PRACTICAL APPROACHES TO MINING OF CLINICAL DATASETS: FROM FRAMEWORKS TO NOVEL FEATURE SELECTION

By

Nongnuch Poolsawad

This thesisis submitted in partial fulfilment of the requirements

for the degree of Doctor of Philosophy

in Computer Science

Department of Computer Science

The University of Hull

May 2014

ABSTRACT

Research has investigated clinical data that have embedded within them numerous complexities and uncertainties in the form of missing values, class imbalances and high dimensionality. The research in this thesis was motivated by these challenges to minimise these problems whilst, at the same time, maximising classification performance of data and also selecting the significant subset of variables. As such, this led to the proposal of a data mining framework and feature selection method. The proposed framework has a simple algorithmic framework and makes use of a modified form of existing frameworks to address a variety of different data issues, called the Handling Clinical Data Framework (HCDF). The assessment of data mining techniques reveals that missing values imputation and resampling data for class balancing can improve the performance of classification. Next, the proposed feature selection method was introduced; it involves projecting onto principal component method (FS-PPC) and draws on ideas from both feature extraction and feature selection to select a significant subset of features from the data. This method selects features that have high correlation with the principal component by applying symmetrical uncertainty (SU). However, irrelevant and redundant features are removed by using mutual information (MI). However, this method provides confidence in the selected subset of features that will yield realistic results with less time and effort. FS-PPC is able to retain classification performance and meaningful features while consisting of non-redundant features. The proposed methods have been practically applied to analysis of real clinical data and their effectiveness has been assessed. The results show that the proposed methods are enable to minimise the clinical data problems whilst, at the same time, maximising classification performance of data.

ACKNOWLEDGEMENT

The completion of this thesis would not have been possible without the guidance and help of a number of individuals who in one way or another contributed and extended their valuable assistance in the preparation and completion of this study.

First and foremost I would like to acknowledge the tireless and prompt help of my supervisor, Dr Chandrasekha Kambhampati. Chandra has always allowed me complete freedom to define and explore my own directions in research. While this proved difficult and somewhat bewildering to begin with, I have come to appreciate the wisdom of his way. I would like to thank Dr Leonardo Bottaci and Dr Darryl Davis for their technical assistance and helpful comments. I would like to thank the Department of Computer Science and all of the staff for providing me with this opportunity and support. I am grateful to the Thai Government, which has provided me with much appreciated financial support during my study period, to make this long-held dream come true.

I thank Mrs Pamela Forster, Dr Septavera Sharvia and Winai Nadee for providing valuable feedback and reading parts of this thesis. My thanks go to Dongyue Tan for helping me when I struggled with mathematics.

Special thanks to my beloved father and mother, my kind sister and brother and not forgetting all my friends for their unconditional love, trust, sacrifice, encouragement, and prayers with during these years.

DECLARATION

Parts of the work reported in this thesis were published as research papers in the following sources:

1. Kambhampati, C., Sarangdhar, M. & Poolsawad, N. 2010. Dysphonia measures in Parkinson's disease and their use in prediction of its progression. In: International Conference on Knowledge Engineering and Ontology Development (KEOD), 2010 Spain. 104.

2. Poolsawad, N., Kambhampati, C. & Cleland, J. G. F. Feature Selection Approaches with Missing Values Handling for Data Mining - A Case Study of Heart Failure. International Conference on Data Mining (ICDM 2011), 2011 Phuket, Thailand. World Academy of Science, Engineering and Technology, 828-836.

3. Poolsawad, N., Moore, L., Kambhampati, C. & Cleland, J. G. F. Handling Missing Values in Data Mining - A Case Study of Heart Failure Dataset. The 2012 8th International Conference on Natural Computation (ICNC'12) and the 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'12), 2012 Chongqing, China. the Institute of Electrical and Electronics Engineers, 2946-2950.

4. Poolsawad, N., Moore, L., Kambhampati, C. & Cleland, J. G. F. 2012. Performance Metrics for Classification in Clinical Dataset. the 19th International Conference on Neural Information Processing (ICONIP2012). Doha, Qatar.

5. Poolsawad, N. & Kambhampati, C. 2014. Issues in the mining of heart failure datasets. International Journal of Automation and Computing, vol. 11, 2, pp. 162-179.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
ANNIGMA	Artificial Neural Net Input Gain Measurement Approximation
AUC	Area Under the ROC Curve
BMI	Body Mass Index
CBFS	Clearness-Based Feature Selection
CFS	Correlation Based Feature Selection
CMCI	Concept Most Common Value Imputation
CRISP	Cross-Industry Standard Process
CRISP-DM	Cross-Industry Standard Process for Data Mining
DFL	Discrete Function Learning
DM	Data Mining
DT	Decision Tree
EM	Expectation-Maximization
EMI	Expectation-Maximization Imputation
FCBF	Fast Correlation Based Filter
FKMI	Fuzzy *K*-Means Clustering Imputation
FN	False Negative
FP	False Positive

FSFS	Feature selection using feature similarity
FS-PPC/PPC	Feature Selection by Projecting onto Principal Component
HCDF	Handling Clinical Data Framework
IBM	International Business Machines Corporation
IG	Information Gain
KMI	*K*-Means Clustering Imputation
*k*NN	*k*-Nearest Neighbour
KNNI	*K*-Nearest Neighbour Imputation
MAE	Mean Absolute Error
MAR	Missing at Random
MCAR	Missing Completely at Random
MCI	Most Common Value Imputation
MI	Mutual Information
MIFS	Mutual Information for Feature Selection
ML	Maximum-Likelihood
MLP	Multilayer Perceptron
MNAR	Missing Not at Random
MV	Missing Values
NHS	National Health Service
NLPCA	Nonlinear Principal Component Analysis
NPV	Negative Predictive Value
NYHA	New York Heart Association

PA	Predictive Accuracy
PC	Principal Component
PCA	Principal Component Analysis
PDF	Probability Density Function
PFA	Principal Feature Analysis
PPV	Positive Predictive Value
RBFN	Radial Basis Function Network
RED	Redundancy Rate
RF	Random Forest
RFS-MRMR	Recursive Feature Selection Based on Minimum Redundancy Maximum Relevancy
RLFS	Unsupervised Feature Selection for Relation Extraction
RMSE	Mean-Squared Error
ROC	Receiver Operating Characteristics
SAS	Statistical Analysis System Institute Inc.
SEMMA	Sample, Explore, Modify, Model, Assess
SFFS	Sequential Floating Forward Selection
SHFM	The Seattle Heart Failure Model
SMO	Sequential Minimal Optimization
SMOTE	Synthetic Minority Over-sampling Technique
SU	Symmetrical Uncertainty
SVM	Support Vector Machine

SVMI	Support Vector Machine Imputation
TN	True Negative
TP	True Positive
UFSN	Unsupervised Feature Selection Scheme for Nominal Data
UFSS	Unsupervised Feature Subset Selection

NOTATIONS
(Most common symbols)

X	=	the datasets, investigated data matrix
	=	$\{x_{i,j}\}, i = 1, 2, 3, \ldots, n\,;\; j = 1, 2, 3, \ldots, m$
	=	$(X_1, X_2, X_3\,, \ldots, X_m)^T$
	=	$X_i \in X \subseteq \mathbb{R}^n\,; i = 1, \ldots n$
$x_{i,j}$	=	each data object, each data element
i	=	row or record of data
j	=	column or attribute of data
n	=	number of dataset records
m	=	number of dataset attribute
Y	=	outcome of fully observed data
p	=	number of dimensions
X_F	=	$X_F(F_0, F_1, \ldots, F_{m-1})$ set of data with m features
F	=	a representation of feature $F_j = \{x_j\}, j = 1, 2, \ldots m$
Φ	=	a representation of it with lower dimensions, $\Phi = (\Phi_1, \Phi_2, \Phi_3, \ldots, \Phi_p)^T$ with $p \leq m$
S	=	subset of features
δ	=	threshold or criteria

CHAPTER 1

INTRODUCTION

Data mining is an evolving area in information technology because of the ready availability of large quantities of data. According to IBM every day 2.5 quintillion bytes of data are generated, so much so that almost 90% of data present today has been generated in the last couple of years (IBM, 2011; Eaton *et al.*, 2012). Datasets have been created that cover all areas of human endeavour, including business, medical and clinical, geographical, and image data. Data mining is used to extract hidden information from large databases (Han *et al.*, 2012). It aims to automatically extract knowledge in an explicit form from large scale data. This means that the information and knowledge mined must be meaningful enough to provide some tangible reasons for this computational effort.

