 8	6	12	8	10	14	22	**36**
FS 9	5	2	9	8	14	10	**24**
FS 10	3	2	8	9	11	11	**22**

With the different FS strategies the features selected are also different and subsequently different proteins signatures are used for training the classifiers. As shown in Table 5-3, the column "Total errors" list the total number of errors for each FS strategies. The total number of errors for the 10 different signatures that resulted from the application of different FS strategies range from 22 to 49, indicating that with different proteins signatures, the accuracy of the classification is affected and the level of impact on

the accuracy is considerable when performing classification on the same data set. For example, the set FS2 has 49 errors which are more than double the errors associated with training with FS1, FS7 and FS10. This shows the importance of selecting the most appropriate FS strategy for finding features to be incorporated for training classifiers.

5.2 Summary

This chapter described an investigation of using different FS strategies to show the importance of selecting the right strategy to obtain a subset of features for classification analysis. The results showed that there are vast differences in the accuracy of the classifiers trained using features selected by different strategies, thus addressing the second aim of this study, which was to investigate the impact on the accuracy of classification models when different FS strategies are employed.

6. Conclusion and Future work

Research into AD is an area of great interest in recent times. A lot of effort, research and money are being spent, in the hope of finding biomarkers, which can be used to diagnose the disease early as well as to develop better means to monitor treatment and a cure for the disease. The search is on for an ideal diagnostic test that is inexpensive and can be carried out easily and accurately. In addition, methods of obtaining samples for diagnosis should be simple. Researchers have been developing various molecular tests and techniques (e.g. microarrays, mass spectrometry) to address the need of finding biomarkers, resulting in many data sets which have a small sample size but are highly dimensional.

Overall, the project completed an in-depth study of both the Ray *et al*, and Ravetti and Moscato investigations. The project also improved existing results by showing that, the classification results of using 4 biomarkers obtained in this study were very similar to that obtained by using 5 biomarkers in the Ravetti and Moscato investigations. In addition, the project also investigated the applications of different FS techniques and demonstrated the importance of selecting an appropriate FS strategy for classification analysis of AD data.

Chapter 1 provided the background information, the current status of AD and the need for a simple and definitive test to diagnose the disease. The chapter also provides insights to the problems of high throughput technologies which require new analytical approaches. DM techniques is one possible approach to address this need. The significance and purpose of this study as well as research questions to be address were also clearly described.

Background and important aspects that led to the use of DM techniques as well as appropriate DM tools for analysing AD data have been discussed in Chapter 2. DM approaches used in health related area in general and AD specifically, together with the 2 previous case studies: Ray et al (2007) and Ravetti and Moscato (2008) have been examined in detail in this chapter.

Chapter 3 has provided descriptions for the data sets and research method. The data format of training and test data sets were described in detail, followed by a brief discussion about the limitations of this study.

The investigation to address the first aim of this study is described in Chapter 4. It provided information related to the experimental procedures, analysis and results associated with the in-depth study of the work of Ray *et al* and Ravetti and Moscato's experiments. Overall, the study has been able to re-create these experiments with similar results. The chapter also described work in obtaining the 4 proteins signature which is a subset of Ravetti and Moscato's 5 proteins signature. The classifiers generated using the 4 proteins signature performed as well as those trained using the 5 proteins signature.

Chapter 5 described the approached taken to address the second aim of this study. WEKA's FS strategies have been successfully used to generate 10 different sets of protein signatures. These were evaluated by using J48 and training data to produce classifiers and then to evaluate the performances of the classifiers using the test sets. The results obtained with J48 showed that there are vast differences in the accuracy of the classifiers trained using features selected by different FS strategies, thus demonstrating the

importance of selecting an appropriate FS strategy for classification analysis.

The primary purpose of this research project was to investigate the application of DM techniques for finding interesting biomarkers from a set of AD related data. The findings from this project will help to analyse the data more effectively and contribute to methods of providing earlier diagnoses of the disease.

Future Work

This study can be considered as a preliminary study into the applications of DM for mining information from health related data sets such as AD. Future work would involve:

- Using the sets of features generated in Chapter 5 and different WEKA classifiers (besides J48), produce classification models and evaluate their performances on the test data sets. In the work in Chapter 5, the sets have only been applied to J48. By evaluating with more classifiers, this will remove any bias associated with specific classification algorithms.
- Use different FS strategies, generate protein signatures of different sizes (besides 18 – obviously selected first in chapter 5 as it is the number in Ray *et al*'s paper) and then use these to train classifiers and to evaluate their classification performances on the test data sets.
- Develop specific DM algorithms tailored specifically to address characteristics of data sets in bioinformatics.

REFERENCE

ADNI (2007). Data. Retrieved May 22, 2009 from http://www.loni.ucla.edu/ADNI/Data/ADNI_Datausers.html

Adrion, W.R. (1993). Research Methodology in Software Engineering, ACM SE Notes, Jan. 1993. Retrieved April 16, 2009 from hemswell.lincoln.ac.uk/~cboldyreff/SPI/new-sp7.ppt

Aguilar-Ruiz, S. & Divina, F. (2005). Evolutionary Biclustering of Microarray Data. Application On Evaluation Computing, 1-10, doi. 10.1007/b106856

Aliferis, C., Statnikov, A. & Tsamrdinos, L. (2006). Challenges in the analysis of mass-throughput data: A technical commentary from the statistical machine learning perpective. Retrieved October 19, 2009 from http://ncbi.nlm.nih.gov/pmc/articles/PMC2675497/pdf/cin-02-133.pdf

Alzheimer's Australia NSW (2009). Death from dementia and Alzheimer's more than double in a decade: ABS. retrieved April 16, 2009 from http://alznews.blogspot.com/2009/03/deaths-from-dementia-and-alzheimers.html

Alzheimer's Disease Education and Referral (ADEAR) center (2008). Alzheimer's disease fact sheet. Retrieved April 17, 2009 from http://www.nia.nih.gov/Alzheimers/Publications/adfact.htm

Alzheimer's Disease Education and Referral (ADEAR) center (2008). General Information. Retrieved April 17, 2009 from http://www.nia.nih.gov/Alzheimers/Publications/GeneralInfo.htm

Alzheimer's Disease International (2008). Statistics. Retrieved April 7, 2009 from http://alz.co.uk/research/statistics.html

Alzheimer Society (2005). Alzheimer's Disease getting a diagnosis. Retrieved April 17, 2009 from http://alzheimer.ca/english/disease/diagnosis.htm

Amiri, M. (2003). Fuzzy C-Means Clustering. Retrieved June 3, 2009 from http://74.125.153.132/search?q=cache:Eo5l2k_LulEJ: ce.sharif.edu/~m_amiri/download/yfcmc/y_fcmc_presentation_v0.8.ppt+Fuzzy+C-means+cluster+analysis+advantages+and+disadvantages&cd=4&hl=en&ct=clnk

An, A. & Slezak, D. (2008). Foundations of intelligent systems: 17 International symposium, ISMIS 2008 Toronto, Canada, May20-23, 2008 proceedings. Lecture notes in artificial intelligence. Volume 4944 of lecture notes in computer science. Springer Berlin/Heidelberg, 4994/2008. doi: 10.1007/978-3-540-68123-6

Bertone, P. & Gerstein, M. (2001). Integrative data mining: the new direstion in bioinformatics. Retrieved February 25, 2009 from http://papers.gersteinlab.org/e-print/integ-datamine-ieee/integ-datamine-ieee.pdf

Biopharm Reports (2009). Biomarkers in Alzheimer's disease, 2009. Retrieved October 20, 2009 from http://biopharmreports.com/Report_alzheimers_biomarkers.html

Bontempi, G (2007). A Blocking Strategy to Improve Gene Selection for Classification of Gene Expression Data. IEEE/ACM Trans. Comput. Biol. Bioinformatics, 14 293-300. doi: 10.1109/TCBB.2007.1014

Borman, S. (2004, 2009). The Expectation Maximization Algorithm A short tutorial. Retrieved June 3, 2009 from http://www.seanborman.com/publications/EM_algorithm.pdf

Bodyanskiy, Y. (n.d.). Computational Intelligence techniques for data analysis. Retrieved October 22, 2009 from http://subs.emis.de/LNI/Proceedings/Proceedings72/GI-Proceedings.72-1.pdf

Breiman, L. & Cutler, A (n.d.). Random Forests. Retrieved September 16, 2009 from http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.html

Brookmeyer, R., Johnson, E., Graham, Z. K. & Arrighi, H. (2007). Forecasting the global burden of Alzheimer's disease. Retrieved May 22, 2009 from http://works.bepress.com/context/rbrookmeyer/article/1022/type/pdf/viewcontent/

Cai, Y., Feng, K., Lu, W. & Chou, K. (2005). Using LogitBoost classifier to predict protein structural classes. Retrieved October 5, 2009 from http://people.virginia.edu/~fc8b/kchou/new/paper/JTB_LogitBoost.pdf

Chen, R. & Herskovits, E. H. (2005). A Bayesian network classifier with inverse tree structure for voxelwise magnetic resonance image analysis. Research track paper, 4-12. doi: 10.1145/1081870.1081875

Christen, P. (2005). Data mining. Retrieved May 22, 2009 from http://datamining.anu.edu.au/talks/2005/datamining-comp2340-2005.8up.ps

Christinat, Y., Wachmann, B. & Zhang, L. (2008). Gene Expression Data Analysis Using a Novel Approach to Biclustering Combining Discrete and Continuous Data. IEEE/ACM Trans. Comput. Biol. Bioinformatics, 5(4) 583-593. doi: 10.1109/TCBB.2007.70251

Chu, G., Narashihan, B., Tibshirani, R. & Tusher, V. (2001). SAM "Significance Analysis of Microarrays" Users guide and technical document. Retrieved July 21, 2009 from http://www-stat.stanford.edu/~tibs/SAM/sam.pdf

Cleary, J. & Trigg, L. (1995). K*: An instance-based learner using an entropic distance measure. Retrieved October 16, 2009 from http://www.cs.waikato.ac.nz/publications/1995/Cleary95-KStar.pdf

CSIRO (2008). Preventative health cluster study fighting Alzheimer's disease. Retrieved April 16, 2009 from http:// www.csiro.au/science/AlzheimerClusterStudy--vgnexfmt-print.html

Dash, M., & Liu, H. (1997). Feature selection for classification. Issues in Intelligent Data Analysis. 1, 131-156.

Dellaert, F. (2002). The Expectation Maximization Algorithm. Retrieved June 3, 2009 from http://www.cc.gatech.edu/~dellaert/em-paper.pdf

Deng, K. & Moore, A. (1998). On the greediness of feature selection algorithms. Retrieve September 10, 2009 from http://www.ri.cmu.edu/pub_files/pub1/deng_kan_1998_2/deng_kan_1998_2.pdf

EverythingBio (n.d.). Definition of Biomarker. Retrieved October 20, 2009 from http://everythingbio.com/glos/definition.php?word=biomarker

Fayyad, U. & Irani, K. (1993). Multi-Interval discretization of continuous – Valued attributes for classification learning. Retrieved July 28, 2009 from http://sci2s.ugr.es/keel/pdf/algorithm/congreso/fayyad1993.pdf

Genetics Home Reference (2008). APP. Retrieved October 12, 2009 from http://ghr.nlm.nih.gov/gene=app

Gilman, M. (2006). A New Technology for Data Mining. Retrieved March 5, 2009 from http://www.data-mine.com/system/files/A+New+Technology+for+Data+Mining.pdf

Groth, R. (2000). *Data Mining building competitive advantage*. New Jersey: Prentice Hall.

Han, J. & Kember, M. (2006). *Data mining concepts and techniques*. India: Morgan Kaufmann.

Hastie, T., Narashihan, B., Tibshirani, R. & Chu, G. (2002). PAM "Prediction Analysis of Microarrays" Users guide and manual. Retrieved July 21, 2009 from http://www-stat.stanford.edu/~tibs/PAM/pam.pdf

Hastie, T. & Tibshirani, R. (2002). Supervised learning. Retrieved August 11, 2009 from http://www-stat.stanford.edu/~tibs/PAM/Rdist/doc.pdf

Hoffman, J. & Froemke, S. (2009). *The Alzheimer's project momentum in science*. USA: PublicAffairs.

Honkela, A. (2001). Multilayer Perceptrons. Retrieved October 16, 2009 from http://www.cis.hut.fi/ahonkela/dippa/node41.html

Hooper, C., Lovestone, S. & Fuertes, R. (2008). Alzheimer's disease, diagnosis and the need for biomarkers. Retrieved October 19, 2009 from http://www.ncbi.nih.gov/pmc/articles/PMC2688363/pdf/bmi-03-317.pdf

Jong, D. Jansen, R., Kremer, B. & Verbeek, M. (2006). Cerebrospinal fluid Amyloid β42/Phosphorylated Tau ratio discriminates between Alzheimer's disease and vascular dementia. Retrieved October 1, 2009 from http://cat.inist.fr/?aModele=afficheN&cpsidt=17995719----

KDNuggets (2006). Data Mining (Analytic) Tools (May 2006). Retrieved March 2, 2009 from http://www.kdnuggets.com/polls/2006/data_mining_analytic_tools.htm

Kim, Y., Park, I. & Lee, D. (2007). Integrated data mining strategy for effective metabolomic data analysis. Retrieved May 12, 2009 from http://www.aporc.org/LNOR/7/OSB2007F07.pdf

Kohavi, R. & John, G. (1997). Wrappers for feature selection. Retrieved September 10, 2009 from http://dx.doi.org/10.1016/S0004-3702(97)00043-X

Kohavi, R. & Quinlan, R. (1999). Decision tree discovery. Retrieved October 10, 2009 from http://ai.stanford.edu/~ronnyk/treesHB.pdf

Kononenko, I. (1993) Inductive and Bayesian learning in medical diagnosis. Issues in Applied Artificial Intelligence. 7[2], 317-337

Landwehr, N. Hall, M. & Frank, E. (2004). Logistic Model Tree. Retrieved October 16, 2009 from http://www.cs.waikato.ac.nz/!ml/publications/2003/landwehr-etal.pdf

Lee-Fryer, B. (2009). Current and future uses of MRI in Alzheimer's diagnosis. Retrieved September 30, 2009 from http://alzheimers.about.com/w/Health-Medicine/Conditions-and-diseases/MRIs-and-Alzheimers-Disease.htm