This thesis investigates the data mining problems of clinical datasets and extracts meaningful information from them in order to develop decision support systems (Ming-Syan *et al.*, 1996). As with all large datasets, clinical data provides its own challenges and complexities. Given the current climate where demands are made for reducing costs, increasing efficiency and improving care, data mining is providing valuable insights into many of these issues (Bardhan and Thouin, 2013; Groves *et al.*, 2013; Gupta and Sharda, 2013). It is also helping to develop new diagnostic, decision support system and medical treatment tools. Clinical datasets that are used in this thesis have a number of variables, which are often used to diagnose a particular disease among a number of patients. Often these variables are directly related to the ailment or are associated with it. For example, heart failure datasets often contain variables related to the heart as well as renal failure. At the same time, the dataset is composed of several types of data, e.g. numerical, continuous and categorical data. This data could consist of historical data of the patients or simply snapshots at any given time of a set of patients. As a result, these datasets present a set of challenges such as (a) various systematic and human errors, (b) a great deal of missing data, (c) non-normally distributed data, (d) imbalanced classes and (e) large numbers of variables (Tanwani and Farooq, 2009; Poolsawad *et al.*, 2011; Poolsawad *et al.*, 2012a; Poolsawad *et al.*, 2012b).

Data mining is often carried out within a framework, which consists of a number of different steps (Azevedo and Santos, 2008; Olson and Delen, 2008; Wright and Sittig, 2008)). A focus of this thesis is the problem of clinical datasets (section 1.1). Given the nature of the dataset, the thesis is concerned with improving the performance of classification; and also this thesis focuses on feature selection techniques for selecting the significant variables.

Methods for reduction of dimensions can be categorised as (a) feature extraction techniques and (b) feature selection techniques. Feature extraction combines all the original features and generates a set of new novel and synthetic features. Principal Component Analysis (PCA) (Tabachnick and Fidell, 1996) is the most commonly used feature extraction technique. It generates a feature set with lower dimensionality and the features do not carry the original labels present in the dataset. An extension of PCA is Nonlinear Principal Component Analysis (NLPCA) (Zabiri *et al.*, 2009). Depending on the requirements for the final solution, an alternative method for reducing the dimensionality is feature selection. Here an optimal subset of original features is selected based on some criterion. It not only reduces the number of features, but also removes irrelevant and redundant features. Feature selection is often used in the development of decision support systems (Polat and Güneş, 2007; Sivasankar and Rajesh, 2012). For clinical applications it is important that the labels of the variables are retained, and thus feature selection is often the strategy employed to reduce the dimensionality of the problem (Hardin and Chhieng, 2007; Bonney, 2011). Feature selection algorithms can be divided into two categories: (a) the filter model (Yu and Liu, 2004) which relies on the general characteristics of the data to evaluate and select the subset of features without involving any mining algorithm and (b) the wrapper model (Kohavi and John, 1997) which requires a data mining algorithm to search for features, as it aims to improve the performance of the subset of features but is more computationally expensive than the filter model.

1.1 Clinical dataset

This research was carried out on a real heart failure dataset called "LIFELAB". They have a large repository of data, historical, and geographical covering generations of the same family. In the thesis, a snapshot at a particular point of LIFELAB is used. It is composed of 463 continuous, categorical variables, and

2,032 patients. This dataset has many entities; these entities and the relationships between them are shown in Fig. 1.1. The examples of the significant entities are *_MAIN* entity contains the general information for each patient, e.g. Link ID, sex, age, height and weight. These entities are linked to each other in a same ways by using Link ID from _ *MAIN* to bring information from each entity. Then *_DEATH* entity is the follow up data that provides mortality 'Dead' and 'Alive' data for each patient, and this data is also used to be a target variable for this research. *_BLOOD*, *_ECG*, *_ECHO*, *_EXAM* and *_PFT* are used for a group of input variables and the list of variables that contained in these entities has shown in Table 1.1. Rest of the entity is examined for the medical purpose that has not used in this current research.

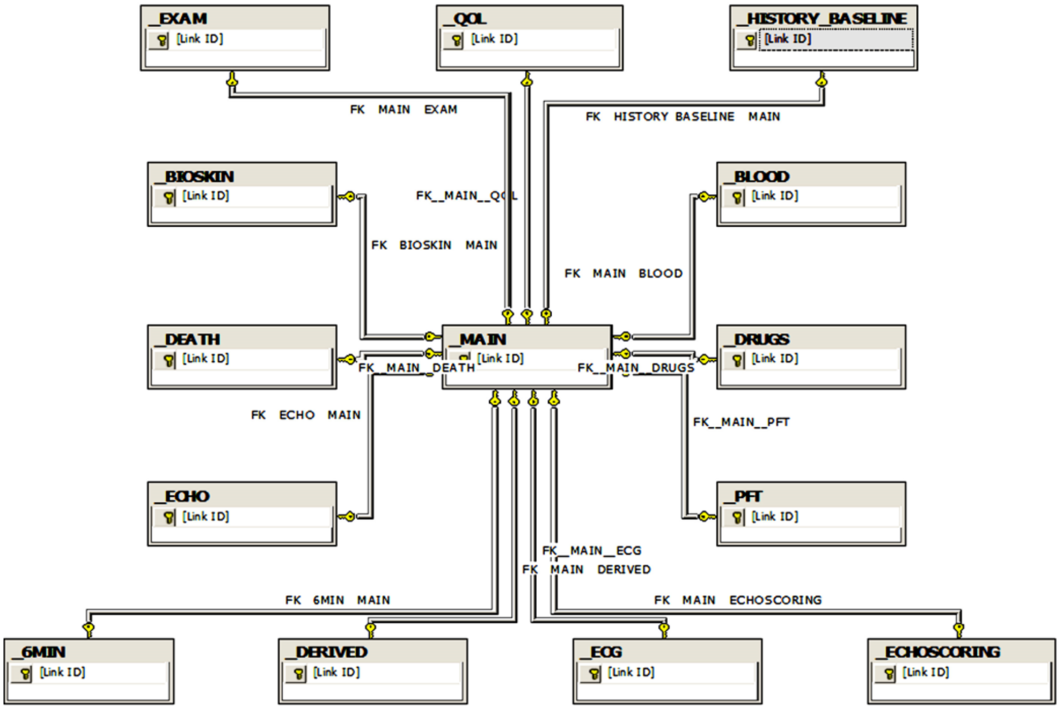


Figure 1.1: LIFELAB's data relationship

The LIFELAB dataset presents information regarding the incidence, prevalence and persistence of Heart Failure. Within the dataset, variables with missing values greater than 25% are excluded to minimise problems during the data mining process. In the presence of moderate missing values (less than 25%), variables correlated with the outcome of interest are more impactful than those correlated with missingness (Collins *et al.*, 2001). As a result, the numbers of variables and patients were substantially reduced to 60 variables (Table 1.1) and 1,944 patients. This indicates the challenges and complexities in clinical datasets, which are discussed in the following sections.

Table 1.1 (1): The analysis set

Variable	Units
Demographics (Main)	
Age	Years
Laboratory (Blood)	
Sodium	mmol/L
Potassium	mmol/L
Chloride	mmol/L
Bicarbonate	mmol/L
Urea	mmol/L
Creatinine	mmol/L
Calcium	mmol/L
Adj Calcium	mmol/L
Phosphate	mmol/L
Bilirubin	mmol/L
Alkaline Phophatase	IU/L
ALT	IU/L
Total Protein	g/L
Albumin	g/L
Uric Acid	mmol/L
Glucose	mmol/L
Cholesterol	mmol/L
Triglycerides	mmol/L
Haemoglobin	g/dL
White Cell Count	10^9/L
Platelets	10^9/L
MCV	fL
Hct	fraction
Iron	umol/L
Vitamin B12	ng/L
Ferritin	ug/L
CRP	mg/L
TSH	mU/L
MR-proANP	
MR-proADM	
CT-proET1	
CT-proAVP	
PCT	
ECG	
Rate	bpm
QRS Width	msec
QT	

Table 1. 1 (2): The analysis set

Variable	Units
ECHO	
LVEDD	cm
LVEDD	Hgt indexed
BSA	m^2
Aortic Root	cm
Left Atrium	cm
Left Atrium	BSA indexed
Left Atrium	Hgt indexed
Aortic Velocity	m/s
E	
Examination	
Height	m
Weight	kg
Body mass index (BMI)	kg/m^2
Pulse	bpm
Systolic BP	mmHg
Diastolic BP	mmHg
Pulse BP	mmHg
Pulmonary function test (PFT)	
FEV1	L
FEV1 Predicted	L
FEV1 % Predicted	%
FVC	L
FVC Predicted	L
FVC % Predicted	%
PEFR	L

1.1.1 Incomplete, errors and noisy data

There is a wealth of clinical and health records generated every day and kept in storage. This raw clinical data is usually incomplete, containing missing values due to different systematic ways of collecting the data by healthcare practitioners. Clinical datasets are usually accompanied by missing values and misclassified values. Methods of data imputation (Acuna and Rodriguez, 2004; Lin and Haug, 2006) or missing value replacement are employed to cope with these issues. Inconsistent data can also exist: for example, the variables may have specific values but another might enter as free text; the data may also contain error and noise. Outliers may be due to many reasons, such as entry errors, and this was inspected to remove irrelevant variables (Lin and Haug, 2006; Olson and Delen, 2008).