Levy, S., Statnikov, A. & Aliferis, C. (2005). Biomaker selection from high-dimensionality data. Retrieved June 3, 2009 from www.statnikov.org/publications/ Pharmaceutical_Discovery_2005.pdf

Madeira, S. C. & Oliveira, A. L. (2004). Biclustering algorithms for biological data analysis: a survey. Retrieved February 25, 2009 from www.inesc-id.pt/ficheiros/publicacoes/1281.pdf

Markov, Z. & Russel, I. (2006). An introduction to the WEKA Data mining system. Retrieved March 1, 2009 from http://www.cs.ccsu.edu/~markov/weka-tutorial.pdf

Matteucci, M. (n.d.). A Tutorial on clustering algorithms - Clustering as a mixture of Gaussians. Retrieved September 16, 2009 from http:// home.dei.polimi.it/matteucc/Clustering/tutorial_html/ hierarchical.html

Matteucci, M. (n.d.). A Tutorial on clustering algorithms – Fuzzy C-Means clustering. Retrieved September 16, 2009 from http:// home.dei.polimi.it/matteucc/Clustering/tutorial_html/ cmeans.html

Matteucci, M. (n.d.). A Tutorial on clustering algorithms – K-Means clustering. Retrieved September 16, 2009 from http:// home.dei.polimi.it/matteucc/Clustering/tutorial_html/ kmeans.html

Matteucci, M. (n.d). A Tutorial on Clustering Algorithms - Index. Retrieved June 3, 2009 from http://home.dei.polimi.it/matteucc/Clustering/ tutorial_html/index.html

McCorquodale, D. & Myers, A. (2008). Biomarkers in the diagnosis and treatement of Alzheimer's disease: potential and pitfalls. Retrieved June 10, 2009 from http://www.futuremedicine.com/doi/pdf/10.2217/17520363.2.3.209?cookieSet=1

Medline Plus (2009). CT Scans. Retrieved October 12, 2009 from http://www.nlm.nih.gov/medlineplus/ctscans.html

Medline Plus (2009). Nuclear scans. Retrieved October 12, 2009 from http://www.nlm.nih.gov/medlineplus/nuclearscans.html

Melville, P. & Mooney, R. (2004). Diverse ensembles for active learning. Retrieved October 18, 2009 from http://www.cs.utexas.edu/users/ml/papers/decorate-icml-04.pdf

Microsoft.com (2000). Sequential minimal optimization. Retrieved October 2, 2009 from http://research.microsoft.com/pubs/68391/smo-book.pdf

Mitchell, T. (2005). Generative and discriminative classifiers: Naive bayes and logistic regression. Retrieved October 17, 2009 from http://www.cs.cmu.edu/~tom/mbook/NBayesLogReg.pdf

Ng, R. & Pei, J. (2007). Introduction to the special issue on data mining for health informatics. Retrieved May 10, 2009 from http://www.Sigkdd.org/explorations/issues/9-1-2007/Intro.pdf

Nuzzo, R. (2007). Using algorithms to tackle Alzheimer's Disease. Retrieved March 25, 2009 from http://www.biomedicalcomputationreview.org/3/4/5.pdf

Park, J., Li, S & Kricka, L. (2005). Clinical Chemistry. Retrieved September 28, 2009 from http://www.clinchem.org/cgi/reprint/52/2/332

Park, S. & Furnkranz, J. (n.d.). Efficient pairwise classification. Retrieved October 4, 2009 from http://www.ke.tu-darmstadt.de/~juffi/publications/ecml-07-EfficientPairwise.pdf

Portinale, L. & Saitta, L. (2002). Feature selection. Retrieved September 10, 2009 from http://www-ai.informatik.uni-dortmund.de/DOKUMENTE/portinale_saitta_2002a.pdf

Pujari, A. K. (2001). *Data mining techniques*. Hyderabad, India: Universities Press.

Qi Tan, Y. L.,Neary, J. Lui, F. & Wang, Y. (2008). Adaptive discriminant analysis for microarray-based classification. ACM Trans. Knowl. Discov. Data 2(1) 1-20. doi: 10.1145/1342320.1342325

Ravetti, M. & Moscato, P. (2008). Identification of a 5-protein biomarker molecular signature for predicting Alzheimer's disease. Retrieved July 17, 2009 from http://www.plosone.org/article/info:doi/10.1371/journal.pone.0003111

Ray, D., Britschgi, M, Herbert, C., Uchimura, Y. T., Boxer, A. Blennow, K., *et al* (2007). Classification and prediction of clinical Alzheimer's diagnosis based on plasma signaling proteins. Nature Medicine 13, 1359 – 1362. doi: 10.1038/nm1653

Rudjer Boskovic Institute (2001). Overview of Data Mining applications. Retrieved April 16, 2009 from http://dms.irb.hr/tutorial/tut_dm_applic_overview.php

Saeys, Y, Inza, I. & Larranaga, P. (2007). A review of feature selection techniques in bioinformatics. Retrieved September 9, 2009 from http://bioinformatics.oxfordjournals.org/cgi/reprint/btm344v1.pdf

Salzberg, S. (1994). Book review: C4.5: Programs for machine learning. Retrieved October 16, 2009 from http://springerlink.com/content/v986m1562062hk51/fulltext.pdf

ScienceDaily (2009). Biomarkers may help predict risk of Alzheimer's disease in patients with mild cognitive impairment. Retrieved October 5, 2009 from http://www.sceincedaily.com/releases/2009/07/090721163106.htm

Seung, S. (2002). Multilayer perceptrons and backpropagation learning. Retrieved October 10, 2009 from http://hebb.mit.edu/courses/9.641/2002/lectures/lecture04.pdf

Tan, P.N., Steinbach, M. & Kumar, V. (2006). *Introduction to data mining*. (China ed.) China: Pearson Education Asia Ltd and Post & Telecom Press.

Tanagra 1.4.31 [Computer software]. (2003). French

Tax, D. & Duin, R. (2002). Using two-class classifiers for multiclass classification. Retrieved October 16, 2009 from http://ict.ewi.tudelft.nl/~davidt/papers/icpr_02_mclass.pdf

The Medical News (2007). Global prevalence of Alzheimer's disease will grow to 106 million by 2050. Retrieved May 23, 2009 from http://www.news-medical.net/news/2007/06/14/26238.aspx

Trojanowskl, J. Q. (2004). Biomarkers of Alzheimer's. Retrieved May 22, 2009 from http://www.loni.ucla.edu/ADNI/About/BioMarker.pdf

Vemuri, P. *et al* (2009). MRI and CSF biomarkers in normal, MCI, and AD subjects. Retrieved October 5, 2009 from http://neurology.org/cgi/content/abstract/73/4/287

Walker, P. R., Smith, B., Liu, Q. Y., Famili, F., Valdes, J. Lui, Z. & Lach, B. (2003). Data mining of gene expression changes in Alzheimer brain. Retrieved March 12, 2009 from http://iit-iti.nrc-cnrc.gc.ca/iit-publications-iti/docs/NRC-45838.pdf

WEKA 3.6.1 [Computer software]. (2009). Hamilton, New Zealand.

Wie, V. (n.d.). Significance Analysis of Microarrays using Rank Scores. Retrieved August 17, 2009 from http://alexandria.tue.nl/repository/books/556087.pdf

Witten, I. & Frank, E. (2005). *Data mining practical machine learning tools and techniques* (China 2nd ed.). China: China Machine Press

Yang, J. & Honavar, V. (1997). Feature subset selection using a generic algorithm. Retrieved September 10, 2009 from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.50.6333&rep=rep1&type=pdf

Yu, L.T.H., Chung, F.L., Chan, S.C.F. & Yuen, S.M.C. (2004). Using emerging pattern based projected clustering and gene expression data for cancer detection. Retrieved February 25, 2009 from http://crpit.com/confpapers/CRPITV29Yu.pdf

Zalik, K. (2006). Fuzzy C-Means clustering and facility location problems. Retrieved October 9, 2009 from http://www.actpress.com/PaperInfo.aspx?PaperID=28189&reason=500

APPENDIX A

Feature selection set 1 (FS1)

=== Run information ===

Evaluator: weka.attributeSelection.FilteredAttributeEval -W "weka.attributeSelection.InfoGainAttributeEval " -F "weka.filters.supervised.instance.SpreadSubsample -M 0.0 -X 0.0 -S 1"
Search: weka.attributeSelection.Ranker -T -1.7976931348623157E308 -N 18
Relation: Rayetal-trainingset R&C reversed-CVS
Instances: 83
Attributes: 121
[list of attributes omitted]
Evaluation mode: evaluate on all training data

=== Attribute Selection on all input data ===

Search Method:
Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 121 CLASS):
Filtered Attribute Evaluator
Filter: weka.filters.supervised.instance.SpreadSubsample -M 0.0 -X 0.0 -S 1
Attribute evaluator: weka.attributeSelection.InfoGainAttributeEval

Ranked attributes:
0.322 29 IL-1a_1
0.171 36 IL-6_1
0.166 77 GCSF_1
0.165 42 MCP-3_1
0.155 59 TNF-a_1
0.15 90 IL-11_1
0.134 53 RANTES_1
0.133 33 IL-3_1
0.131 8 EGF_1
0.13 47 MIP-1d_1
0.124 52 PDGF-BB_1
0.112 5 BMP-6_1

0 2 BDNF_1
0 3 BLC_1
0 44 M-CSF_1
0 43 MCP-4_1
0 4 BMP-4_1
0 39 LIGHT_1

Selected attributes:
29,36,77,42,59,90,53,33,8,47,52,5,2,3,44,43,4,39 : 18

Feature selection set 2 (FS2)

=== Run information ===

Evaluator: weka.attributeSelection.ConsistencySubsetEval
Search: weka.attributeSelection.GreedyStepwise -R -T -1.7976931348623157E308 -N 18
Relation: Rayetal-trainingset R&C reversed-CVS
Instances: 83
Attributes: 121
[list of attributes omitted]
Evaluation mode: evaluate on all training data

=== Attribute Selection on all input data ===

Search Method:
Greedy Stepwise (forwards).
Start set: no attributes

Attribute Subset Evaluator (supervised, Class (nominal): 121 CLASS):
Consistency Subset Evaluator

Ranked attributes:
0.819 29 IL-1a_1
0.867 33 IL-3_1
0.904 5 BMP-6_1
0.916 77 GCSF_1
0.94 59 TNF-a_1
0.94 1 ANG_1
0.94 2 BDNF_1
0.94 3 BLC_1
0.94 4 BMP-4_1
0.94 6 CK b8-1_1

0.94 7 CNTF_1
0.94 8 EGF_1
0.952 47 MIP-1d_1
0.964 42 MCP-3_1
0.988 90 IL-11_1
0.988 9 Eotaxin_1
0.988 10 Eotaxin-2_1
0.988 11 Eotaxin-3_1

Selected attributes: 29,33,5,77,59,1,2,3,4,6,7,8,47,42,90,9,10,11 : 18

Feature selection set 3 (FS3)

=== Run information ===

Evaluator: weka.attributeSelection.SymmetricalUncertAttributeEval
Search: weka.attributeSelection.Ranker -T -1.7976931348623157E308 -N 18
Relation: Rayetal-trainingset R&C reversed-CVS
Instances: 83
Attributes: 121
[list of attributes omitted]
Evaluation mode: evaluate on all training data

=== Attribute Selection on all input data ===

Search Method:
Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 121 CLASS):
Symmetrical Uncertainty Ranking Filter

Ranked attributes:
0.322 29 IL-1a_1
0.215 36 IL-6_1
0.199 59 TNF-a_1
0.17 77 GCSF_1
0.167 52 PDGF-BB_1
0.165 42 MCP-3_1
0.162 53 RANTES_1
0.15 90 IL-11_1

0.15 5 BMP-6_1
0.149 33 IL-3_1
0.136 47 MIP-1d_1
0.134 8 EGF_1
0 4 BMP-4_1
0 45 MDC_1
0 43 MCP-4_1
0 44 M-CSF_1
0 38 LEPTIN(OB)_1
0 6 CK b8-1_1

Selected attributes:
29,36,59,77,52,42,53,90,5,33,47,8,4,45,43,44,38,6 : 18

Feature selection set 4 (FS4)

=== Run information ===

Evaluator: weka.attributeSelection.ClassifierSubsetEval -B weka.classifiers.functions.Logistic -T -H "Click to set hold out or test instances" -- -R 1.0E-8 -M -1
Search: weka.attributeSelection.RaceSearch -R 0 -L 0.0010 -T 0.0010 -F 0 -Q -N 18 -J -1.7976931348623157E308 -A weka.attributeSelection.GainRatioAttributeEval --
Relation: Rayetal-trainingset R&C reversed-CVS
Instances: 83
Attributes: 121
[list of attributes omitted]
Evaluation mode: evaluate on all training data

=== Attribute Selection on all input data ===

Search Method:
RaceSearch.
Race type : forward selection race
Base set : no attributes
Cross validation mode : 10 fold
Merit of best subset found : 0.347

Attribute Subset Evaluator (supervised, Class (nominal): 121 CLASS):
Classifier Subset Evaluator
Learning scheme: weka.classifiers.functions.Logistic

Scheme options: -R 1.0E-8 -M -1
Hold out/test set: Training data
Accuracy estimation: classification error

Ranked attributes:
 0.387   29 IL-1a_1
 0.296   90 IL-11_1
 0.263   50 NT-3_1
 0.256   99 Lymphotactin_1
 0.221   57 TGF-b_1
 0.201   44 M-CSF_1
 0.202   77 GCSF_1
 0.191   30 IL-1b_1
 0.164   110 sTNF RI_1
 0.135   78 GITR_1
 0.103   94 IL-1R4 /ST2_1
 0.109   27 IL-15_1
 0.12   86 IGF-1 SR
 0.113   2 BDNF_1
 0.115   53 RANTES_1
 0.132   21 IGF-1_1
 0.164   108 PIGF_1
 0.112   9 Eotaxin_1

Selected attributes:
29,90,50,99,57,44,77,30,110,78,94,27,86,2,53,21,108,9 : 18

Feature selection set 5 (FS5)