1.1.2 Diverse clinical features and their scales

There are approximately 400 features in the dataset, comprising of many scales of measurement. Some variables consist of integer and decimal values and some scales have a wide range while some have a small range. Normalisation (Han *et al.*, 2012) will be applied to solve these problems so that the data elements are within the same scale and manageable for sequential data mining processes.

1.1.3 Class imbalance

Medical data commonly has an imbalanced class distribution. Positive samples are special or rare cases that occur infrequently while negative samples are abundant. Then there are the causes of imbalanced classes. On the other hand, imbalanced classes mean that one class is represented by a large number of samples while the others are represented by small numbers.

1.1.4 Large dimensionality

From the issue of diverse features, we proceed to reducing the dimensionality of the dataset. Large dimensionality refers to the problem of the data containing too many features. Feature selection efficiently copes with this issue; the technique selects meaningful features that can be used in predictive modelling.

1.2 Motivation and research problem

Healthcare systems typically generate huge amounts of data, which includes numbers, text, charts and images. Unfortunately, all of these data are rarely used to develop decision support systems. There is a wealth of hidden information in this data that is largely untapped (Palaniappan and Awang, 2008). Consider the case of deaths due to heart failure; a simple statistical analysis indicates that most human deaths are due to heart failure (Rees, 1997; NHS, 2010). At the same time, there is likely to be a huge underestimate of the actual number of deaths caused by heart failure. However, there are no models present to predict the progression of heart failure, the fast and efficient diagnosis of heart failure and the relationship with various medical titrations available. These models have become even more crucial with the recent demographic changes and the increased need for planned care. The Department of Cardiology of the Hull York Medical School (HYMS) is at the forefront of research in both Heart failure and the use of Tele-Health techniques. (Cleland *et al.*, 1999; Cleland *et al.*, 2009; Paredes *et al.*, 2009).

Modern medicine is faced with the challenge of acquiring, analysing and applying the large amount of knowledge necessary to solve complex clinical problems (Ramesh *et al.*, 2004). A major challenge facing healthcare organisations (hospitals, medical centres, etc.) is the provision of quality services at affordable costs. Quality service implies the timely diagnosis of patients and the timely administration of treatments that are effective. Poor clinical decisions can lead to disastrous consequences which are unacceptable; thus, accuracy of results is given a high premium (Gupta *et al.*, 2005). Currently, healthcare systems are often developed based on experts' knowledge, intuition and experience rather than on the knowledge-rich data hidden in the database. A common feature of diagnostic/prognostic predictive models is the need for medical experts or someone who has specific knowledge of a subject, for example, cardiologist and clinician. It is often the case that this expert will decide on what are the significant variables that will be used to develop the models, e.g. Seattle Heart Failure Model (SHFM) (Levy *et al.*, 2006; Ketchum *et al.*, 2010). However, a problem arises because not all of their expert knowledge is documented and is often focused within a narrow range of data that is available. Because of this, the results can have unwanted biases, errors and excessive medical costs, which may affect the quality of care and service.

Implicit in the above is the importance of selecting significant or correct variables for developing accurate and correct predictive models. As a result, it is necessary to investigate further and develop an efficient predictive model, which would become a basis for developing proper and appropriate tools. Integrating the computational and medical knowledge for optimal care improves survival and quality of life for the many patients that suffer from diseases. One of the data mining techniques that we will investigate particularly and attempt to develop is feature selection. Herein, the properties of various feature selection schemes are considered with regard to heart failure clinical datasets. We strive to transform the dataset into an appropriate form so that data mining algorithms can be used successfully to develop models for designing treatments to be use on patients with heart failure. Thus in turn would enable provision of optimal care that would increase the survival rate, and also improve quality of life for the many patients who suffer from heart failure.

1.3 Research aim and objectives

From the above it is apparent that there are numbers of challenges in clinical datasets. The main goal of this thesis is to address data complexities and improve the performance of classification. The former focuses on understanding the relationships between the properties of data as well as data mining problems. The development of the data mining framework and feature selection methodology will take into account the issues associated with missing values, and imbalances in classes.

Thus, the objectives of the research are as follows:

- To develop a data mining framework for classification based on the underlying statistical properties of the datasets and the existing frameworks
- To investigate the relationship between the methods for imputation and the statistical properties of the datasets
- To discover the effect of class imbalance on performance of classification and propose the sampling data method for balancing data
- To investigate feature selection techniques in clinical datasets
- To develop a new method for selecting the significant variables by integrating two techniques of dimensionality reduction, namely, feature extraction and feature selection.

1.4 Thesis structure

This thesis will present the data mining techniques to address the research objectives stated above. These will be dealt with in the next seven chapters of this thesis. In Chapter 2, the framework for mining data is discussed and outlined. This framework consists of six stages, namely 1) Data analysis, 2) Imputation, 3) Data sampling, 4) Dimensionality reduction, 5) Classification and 6) Evaluation. After a discussion of the framework, in Chapter 3 the issue of imputing missing values is taken up. In this chapter, the relationships between the fundamental statistical properties of the data and the imputing methodologies are discussed. In Chapter 4, the principles for reduction of dimensions are discussed. This chapter covers both feature extraction and feature selection techniques. Missing values and the dimensions of the problems play a role in developing predictive models. The model developed is for classifying patients in terms of life expectancy. Imbalanced class is the one of the crucial issue, which is discussed in Chapter 5. In Chapter 6, a new methodology for selection of features is developed. The method is based on the use of Mutual Information and Symmetrical Uncertainty. In Chapter 7, the results of data mining using various methods are shown and discussed, in terms of missing values imputation, class balancing and feature selection affect classification. Chapter 8 concludes the thesis with a summary of the main contributions of the thesis and gives some suggestions for future work.

CHAPTER 2

FRAMEWORKS FOR DATA MINING

2.1 Introduction

Data mining in the main is a process whereby meaning, information and knowledge can be extracted from a dataset (Witten and Frank, 2005; Han *et al.*, 2012). A naïve approach would be to collect the data, and run clustering, classification, model identification or estimation algorithms on the data. However, such an approach is not likely to give good results since the data could have a large number of variables, both irrelevant and redundant, there could be pieces of data missing, and the outcomes in the dataset may not be balanced. These are some of the challenges faced by miners of data. Thus often, data mining algorithms are preceded by pre-processing and there may be some post processing to solve the data issues. Given the options available, there is a requirement for developing a framework for data mining (Azevedo and Santos, 2008; Olson and Delen, 2008; Kamath, 2009).

There are a number of different frameworks available, e.g. CRISP (Chapman *et al.*, 2000; Wirth and Hipp, 2000) and SEMMA (SAS Institute Inc.). These frameworks are similar, with small variations depending on the nature and type of data available for mining. In this chapter, a generic framework based on these frameworks with modifications for clinical datasets is outlined.

In Chapter 1, the challenges posed by clinical datasets were discussed. In this chapter, the data mining framework is proposed to deal with these issues. The proposed framework has a simple algorithmic framework and makes use of modification for different data issues. The main goal of this framework is to minimise the problems raised by missing values, imbalanced class and high dimensionality of data. The research in this thesis aims to enable more accurate selection of significant variables from clinical datasets. The chapter is organised along the following lines. In section 2.2, the concept of data mining is discussed. This section presents a general classification of data mining tasks, which can be classified into two categories: descriptive and predictive data mining approaches. Along with this, this section also outlines the approaches of supervised and unsupervised learning. In section 2.3, data mining frameworks are discussed, focusing on CRISP and SEMMA, which are the basic frameworks for development. Later on in section 2.3.3, the six procedures in the proposed data mining framework−Handling Clinical Data Framework (HCDF)−are outlined: (1) data analysis (2) imputation, (3) sampling data, (4) dimensionality reduction, (5) classification and (6) evaluation. Specific examples using this framework are outlined in the later chapters of this thesis. It should be noted that whereas a data mining framework outlines an approach for solving the problem, a data mining technique applies the framework. This framework is used for handling clinical data issues to provide a better understanding of their characteristics, and producing better performance of classification on significant variables.

2.2 Data mining

Data mining is one of the processes for discovering new knowledge and information. It can be defined as finding hidden information from a given set of data through the use of statistical and mathematical algorithms to extract meaningful information, trends and patterns (Han *et al.*, 2012). Thus, data mining can be viewed as the analysis of observational datasets to find unexpected relationships and to summarise data in a meaningful manner that is both understandable and useful. It is an interdisciplinary area bringing together methods from machine learning, pattern recognition, statistics, datasets and visualization to address the issue of information extraction. One of the two primary goals of data mining is to perform prediction through the use of predictive models. The second goal of data mining is the construction of a descriptive model, which is able to identify patterns or relationships in data while exploring the underlying properties of data.

2.2.1 Predictive modelling

Predictive modelling (Witten and Frank, 2005; Hardin and Chhieng, 2007; Han *et al.*, 2012) falls into the category of supervised learning. Thus, one variable is clearly labelled as the target variable and is a function of the other variables. Most models are typically built to predict the behaviour of new cases and to extend the knowledge to objects that are new or not yet understood. The following briefly explains several types of algorithms that are useful for predictive modelling:

1) Prediction: given a data item and a predictive model, it predicts the value for a specific attribute of the data item.
2) Regression: determines the best function that is suitable for the data. Here data is mapped to show real value prediction of the target variable. On the other words, regression is used when the target variable is a continuous variable.
3) Classification: given a set of predefined categorical classes (or discrete variable), it determines to which of these groups or classes a specific data item belongs.
4) Evolution and deviation analysis: focuses on discovering the most significant changes in the data from previously measured values.