=== Run information ===

Evaluator: weka.attributeSelection.FilteredSubsetEval -W
"weka.attributeSelection.CfsSubsetEval " -F
"weka.filters.supervised.instance.SpreadSubsample -M 0.0 -X 0.0 -
S 1"
Search: weka.attributeSelection.GreedyStepwise -R -T -
1.7976931348623157E308 -N 18
Relation: Rayetal-trainingset R&C reversed-CVS
Instances: 83
Attributes: 121
              [list of attributes omitted]
Evaluation mode: evaluate on all training data

=== Attribute Selection on all input data ===

Search Method:
	Greedy Stepwise (forwards).
	Start set: no attributes

Attribute Subset Evaluator (supervised, Class (nominal): 121
CLASS):
Filtered Attribute Evaluator
Filter: weka.filters.supervised.instance.SpreadSubsample -M 0.0 -X
0.0 -S 1
Attribute evaluator: weka.attributeSelection.CfsSubsetEval

Ranked attributes:
 0.322   29 IL-1a_1
 0.368   36 IL-6_1
 0.392   52 PDGF-BB_1
 0.413   59 TNF-a_1
 0.428   77 GCSF_1
 0.44   90 IL-11_1
 0.451   5 BMP-6_1
 0.458   42 MCP-3_1
 0.467   53 RANTES_1
 0.472   8 EGF_1
 0.475   33 IL-3_1
 0.479   47 MIP-1d_1
 0.317   1 ANG_1
 0.25   2 BDNF_1
 0.211   3 BLC_1
 0.185   4 BMP-4_1
 0.165   6 CK b8-1_1
 0.15   7 CNTF_1

Selected attributes: 29,36,52,59,77,90,5,42,53,8,33,47,1,2,3,4,6,7
: 18

Feature selection set 6 (FS6)

=== Run information ===

Evaluator: weka.attributeSelection.GainRatioAttributeEval
Search: weka.attributeSelection.Ranker -T -
1.7976931348623157E308 -N 18
Relation: Rayetal-trainingset R&C reversed-CVS
Instances: 83

Attributes: 121
[list of attributes omitted]
Evaluation mode: evaluate on all training data

=== Attribute Selection on all input data ===

Search Method:
Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 121 CLASS):
Gain Ratio feature evaluator

Ranked attributes:
0.322 29 IL-1a_1
0.288 36 IL-6_1
0.275 59 TNF-a_1
0.251 52 PDGF-BB_1
0.226 5 BMP-6_1
0.204 53 RANTES_1
0.173 77 GCSF_1
0.168 33 IL-3_1
0.166 42 MCP-3_1
0.15 90 IL-11_1
0.143 47 MIP-1d_1
0.137 8 EGF_1
0 4 BMP-4_1
0 6 CK b8-1_1
0 45 MDC_1
0 43 MCP-4_1
0 44 M-CSF_1
0 38 LEPTIN(OB)_1

Selected attributes:
29,36,59,52,5,53,77,33,42,90,47,8,4,6,45,43,44,38 : 18

Feature selection set 7 (FS7)

=== Run information ===

Evaluator: weka.attributeSelection.ClassifierSubsetEval -B weka.classifiers.functions.Logistic -T -H "Click to set hold out or test instances" -- -R 1.0E-8 -M -1
Search: weka.attributeSelection.GreedyStepwise -R -T -1.7976931348623157E308 -N 18
Relation: Rayetal-trainingset R&C reversed-CVS

```
Instances:    83
Attributes:   121
              [list of attributes omitted]
Evaluation mode:    evaluate on all training data

=== Attribute Selection on all input data ===

Search Method:
        Greedy Stepwise (forwards).
        Start set: no attributes

Attribute Subset Evaluator (supervised, Class (nominal): 121
CLASS):
        Classifier Subset Evaluator
        Learning scheme: weka.classifiers.functions.Logistic
        Scheme options: -R 1.0E-8 -M -1
        Hold out/test set: Training data
        Accuracy estimation: classification error

Ranked attributes:
-0.1807    29 IL-1a_1
-0.1566    52 PDGF-BB_1
-0.1205    92 IL-12 p70_1
-0.0964    33 IL-3_1
-0.0964    16 GCP-2_1
-0.0843     2 BDNF_1
-0.0723    90 IL-11_1
-0.0602     8 EGF_1
-0.0482    37 IL-7_1
-0.0482     9 Eotaxin_1
-0.0482    12 FGF-6_1
-0.0482     7 CNTF_1
-0.0482    14 Fit-3 Ligand_1
-0.0482    17 GDNF_1
-0.0361   113 TIMP-1_1
-0.0241    59 TNF-a_1
 0          3 BLC_1
 0          1 ANG_1

Selected attributes:
29,52,92,33,16,2,90,8,37,9,12,7,14,17,113,59,3,1 : 18
```

Feature selection set 8 (FS8)

```
=== Run information ===

Evaluator:     weka.attributeSelection.OneRAttributeEval -S 1 -F 10 -
B 6
Search:        weka.attributeSelection.Ranker -T -
1.7976931348623157E308 -N 18
Relation:      Rayetal-trainingset R&C reversed-CVS
Instances:     83
Attributes:    121
               [list of attributes omitted]
Evaluation mode:     evaluate on all training data


=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 121 CLASS):
        OneR feature evaluator.

        Using 10 fold cross validation for evaluating attributes.
        Minimum bucket size for OneR: 6

Ranked attributes:
79.5181    29 IL-1a_1
71.0843    53 RANTES_1
69.8795    44 M-CSF_1
69.8795    46 MIG_1
68.6747    42 MCP-3_1
67.4699   118 uPAR_1
67.4699    31 IL-1ra_1
67.4699    25 IL-10_1
67.4699    90 IL-11_1
62.6506    38 LEPTIN(OB)_1
62.6506    47 MIP-1d_1
62.6506    62 AgRP(ART)_1
62.6506    11 Eotaxin-3_1
62.6506    82 HCC-4_1
61.4458    76 FGF-9_1
61.4458   117 TRAIL R4_1
61.4458   112 TECK_1
61.4458    63 ANG-2_1
```

Selected attributes:
29,53,44,46,42,118,31,25,90,38,47,62,11,82,76,117,112,63 : 18

Feature selection set 9 (FS9)

=== Run information ===

Evaluator: weka.attributeSelection.ReliefFAttributeEval -M -1 -D 1 -K 10
Search: weka.attributeSelection.Ranker -T -1.7976931348623157E308 -N 18
Relation: Rayetal-trainingset R&C reversed-CVS
Instances: 83
Attributes: 121
[list of attributes omitted]
Evaluation mode: evaluate on all training data

=== Attribute Selection on all input data ===

Search Method:
Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 121 CLASS):
ReliefF Ranking Filter
Instances sampled: all
Number of nearest neighbours (k): 10
Equal influence nearest neighbours

Ranked attributes:
0.0826 29 IL-1a_1
0.0755 17 GDNF_1
0.0722 59 TNF-a_1
0.0486 77 GCSF_1
0.0453 53 RANTES_1
0.0377 2 BDNF_1
0.0357 44 M-CSF_1
0.0326 52 PDGF-BB_1
0.0325 38 LEPTIN(OB)_1
0.0316 8 EGF_1
0.0296 10 Eotaxin-2_1
0.0283 33 IL-3_1
0.0266 51 PARC_1
0.0248 22 IGFBP-1_1
0.0244 36 IL-6_1
0.0242 25 IL-10_1