2.2.2 Descriptive modelling

For a preliminary exploration of data, a general description is required. In order to obtain this, algorithms for density estimation, smoothing, data segmentation and clustering are run on the data. These can be classified as unsupervised algorithms, since they do not have a specific target. Clustering is a well-studied technique in statistics often used for initial exploration. There is an underlying assumption that the dataset contains natural clusters which, when discovered, can be characterized and labelled. While for some cases it might be difficult to decide to which group they belong, it is often assumed that the resulting groups are clear-cut and carry an intrinsic meaning. In contrast, in segmentation analysis, the user typically sets the number of groups in advance and tries to partition all cases into homogeneous subgroups. The following list describes methods of descriptive modelling (Witten and Frank, 2005; Hardin and Chhieng, 2007; Han *et al.*, 2012):

1) Clustering: similar to classification but the class(es) of data is not predefined. Clustering is best used for finding groups of items that are similar.
2) Link Analysis (Associations rule or Affinity analysis): given a set of data items, it identifies relationships between attributes and items, such as the presence of one pattern implies the presence of another pattern. The investigation of relationships between items over a period of time is also often referred to as 'sequential pattern analysis'.
3) Summarization: involves methods for finding a compact description for a subset of data. Summarization techniques are often applied to interactive exploratory data analysis and automated report generation.

2.2.3 Supervised and unsupervised learning

Machine learning has become essential in the mining of data as knowledge discovery and adaptive information extraction has to play an important role in modern life. Machine learning can be broadly classified as supervised or unsupervised learning (Hardin and Chhieng, 2007). *Supervised* learning creates models by using input attributes to predict the output attribute values. Output attributes are also known as dependent variables as their outcome depends on the values of one or more input attributes. Input attributes are referred to as independent variables. When learning is *unsupervised*, an output attribute does not define for predicting. Therefore, all attributes used for model building are independent variables.

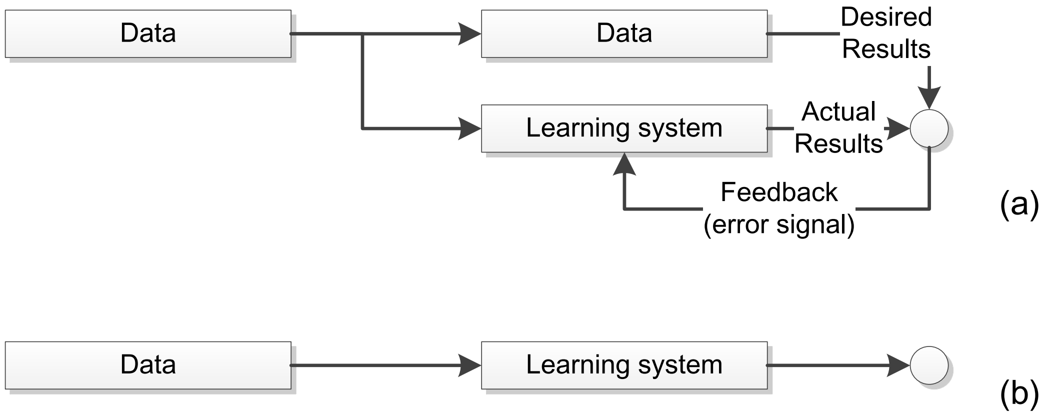


Figure 2.1: Supervised and unsupervised learning

(a) Supervised learning.

(b) Unsupervised learning

(Hardin and Chhieng, 2007)

Supervised learning (Fig. 2.1(a)) strategies can be further categorised according to whether output attributes are discrete or categorical, as well as by whether models are designed to determine a current condition or predict future outcome. Methods used for mining datasets are, mainly, supervised methods; thus, there is a particular pre-specified target variable and data is readily available. Within the method, there are many algorithms where the value of the target variable is provided so that the algorithm may learn which values of the target variable are associated with which values of the predictor variables.

On the other hand, unsupervised learning (Figure 2.1(b)) searches the data for interesting associations and does not have a specific target assigned to it. It attempts to group elements into a number of classes to cover all the items in the dataset. Thus unsupervised algorithms search for patterns and structure among all the variables. In unsupervised learning situations, all variables are treated in the same way. There is no distinction between explanatory and dependent variables. Intuitively, it can be seen that clustering would be the most popular and common method which is used.

The dividing line between supervised learning and unsupervised learning is similar to the one that distinguishes discriminant analysis from cluster analysis. Supervised learning requires that the target variable is well defined and that a sufficient number of its values are given. For unsupervised learning, typically, the target variable is either unknown or has only been recorded for a small number of cases (Gentle and Hardle, 2004).

2.3 Data mining frameworks

The data mining framework consists of a method for representing the data and knowledge, and a method for data manipulation (Anand *et al.*, 1996).The framework provides a solution to deal with the complexities of data by using the potential of data mining algorithms. The realisations of processes in the framework are a unified approach for solving the different tasks of data mining. Approaches used for data mining have been problem specific, for example, classification problems. In this thesis, SEMMA and CRISP-DM have been chosen, because they are considered to be the most popular and presented in many of the publications in the area and are widely used in practice (Azevedo and Santos, 2008).

2.3.1 CRISP-DM

There is a Cross-Industry Standard Process for Data Mining (CRISP-DM) (Chapman et al., 2000) widely used by industry members. This model consists of six phases intended as a cyclical process (see Fig. 2.2).

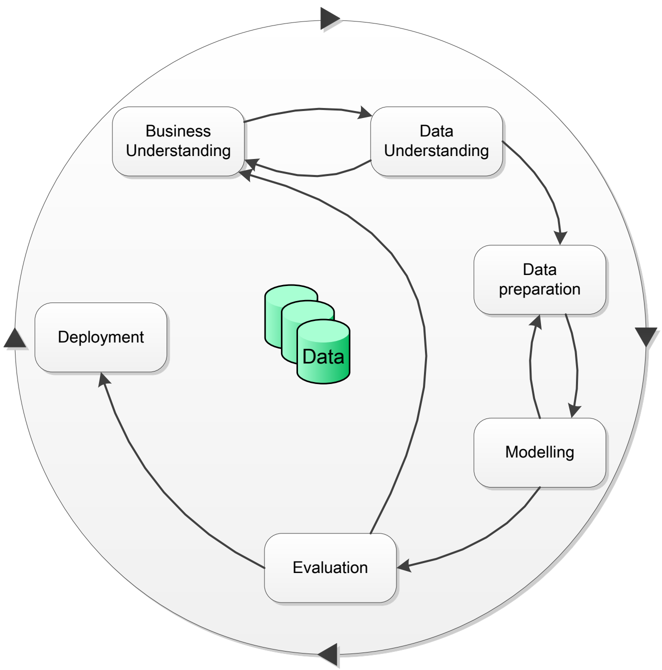


Figure 2.2: CRISP-DM Process Model (Chapman *et al.*, 2000; Wirth and Hipp, 2000)

1) **Business Understanding:** Business understanding includes determining business objectives, assessing the current situation, establishing data mining goals, and developing a project plan.

2) **Data Understanding:** the step considers data requirements including initial data collection, data description, data exploration, and the verification of data quality.

3) **Data Preparation:** Once the data are available, data cleaning and data transformation need to occur in this phase.

4) **Modelling:** Data mining models can be applied, for example, visualization and cluster analysis are useful for analysis and generalized rules can develop association rules.

5) **Evaluation:** Model results should be evaluated in the context of the business objectives established in the business understanding phase. This phase will show the results using visualization, statistical, and artificial intelligence tools that show the user new relationships that provide a deeper understanding of organizational operations.

6) **Deployment:** Models can be obtained that may then be applied to business operations for many purposes, including prediction or identification of key situations. These models need to be monitored for changes in operating conditions. If significant changes do occur, the model should be redone.

CRISP-DM is extremely complete and well-documented. All its stages are duly organized, structured and defined, enabling a project to be easily understood or revised (Azevedo and Santos, 2008). This six-phase process is not rigid, in terms of the order of procedure. Additionally, experienced analysts may not need to apply each phase for every study.

2.3.2 SEMMA

SEMMA (SAS Institute Inc.) is the methodology that SAS proposed for developing data mining products. The acronym SEMMA stands for ***S**ample*, ***E**xplore*, ***M**odify*, ***M**odel*, ***A**ssess*. Beginning with a statistically representative sample of the data, SEMMA is intended to make it easy to apply exploratory statistical and visualisation techniques, select and transform the most significant predictive variables, model the variables to predict outcomes, and finally confirm a model's accuracy (Olson and Delen, 2008). A pictorial representation of SEMMA is given in Fig. 2.3.

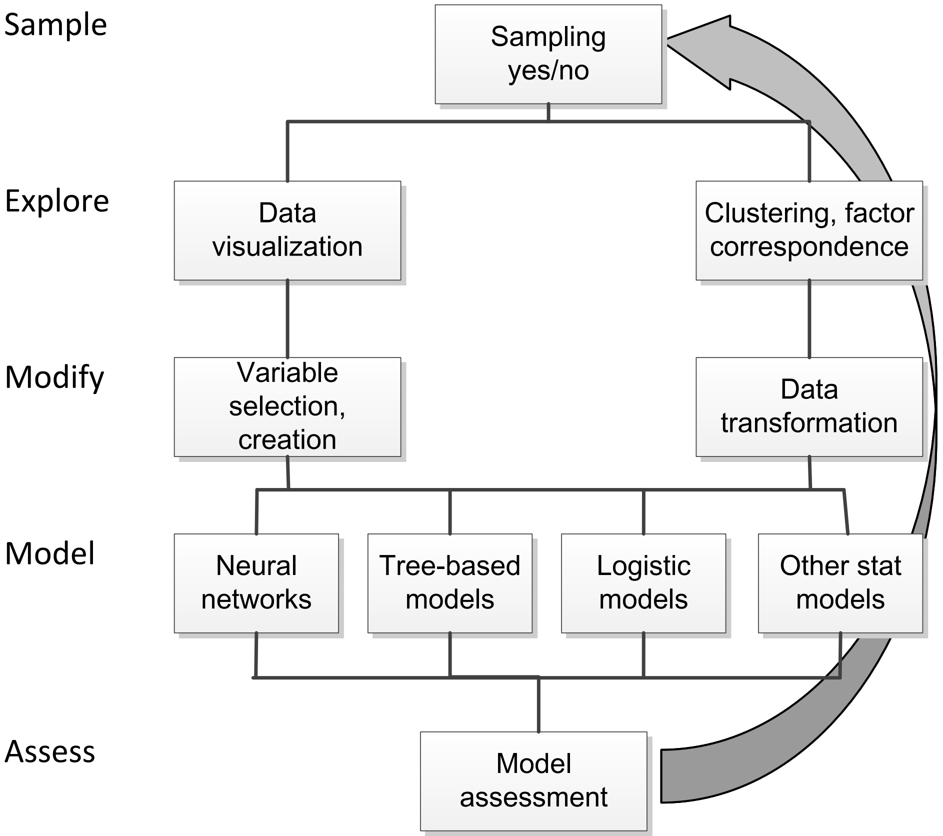


Figure 2.3: Steps in the SEMMA Methodology (SAS Institute Inc.)

1) **Sample:** For optimal cost and computational performance, a sampling strategy applies a reliable, statistically representative sample of the full detailed data. In the case of very large datasets, mining a representative sample instead of the whole volume may drastically reduce the processing time required to get crucial business information. It is also advised to create partitioned data sets for better accuracy assessment.

2) **Explore:** Exploration helps refine and redirect the discovery process. If visual exploration does not reveal clear trends, one can explore the data through statistical techniques including factor analysis, correspondence analysis, and clustering.

3) **Modify:** This is where the user creates, selects, and transforms the variables upon which to focus the model construction process. It may also be necessary to look for outliers and reduce the number of variables, to narrow them down to the most significant ones.

4) **Model:** Modelling techniques in data mining include artificial neural networks, decision trees, rough set analysis, support vector machines, logistic models, and other statistical models – such as time series analysis, memory-based reasoning, and principal component analysis. Each type of model has particular strengths, and is appropriate within specific data mining situations depending on the data.

5) **Assess:** This is where the user evaluates the usefulness and the reliability of findings from the data mining process. In this final step of the data mining process, the user assesses the model to estimate how well it performs.

Although SEMMA is a methodology, it is based on the technical part of the project only, i.e. its aim is to solve the data mining part and it does not take into account all the management side. Like the above approach, SEMMA also sets out a waterfall life cycle, as the project is developed through to the end. If the solution is not interesting, developers go backwards through the stages (Azevedo and Santos, 2008).

2.3.3 Handling clinical data framework (HCDF)

We are developing an already existing frameworks (SAS Institute Inc.; Chapman *et al.*, 2000; Wirth and Hipp, 2000; Wright and Sittig, 2008; Poolsawad *et al.*, 2011), being motivated by clinical dataset challenges to minimise these problems whilst, at the same time, maximising classification performance of data. Thus, the framework consists of the processes of (1) data analysis (2) imputation (3) data sampling (4) dimensionality reduction (5) classification and (6) evaluation. These are flexible procedures for dealing with a variety of data issues. This framework is outlined in Fig. 2.4 below:

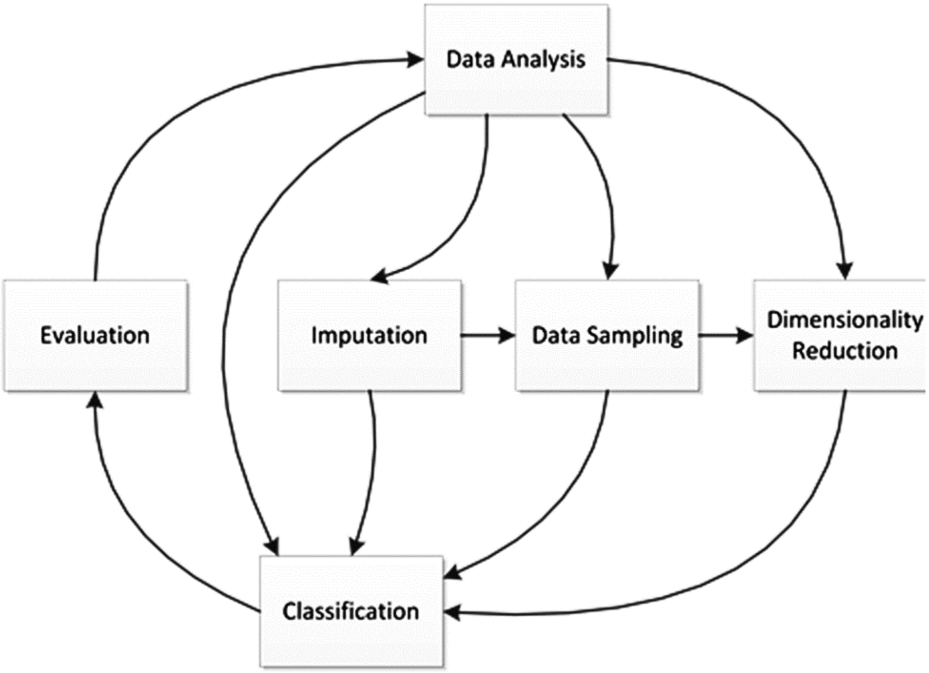


Figure 2.4: Handling clinical data framework (HCDF)

2.3.3.1 Data analysis

Data analysis is the process of understanding the requirements according to the business objectives and data mining goals. The first stage of the data mining process is to select the related data from available sources to correctly describe a given business task. There are at least three issues to be considered in the data selection. The first issue is to set up a concise and clear description of the problem. For example, with a LIFELAB dataset, the data mining project may seek to identify the mortality in patients suffering from heart failure. Another example may be to identify the significant variables associated with patient mortality. The second issue would be to identify the relevant data for the problem description. Most demographic, laboratory and examination data could be relevant to both examples. The third issue is that selected variables for the relevant data should be independent of each other. Variable independence means that the variables do not contain overlapping information. The variable selection issue will be discussed in details in discussing the dimensionality reduction process.

Data variables for mining can vary; the data type can be categorised as quantitative or qualitative data. Quantitative data is measurable using numerical values. It can be either discrete (such as integers) or continuous (such as real numbers). Qualitative data, also known as categorical data, contains both nominal and ordinal data. Nominal data has finite non-ordered values; however, ordinal data has finite ordered values. Quantitative data can be represented by some sort of probability distribution. A probability distribution describes how the data is dispersed and shaped. For instance, normally distributed data is symmetric, and is commonly referred to as normal distribution. Qualitative data may be converted to numbers and then be described by frequency distributions.

Once data analysis is discovered according to the data mining business objectives, data pre-processing should be pursued. In this framework, the pre-processing is the main process for addressing the dataset issues. The data preparation is divided into three procedures: imputation, data sampling and dimensionality reduction. There are many statistical methods and visualisation tools that can be used to analyse the data. Common statistics, such as max, min, mean, and mode can be readily used to aggregate or smooth the data, while scatter plots and box plots are usually used to filter outliers. More advanced data mining techniques such as regression analyses, cluster analysis, or decision tree may be applied depending on the requirements for the quality of the data. The pre-processing can provide the flexibility to implement various data mining algorithms and can make a difference to the data mining results.