0.0237 50 NT-3_1
0.0233 42 MCP-3_1

Selected attributes:
29,17,59,77,53,2,44,52,38,8,10,33,51,22,36,25,50,42 : 18

Feature selection set 10 (FS10)

=== Run information ===

Evaluator: weka.attributeSelection.WrapperSubsetEval -B weka.classifiers.trees.J48 -F 5 -T 0.01 -R 1 -- -C 0.25 -M 2
Search: weka.attributeSelection.GreedyStepwise -R -T -1.7976931348623157E308 -N 18
Relation: Rayetal-trainingset R&C reversed-CVS
Instances: 83
Attributes: 121
[list of attributes omitted]
Evaluation mode: evaluate on all training data

=== Attribute Selection on all input data ===

Search Method:
Greedy Stepwise (forwards).
Start set: no attributes

Attribute Subset Evaluator (supervised, Class (nominal): 121 CLASS):
Wrapper Subset Evaluator
Learning scheme: weka.classifiers.trees.J48
Scheme options: -C 0.25 -M 2
Accuracy estimation: classification error
Number of folds for accuracy estimation: 5

Ranked attributes:
-0.205 29 IL-1a_1
-0.157 52 PDGF-BB_1
-0.118 33 IL-3_1
-0.116 16 GCP-2_1
-0.116 19 I-309_1
-0.116 21 IGF-1_1
-0.116 23 IGFBP-2_1
-0.116 9 Eotaxin_1
-0.111 41 MCP-2_1
-0.108 50 NT-3_1

-0.108 3 BLC_1
-0.108 2 BDNF_1
-0.108 4 BMP-4_1
-0.108 5 BMP-6_1
-0.108 7 CNTF_1
-0.108 10 Eotaxin-2_1
-0.108 17 GDNF_1
-0.108 18 GM-CSF_1

Selected attributes:
29,52,33,16,19,21,23,9,41,50,3,2,4,5,7,10,17,18 : 18

APPENDIX B

The following are some screen snapshots of WEKA to illustrate the program has many menu options, classifiers, feature selection methods, test options and output result formats available in different user interfaces.

Figure 1: Explorer interface with different menu options

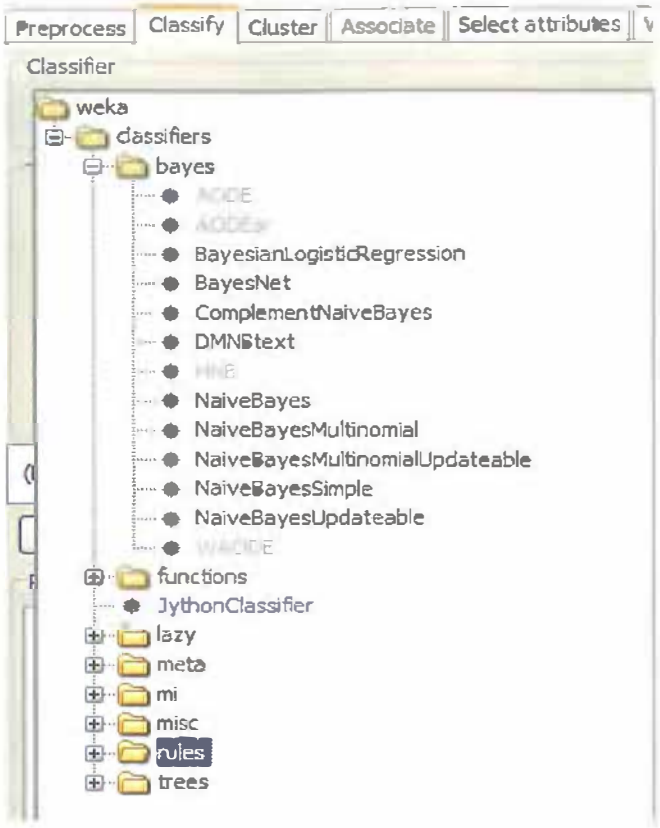


Figure 2: Explorer with many classifiers provided.