2.3.3.2 Imputation

The purpose of data pre-processing is to clean and prepare available data for better quality. Some available data may have different formats because they are chosen from different data sources and different data collections. However, they should be converted to a consistent electronic format. Data cleaning generally refers to filtering data and filling in missing values (imputation). This process emphasises the missing values issue because it is the one of major problem in clinical datasets. Missing values imputation smoothes data by imputing them with reasonable values. The imputed values could be calculated by various methods such as the mean, the mode and data-mining algorithms to discover knowledge patterns. However, the missing values issue and imputation will be discussed in detail in next chapter (Chapter 3). Moreover, other preparations also can be made in this process, such as transforming data and normalising data, which eliminate differences in variable scales, and furthermore filtering data, which can be examined for outliers and redundancies. Outliers are data that differ greatly from the majority of the data, or are clearly out of the range of the data. Outliers may be caused by many reasons, such as human errors or technical errors, or may naturally occur in a dataset due to unfortunate events. Arbitrarily deleting an outlier could dismiss valuable information.

Redundant data occur where the same information is recorded in several different ways for example age and date of birth of patient. The result of the data cleaning process is a dataset that can readily be used for the process of applying data mining algorithms.

2.3.3.3 Data sampling

After sampling the data, unanticipated trends and anomalies can be found in order to gain a better understanding of the dataset. Data analysis and imputation processes help refine and redirect the discovered data. If a data mining algorithm does not reveal clear important results or cannot meet the business requirements, data sampling can help to improve the prepared data, especially where data poses the problem of imbalanced class, which are common in clinical datasets. In the other words, the proportion of positive and negative cases in a dataset is not equal. Usually only a small number of people diagnosed actually have the disease. There are many more negative cases ('Alive' class in our instance) than positive cases ('Dead' class). A limitation of data mining algorithms is that they often show a strong bias toward the majority class (negative case), since the goal of learning algorithms for clinical datasets is to minimise the overall prediction error rate especially the minority class (positive case). Thus, class balancing is an important process to improve the data mining performance. In this thesis, two strategies of sampling data, (1) over-sampling and (2) under-sampling, that can solve this problem, are outlined in Chapter 5. It should be noted that the size of samples for each class should be big enough to contain the significant information and not too small to represent the data. A sampling strategy should yield a reliable, statistically representative sample of the full data. It is also advised to apply data sampling on imbalanced datasets for more accurate performance.

2.3.3.4 Dimensionality reduction

Clinical data often contains an extremely large number of variables. Using all of the variables in a data mining model is not practical in general, so dimensionality reduction (see more details in Chapter 4) plays a critical role in data mining modelling to reduce the number of variables. In this framework, feature selection is used to select the meaningful and relevant variables. The previous processes aim to prepare data for applying a data mining algorithm and gaining better accuracy in data mining modelling. Feature selection performs variable selection; however, this method may not perform well when one is evaluating datasets that contain hundreds of potential input variables. Furthermore, it should be kept in mind that the purpose of stepwise selection is not to improve the performance of accuracy but to gain an optimal subset of variables. Feature selection enables evaluation of the importance of input variables in predicting or classifying the target variable. Variable selection is often more critical for clinical or large datasets than others. Some of the data are not directly pertinent to the data mining exercise, and so may be eliminated. It may also be necessary to seek to reduce the number of variables, to narrow them down to the most significant ones.

2.3.3.5 Classification

Once data are prepared, a data mining model can be constructed that explains patterns and extracts knowledge from the data. Data mining can be achieved by association, classification, clustering, predictions, sequential patterns, and similar time sequences. Modelling techniques in data mining include artificial neural networks, decision trees, support vector machines, and other statistical models. Each type of model has particular strengths, and is appropriate within specific data mining situations, depending on the data. Data mining modelling is used to generate results for various situations. In this framework, classification is used for data mining modelling, which assumes a given set of predefined classes of samples. Classification is one of the most important data mining problem types that occur in a wide range of predictive modelling or clinical applications such as diagnostic or prognostic models. The key examination problems related to classification results are the evaluation of misclassification and prediction performance.

2.3.3.6 Evaluation

The data evaluation stage is vital to assess clinical datasets. This process consists of two issues essential for evaluating the performance of a data mining model. One is how to identify the business objective from knowledge patterns discovered in the data mining stage. Another issue is how to show the data mining results. Good evaluation leads to productive and reliable business decisions, while poor interpretation analysis may miss useful information. In this thesis, the performance accuracy and redundancy rate are used for assessing the data mining model to reveal the optimal subset of variables with good performance of accuracy. The results of the data mining study need to give feedback for improvement. The data mining study has uncovered new knowledge, which needs to be applied to the data mining project goals. It is important that the knowledge gained from a particular data mining study be monitored for change.

Table 2.1: Comparison of the data mining frameworks

CRISP	HCDF	SEMMA
Business understanding	Data analysis	-
Data understanding		Sample Explore
Data preparation	Imputation	Modify
	Data sampling	
	Dimensionality reduction	
Modelling	Classification	Model
Evaluation	Evaluation	Assessment
Deployment		

A data mining framework is used for specific application, involving interesting aspects of data handling requirements. CRISP and SEMMA were created as broad frameworks, which need to be adapted to specific circumstances. The proposed framework started with a clearly defined goal – to develop tools that would better utilise the clinical decision support system. Table 2.1 compares the Handling Clinical Data Framework (HCDF) with the CRISP and SEMMA frameworks by comparing the processes of the frameworks. HCDF includes data sampling effort, like SAMMA, while CRISP would include it in data preparation. Data is handled by imputing missing values and reducing the dimensionality, which are equivalent to data preparation in CRISP and Modify in SEMMA. Thus, HCDF is the extension framework of SEMMA that divided the ‘Modify’ step into imputation, data sampling and dimensionality reduction. It has seen that HCDF gives more flexibility and practical uses for handling different data issues.

In the case study of the heart failure clinical dataset being analysed in this thesis, the primary requirement was to build a classification model for prediction. The design of the framework was primarily driven by the data mining analytics needed. There are six processes involved in effectively using the data mining functionality to provide predictive solutions. Each of these processes is performed using the data mining techniques, using the clinical data as input to the data mining process. These processes will be discussed in Chapter 3, 4, 5 and 6.

2.4 Summary

Data mining techniques are defined by an algorithm and the most commonly used techniques include artificial neural networks, decision trees, and the nearest-neighbour method. Supervised and unsupervised learning are the techniques that distinguish the features and output of the data. The data mining model is then adjusted to minimize the error rate in the datasets. The data at hand is perfectly described, but generalisation to other data yields unsatisfactory outcomes. It is not only different data that might yield different models; different statistical methods or techniques can also affect the outcome of the model. The choice of the method is open to the user.

In our approach, it is very important to point out that our framework does not eliminate the need to cleanse the data and analyse the characteristics of raw data. In fact, to build predictive models for classification in clinical data, our data mining processes perform classification and prediction and present mining results. The mining algorithms in the data mining framework are designed to work on clinical data. The framework takes into consideration the data required for all kinds of analysis that are carried out. The results from this methodology can reveal the significant variables that will be used for a decision support system and this methodology also gives satisfactory predictive performance.

CHAPTER 3

HANDLING MISSING VALUES

3.1 Introduction

There are two main drivers for the use of data mining techniques and the design of decision support systems: (a) the availability of rich and diverse types of data; (b) the drive to improve patient care and reduce hospitalisation costs. Hence, handling missing values is investigated with the aim of data preparation and performance improvement, since most medical applications encounter missing values in their data. Values can be missing for several reasons including incorrect data entry, erroneous measurements, equipment faults or unrecorded measurements, such as some variables that would not be important for a particular medical diagnosis (Juhola and Laurikkala, 2013). In addition, there could be other reasons for the data to be missing, e.g. patient refusal to continue in the study, patient withdrawals due to treatment failure, treatment success or adverse events, or patients relocating. At the same time in many clinical data, the medical records allow for some attributes to be left blank, if they are not relevant to some classes of illness or if the patient objects to the recording of this information (Committee for Medicinal Products for Human Use, 2009). Missingness can be defined as both the existence of missing data and the mechanism that explains the reason for the data being missing. Existence of missing values causes a number of problems, the primary one being the inability to extract and discover knowledge either manually or computationally. The reason for this is that there is an inbuilt biasing of the data even before the processing starts (Barnard and Meng, 1999; Jagannathan and Petrovic, 2009). On the other hand, ignoring missing data is not an acceptable option when planning, conducting or interpreting the analysis of a confirmatory clinical data (Committee for Medicinal Products for Human Use, 2009). This is important because missing data are a potential source of bias when analysing data from clinical data. At the same time, care should be taken that the strategy employed to handle missing values should not in itself become a source of bias.

Along with the other issues discussed in Chapters 1 and 2, missing values create a unique set of problems for developing appropriate classification or prediction models, which aid in the development of decision support systems (Sittig *et al.*, 2008; Fox *et al.*, 2010). Three types of major difficulties usually associated with missing values challenge in data mining are: (a) loss of information and efficiency; (b) complication in handling and analysing the data; and (c) bias resulting from differences between missing and complete data (Barnard and Meng, 1999; Wang and Wang, 2010). In the particular case of classification, missing values and also missing values handling affect the classification performance (Acuna and Rodriguez, 2004; Juhola and Laurikkala, 2013). Therefore, missing values should be handled in such a way that the classification can tolerate even high numbers of missing values. When high numbers of missing values occur, the uncertainty of the likely treatment effect can become such that it is difficult to conclude that evidence of efficacy has been established. However, impact of missing values has seldom been investigated for the purpose of evaluating their influence on classification results. In this thesis, a multi-stage process is suggested for handling missing values, which consists of three stages, namely: (1) discard, (2) imputation and (3) evaluation, as shown in Fig. 3.1.

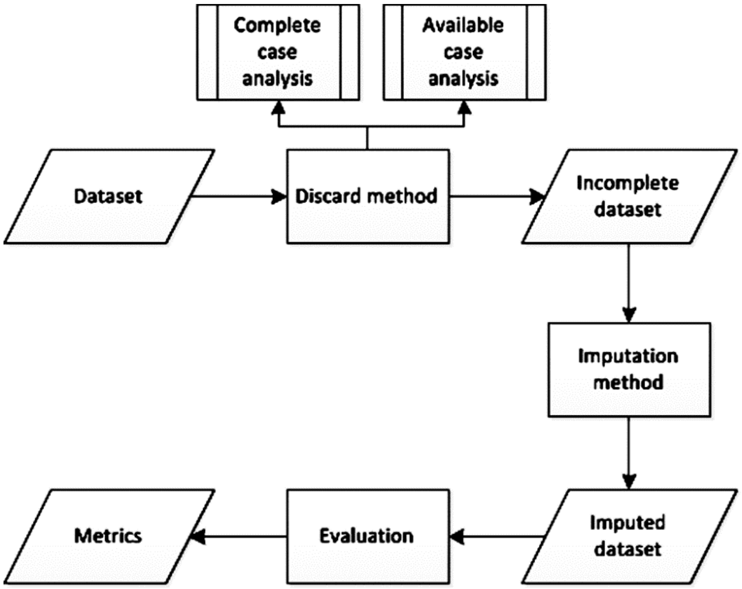


Figure 3.1: A multi-stage process for handling missing values

3.2 Types of missing values

In order to decide how to handle missing values, it is helpful to understand why they are missing. Little and Rubin (Little and Rubin, 1978) classify missing data into three categories: (1) missing completely at random (MCAR), (2) missing at random (MAR), and (3) missing not at random (MNAR).

1) **Missing completely at random (MCAR)**

If data are missing completely at random, then throwing out cases with missing data does not bias inferences. There is no relationship between whether a data point is missing and any values in the data set, missing or observed. An example of a MCAR mechanism would be that a laboratory sample is dropped, so the resulting observation is missing.

2) **Missing at random (MAR)**

Most missingness is not completely at random, as can be seen from the data itself. MAR means the propensity for a data point to be missing is not related to the missing data, but it is related to some of the observed data. For example, depressed people may be more likely to decline to report income, and thus when there is a high rate of missing data among depressed individuals, the existing mean income might be lower than it would be in observed data.

3) **Missing not at random (MNAR)**

This is missingness that depends on unobserved predictors. In addition, data is missing because there is no information recorded to confirm it so as to be able to predict the missing values. An example of MNAR is that some patients might drop out because they believe the treatment is not effective.

3.3 Discarding method

There are two main principles that are applied to discard data with missing values. These are:

3.3.1 Complete case analysis

This is a direct approach to missing data and it excludes these records. In other words, only those data or records with all attributes having value are retained for analysis. However, there two problems with this approach:

a) If the units with missing values differ systematically from the completely observed cases, a bias would be incorporated into the dataset. For example, it is possible that the majority of missing attributes could be for one class, and removing these data records could result in more complication in the analysis, including bias and the introduction of an imbalance of classes.

b) If there are a large number of variables required for the model, it is possible that the number of records available would be far less than is required.

3.3.2 Available case analysis

Available-case analysis arises when a variable or set of variables are completely excluded from the analysis because of their percentage of missing data. This method consists of determining the extent of missing data in each instance and attribute, and deletes the instances and/or attributes with high levels of missing data. In a causal inference context, this may lead to omission of a variable that is necessary to satisfy the assumptions necessary for desired interpretations.

Thus, before deleting any attribute, consideration must be given to whether that variable or attribute is necessary to the analysis. In some situations, attributes should be retained even in the presence of a high degree of missing values. Both methods, complete case analysis and discarding instances and/or attributes, should be applied only if missing data are MCAR, because missing data that are not MCAR have non-random elements that can bias the results. Table 3.1 shows the reasons for missing physical data and the different imputation decisions to deal with the problems that are raised from different causes of missing values.

3.4 Techniques for imputing

Imputation is challenging as it could introduce additional biases, since records with missing values are often systematically different from records without missing values, even when they belong to the same class (Rubin, 1987). This method is different from complete-case and available-case analysis because rather than removing variables or observations with missing data, this approach is to fill in or impute missing values. At the same time, this method retains the full sample size.

Table 3.1: Reasons for missing physical data (Gilchrist *et al.*, 2008)

Reason	Definition	Imputation Decision
No information	The default null value, no context is provided and the value cannot be interpreted further.	If MCAR/MAR, impute using entire dataset. If not MCAR/MAR, impute using data from individual patient or similar patients.
Not applicable	The data element does not apply in a given context, e.g., an answer to "gestational age" for an adult patient.	Do not impute.
Unknown	The information may be applicable, but is not known in the given context.	Impute a value based on similar patients.
Not collected/tested	The value was not collected/tested because that information was deemed unnecessary.	Impute a normal value.
Tested but unknown	The value was tested/observed, but not recorded.	Impute a value based on individual patient or similar patients.

A major methodological research interest is the analysis of data with missing values (Little and Rubin, 1978; Poolsawad *et al.*, 2012a; Zhang *et al.*, 2012). However, they can yield different kinds of bias, as detailed later in this section. These biases are difficult to eliminate since the precise reasons for missing data are usually not known. The approaches for imputation can be grouped into two broad categories: single imputation methods and multiple imputation methods.

3.4.1 Single imputation

As a method to deal with missing data, single imputation is often utilised because it is intuitively attractive. In single imputation, we fill in missing values with some type of predicted values. Kalton and Kasprzyk (1982) described many different methods for single imputation and the most important characteristics of the commonly used imputation methods (Kalton and Kasprzyk, 1982). The most common is mean imputation or median imputation, where a sample mean (median) of a variable replaces any missing data for that variable or feature or attribute (Here, variable, feature, dimension and attribute are interchangeably used). Missing values can also be replaced using values taken from matching records, also known as Hot-decking for example, *K*-Nearest Neighbour Imputation (KNNI), *K*-Means Clustering Imputation (KMI), Fuzzy *K*-Means Clustering Imputation (FKMI) and Support Vector Machine Imputation (SVMI).

Imputation performed by data predictors also has the important advantage of allowing the use of information available to the data predictors. This information may involve detailed knowledge of interviewing procedures and reasons for nonresponse that cannot be placed because of confidentiality constraints. Imputation by single imputation leads to greater consistency and thereby to reduced costs (Rubin, 1988).

3.4.2 Multiple Imputation

A more computationally demanding and intensive approach is to use a strategy called multiple imputation (Rubin, 1987). In multiple imputation each of the missing values is replaced by imputed values to create complete datasets. Once this is done, analysis is carried out under each set of imputation and analyses combined to reflect within-imputation and between-imputation variability. A number of techniques are available for multiple imputation; the most popular is Expectation-Maximization (EM) algorithm (Little and Rubin, 1978; Rubin, 1987; Schafer, 1997; Schafer and Olsen, 1998).

Multiple imputation methods have advantages and disadvantages. The major advantages of multiple imputation as indicated by Rubin (Rubin, 1987) are that standard complete-data methods are used to analyse each completed dataset. Moreover, the ability to utilise data collectors' knowledge in handling the missing values is not only retained but also actually enhanced. In addition, multiple imputations allow data collectors to reflect their uncertainty as to which values to impute. Disadvantages include the time intensiveness involved when imputing five to ten data sets, testing models for each dataset separately, and recombining the model results into one summary (Rubin, 1987).

In this thesis, a set of imputation methods are employed to handle missing values in a clinical dataset, however, the complete data sets obtained are not analysed in the same way using multiple imputation. Instead, metrics are used to compare the performance of different imputation methods. Whenever a single imputation strategy is used, the standard errors of estimates tend to be too low. The intuition here is that we have substantial uncertainty about the missing values, but by choosing a single imputation, we in essence pretend that we know the true value with certainty. Examples of imputations are discussed as follows:

3.4.3 Imputation methods

Imputation methods involve replacing missing values with estimated ones based on information available within the data set. These methods vary from simple mean imputation, to the more robust methods based on relationships between the attributes (Zhang *et al.*, 2012). In what follows is a brief description of these methods.

3.4.3.1 Most common value imputation (MCI)

This method is one of the simplest methods to implement amongst existing methods (Zhang *et al.*, 2012). Depending on the nature of the attribute, there are differences in the manner in which MCI replaces missing values.