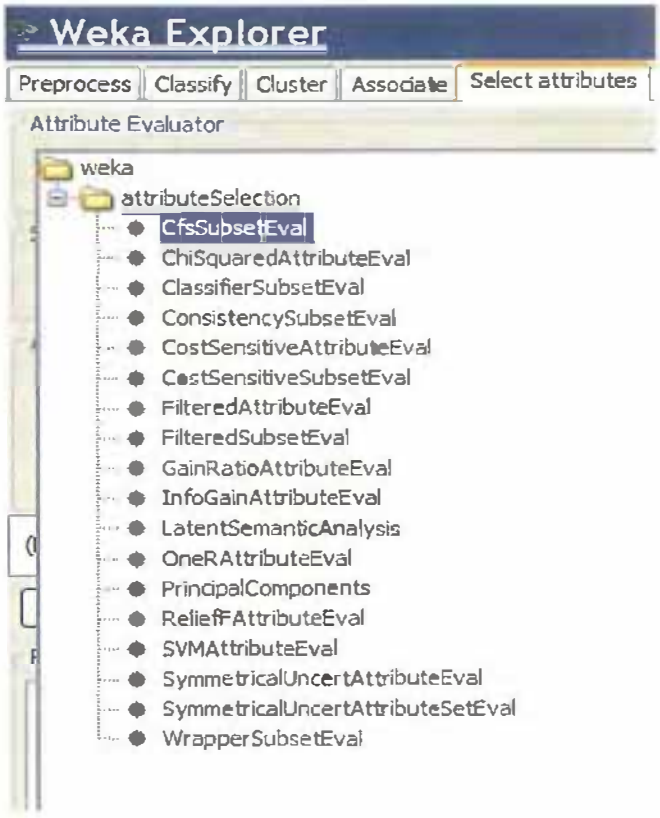


Figure 3: Feature selection with many attribute evaluators

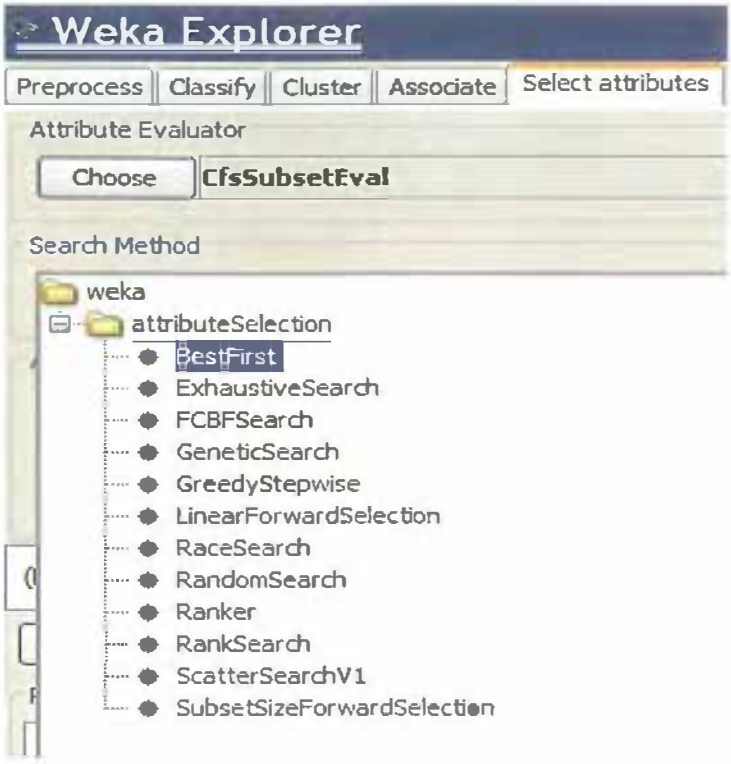


Figure 4: Feature selection with many search methods

```
=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances          81               97.5904 %
Incorrectly Classified Instances         2                2.4096 %
Kappa statistic                          0.9518
Mean absolute error                      0.0395
Root mean squared error                  0.1405
Relative absolute error                  7.9045 %
Root relative squared error             28.1154 %
Total Number of Instances               83

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.953     0          1         0.953     0.976      0.988    AD
                 1         0.047      0.952     1         0.976      0.988    NDC
Weighted Avg.    0.976     0.022      0.977     0.976     0.976      0.988

=== Confusion Matrix ===

  a  b   <-- classified as
 41  2 |  a = AD
  0 40 |  b = NDC
```

Figure 5: View output of classification in text form

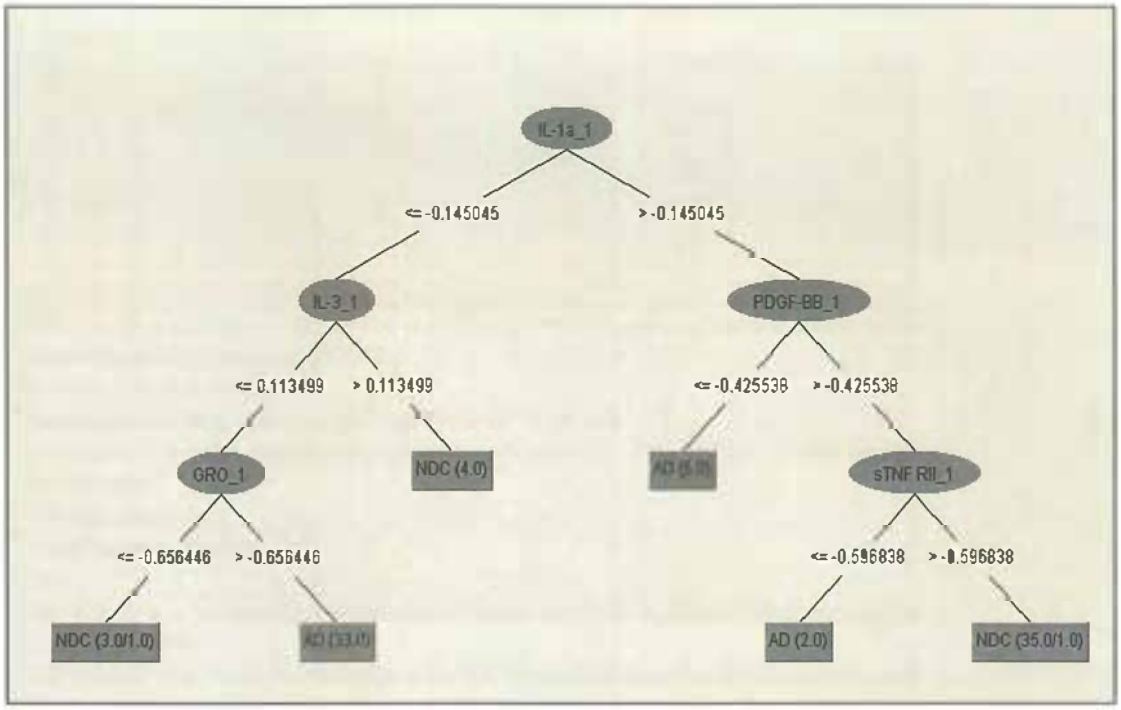


Figure 6: View output of classification in graphics form

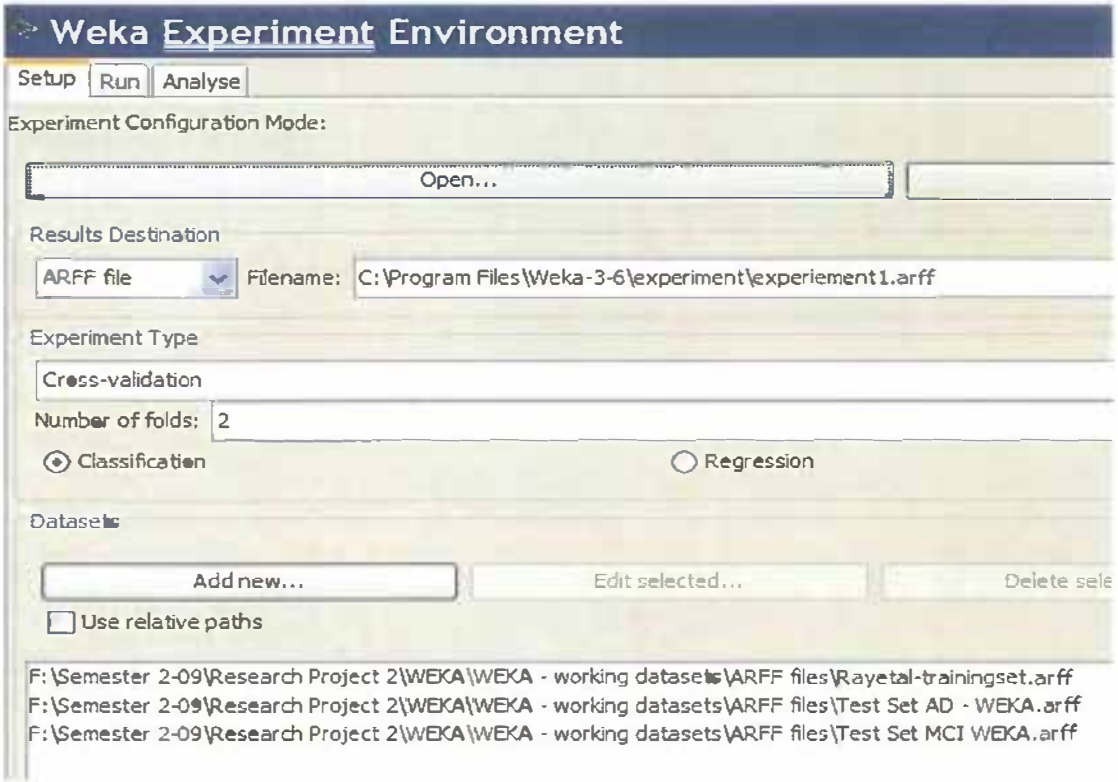


Figure 7: Experiment Environment with multiple data sets setup

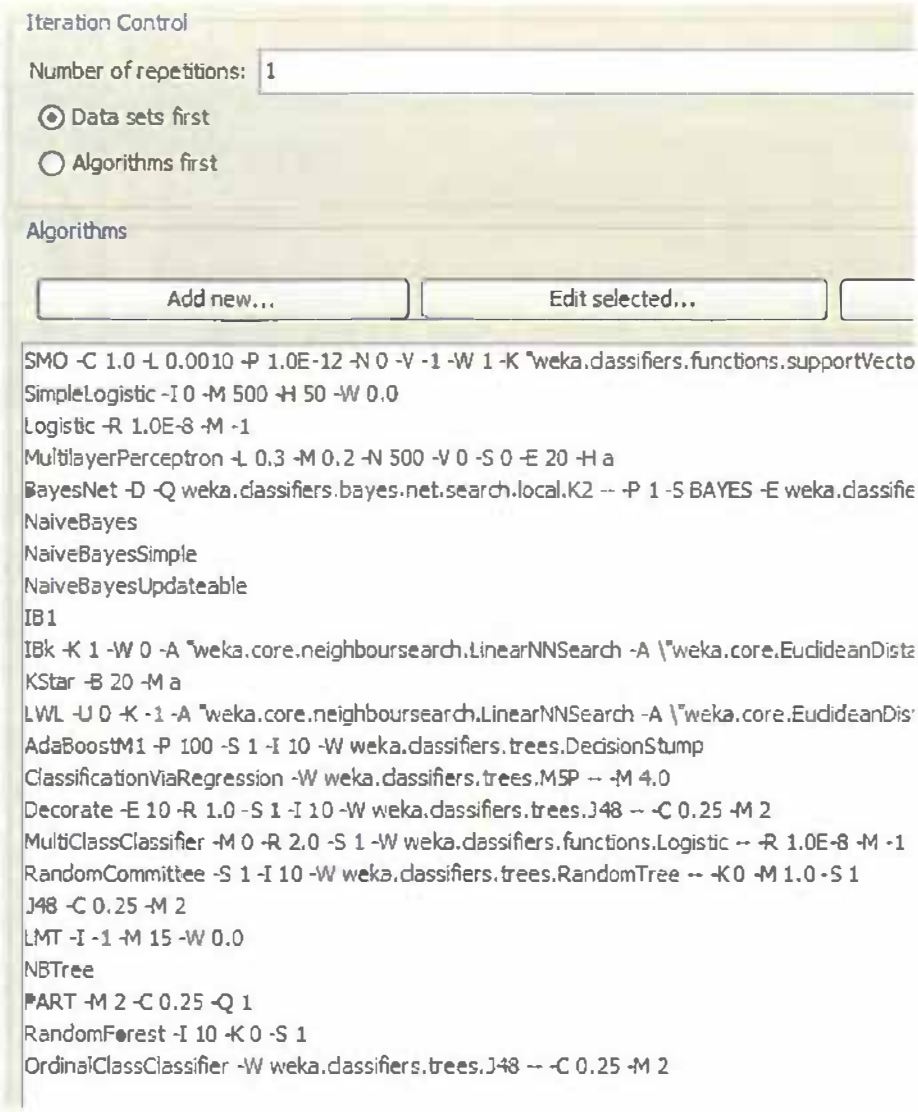


Figure 8: Experiment Environment with multiple classifiers setup

APPENDIX C

The following references are extracted from WEKA 3.6.1 (2009):

SMO

"J. Platt: Machines using Sequential Minimal Optimization. In B. Schoelkopf and C. Burges and A. Smola, editors, Advances in Kernel Methods - Support Vector Learning, 1998."

S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, K.R.K. Murthy (2001). Improvements to Platt's SMO Algorithm for SVM Classifier Design. Neural Computation. 13(3):637-649."

Logistic

"le Cessie, S., van Houwelingen, J.C. (1992). Ridge Estimators in Logistic Regression. Applied Statistics. 41(1):191-201."

Simple Logistic

"Niels Landwehr, Mark Hall, Eibe Frank (2005). Logistic Model Trees. Marc Sumner, Eibe Frank, Mark Hall: Speeding up Logistic Model Tree Induction. In: 9th European Conference on Principles and Practice of Knowledge Discovery in Databases, 675-683, 2005."

Bayes Net

"http://www.cs.waikato.ac.nz/~remco/weka.pdf"

Naïve Bayes

"George H. John, Pat Langley: Estimating Continuous Distributions in Bayesian Classifiers. In: Eleventh Conference on Uncertainty in Artificial Intelligence, San Mateo, 338-345, 1995."

Naïve Bayes Simple

"*Richard Duda, Peter Hart (1973). Pattern Classification and Scene Analysis. Wiley, New York.*"

Naïve Bayes Updatable

"*George H. John, Pat Langley: Estimating Continuous Distributions in Bayesian Classifiers. In: Eleventh Conference on Uncertainty in Artificial Intelligence, San Mateo, 338-345, 1995.*"

IB1

"*D. Aha, D. Kibler (1991). Instance-based learning algorithms. Machine Learning. 6:37-66.*"

IBk

"*D. Aha, D. Kibler (1991). Instance-based learning algorithms. Machine Learning. 6:37-66.*"

KStar

"*John G. Cleary, Leonard E. Trigg: K*: An Instance-based Learner Using an Entropic Distance Measure. In: 12th International Conference on Machine Learning, 108-114, 1995.*"

LWL

"*Eibe Frank, Mark Hall, Bernhard Pfahringer: Locally Weighted Naive Bayes. In: 19th Conference in Uncertainty in Artificial Intelligence, 249-256, 2003.*

C. Atkeson, A. Moore, S. Schaal (1996). Locally weighted learning. AI Review."

AdaBoost

"Yoav Freund, Robert E. Schapire: Experiments with a new boosting algorithm. In: Thirteenth International Conference on Machine Learning, San Francisco, 148-156, 1996."

Classification Via Regression

"E. Frank, Y. Wang, S. Inglis, G. Holmes, I.H. Witten (1998). Using model trees for classification. Machine Learning. 32(1):63-76."

Decorate

"P. Melville, R. J. Mooney: Constructing Diverse Classifier Ensembles Using Artificial Training Examples. In: Eighteenth International Joint Conference on Artificial Intelligence, 505-510, 2003.

P. Melville, R. J. Mooney (2004). Creating Diversity in Ensembles Using Artificial Data. Information Fusion: Special Issue on Diversity in Multiclassifier Systems."

Ordinal classifier

"Eibe Frank, Mark Hall: A Simple Approach to Ordinal Classification. In: 12th European Conference on Machine Learning, 145-156, 2001."

PART

"Eibe Frank, Ian H. Witten: Generating Accurate Rule Sets Without Global Optimization. In: Fifteenth International Conference on Machine Learning, 144-151, 1998."

J48

"Ross Quinlan (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA."

LMT

"*Niels Landwehr, Mark Hall, Eibe Frank (2005). Logistic Model Trees. Machine Learning. 95(1-2):161-205.*

Marc Sumner, Eibe Frank, Mark Hall: Speeding up Logistic Model Tree Induction. In: 9th European Conference on Principles and Practice of Knowledge Discovery in Databases, 675-683, 2005."

NBTree

"*Ron Kohavi: Scaling Up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid. In: Second International Conference on Knoledge Discovery and Data Mining, 202-207, 1996.*"

Random Forest

"Leo Breiman (2001). Random Forests. Machine Learning. 45(1):5-32."