- For nominal attributes, MCI imputes the missing value with the most common value of the attribute.
- For numerical attributes, the missing value is replaced with the average value of the attribute.
- For symbolic attributes, every missing attribute value should be replaced by the most common attribute value.

3.4.3.2 Concept most common value imputation (CMCI)

CMCI is similar to MCI, the main difference being that it takes into account only instances with the same class rather than the whole dataset (Zhang *et al.*, 2012) (or all the classes put together) (Here a concept is a subset of the set of all cases with the same outcome). Thus

- The missing value is replaced by the mode if the attribute is nominal.
- It is replaced by the mean value if it is numerical.
- For symbolic attributes, every missing attribute value should be replaced by the most common attribute value that occurs for the class.

3.4.3.3 *K*-nearest neighbour imputation (KNNI)

The previous methods use a mean across the dataset or a mean/mode within a class. These methods do not generally take into account either values of other variables or attributes within the data space. On the other hand, KNNI (Batista and Monard, 2003) is an instance-based algorithm; for every missing value that is found, KNNI uses the *k*-nearest neighbours in order to determine a value from them, which is then imputed. With this method, a proximity measure between instances has to be defined. A default or near universal measure is the Euclidean distance. (Other distances are also used depending on the nature of the attributes). Typically for nominal attributes, the most common value amongst all neighbours is taken, and for numerical values, the average value is used (Batista and Monard, 2003).

There are advantages in the use of this approach. These are (a) it can be used to predict both qualitative attributes (the most frequent value among the *k*-nearest neighbours) and quantitative attributes (the mean among the *k*-nearest neighbours); and (b) a predictive model for each attribute is not necessary. Here the dataset is used as a non-parametric lazy learning algorithm. As a result, the *k*-nearest neighbour algorithm can be easily adapted to work with any attribute classes, by modifying the attributes to be considered in the distance metric. Another advantage is that this approach can be extended to situations where there are multiple missing values within a record. The main drawback of the *k*-nearest neighbour approach is that it can become computationally expensive. Whenever the *k*-nearest neighbour looks for the most similar instances, it has to search for these in the whole data set.

3.4.3.4 *K*-means clustering imputation (KMI)

Both CMCI and KNNI use some form of a measure of similarity. The KMI (Li *et al.*, 2004) method uses the dissimilarity measure within the cluster through the addition of distances among the objects and the centroid of the cluster to which they are assigned. A cluster centroid represents the mean value of the objects in the cluster. Once the clusters have converged, the last process is to fill in all the non-reference attributes for each incomplete object based on the cluster information. Data objects that belong to the same cluster are taken to be nearest neighbours of each other, and KMI applies a nearest neighbour algorithm to replace missing values, in a similar way to KNNI (Li *et al.*, 2004).

Thus the algorithm for missing data imputation with K-means clustering method is multi-staged, and is shown in Algorithm 3.1

Algorithm 3.1: *K*-means clustering imputation method

(i) Randomly select *K* complete data objects as *K* centroids.

(ii) Iteratively modify the partition to reduce the sum of the distances for each object from the centroid of the cluster to which the object belongs. The process terminates once the summation of distances is less than a user-specified threshold ε.

(iii) Fill in all the non-reference attributes for each incomplete object based on the cluster information.

The two common measure used in this scheme are

(a) A norm distance

$$d(v_k, x_i) = (\sum_{j=1}^{m} \left| x_{i,j} - v_{k,j} \right|^p)^{1/p} \tag{3.1}$$

where $d(\cdot)$ is the distance, $v_{k,j}$ is the centroid and $x_{i,j}$ is the data object in the cluster. The Euclidean distance $p = 2$.

(b) Cosine distance which is calculated from Cosine Similarity,

$$Sim(v_k, x_i) = \frac{\sum_{j=1}^{m} x_{i,j} \; * \; v_{k,j}}{\sqrt{\sum_{j=1}^{m} x_{i,j}^2 \; \sum_{j=1}^{m} v_{k,j}^2}} \text{ , and } d(v_k, x_i) = e^{-Sim(v_k, x_i)} \tag{3.2}$$

where the notations are as before.

3.4.3.5 Fuzzy *K*-means clustering imputation (FKMI)

In FKMI (Li *et al.*, 2004), a data object cannot be assigned to a cluster represented by a cluster centroid (as is done in the basic *K*-mean clustering algorithm), because each data object belongs to all *K* clusters with different membership degrees. FKMI replaces non-reference attributes for each incomplete data object based on the information about membership degrees and the values of cluster centroids (Li *et al.*, 2004).

In fuzzy clustering, each data object x_i has a membership function, which describes the degree to which a data object belongs to particular cluster v_k. The membership function is defined in Eq. (3.3):

$$U(v_k, x_i) = \frac{Sim(v_k, x_i)^{-2/(m-1)}}{\sum_{k=1}^{K} d(v_k, x_i)^{-2/(m-1)}} \tag{3.3}$$

where $m > 1$ is the fuzzifier, which deals with a mapping from the input space to the fuzzy set, and $\sum_{k=1}^{K} U(v_k, x_i) = 1$ for any data object x_i $(1 \leq i \leq N)$. We cannot simply compute the cluster centroids by the mean values. Instead, we need to consider the membership degree of each data object. Eq. (3.4) provides the formula for cluster centroid computation:

$$v_k = \frac{\sum_{i=1}^{N} U(v_k, x_i) * x_i}{\sum_{i=1}^{N} U(v_k, x_i)} \tag{3.4}$$

Since there are unavailable data in incomplete objects, we use only reference attributes to compute the cluster centroids. The FMI approach is outlined in Algorithm 3.2

Algorithm 3.2: Fuzzy *K*-means clustering imputation method

(i) Pick *K* centroids, which are evenly distributed.

(ii) Iteratively update membership functions and centroids until the overall distance meets the user-specified distance threshold *ε*.

(iii) Impute non-reference attributes for each incomplete object, x_i based on the information and the values of cluster centroids.

$$x_{i,j} = \sum_{k=1}^{K} U(x_i, v_k) * \ v_{k,j}$$

3.4.3.6 Expectation-maximization imputation (EMI)

EMI iteratively computes the expected values for missing observations by repeatedly updating maximum-likelihood (ML) parameter estimates and imputing updated expected values until convergence is achieved (Dempster *et al.*, 1977). The expectation-maximization (EM) algorithm is an iterative method for solving the maximum-likelihood estimates for missing values.

The algorithm proceeds in two steps (a) The E-Step: The expectation step evaluates the posterior probabilities of the unobserved data and (b) M-Step: The subsequent maximization step updates the model parameters using the posterior distribution of the missing data evaluated in the E-step. The EM algorithm is outlined in Algorithm 3.3.

Algorithm 3.3: The EM Algorithm for data with missing values

Initial:

A joint distribution $p(X_{obs}, X_{mis}|\theta)$ over the observation values X_{obs} and the missing values X_{mis}, and the model parameters θ.

The goal is to maximize the likelihood function $p(X_{obs}|\theta)$with respect to θ.

1. Choose an initial setting parameter θ.

   E-step, evaluate $p(\,X_{mis}|X_{obs}, \theta^{old})$

   $$\mathcal{Q}(\theta, \theta^{old}) = \sum_{X_{mis}} p(\,X_{mis}|X_{obs}, \theta^{old})\ \ln p(X_{obs}, X_{mis}|\theta)$$

2. M-step, evaluate θ^{new} given by

   $$\theta^{new} = \arg\max_{\theta} \mathcal{Q}(\theta, \theta^{old})$$

3. Check for convergence of either the log likelihood or the parameter values.

   If the convergence criterion is not satisfied, then let

   $$\theta^{old} \leftarrow \theta^{new}$$

3.4.3.7 Support vector machine imputation (SVMI)

A support vector machine (SVM) can be used to impute the missing values as well. For each attribute in the training set that has missing values, an SVM is trained using all of the examples that have no missing values. It uses the decision attributes (output or classes) as the condition attributes (input attributes) and the condition attributes as the decision attributes, and then SVM is used to predict the missing condition attribute values (Honghai *et al.*, 2005).

This method ignores the original classification value from the dataset and uses the value of the attribute being imputed as the target value. It also ignores any other attributes that have any missing values when generating this new training data. If the attribute being imputed is continuous, the SVMI will use regression to generate the value. If the attribute is continuous, we need to classify it with each of the SVM models and select the value corresponding to the SVM that classifies the example as positive. If more than one SVM generates a positive classification, we randomly select one value.

3.5 Imputation for clinical data

It has been seen that imputation of missing values can alleviate this problem in clinical data (Barnard and Meng, 1999; Abdala and Saeed, 2004; Vorobieva *et al.*, 2007; Gilchrist *et al.*, 2008). However, it should be kept in mind that imputed values are only estimates of original values (Jagannathan and Petrovic, 2009). In this chapter, a variety of methods for handling missing data have been discussed. The main goal of so doing is to understand the mechanisms and study their effect on the mining of data. Real-life clinical data are often incomplete, and can have errors and inconsistencies. Table 3.2 summarises the unusable data in the LIFELAB dataset. It can be seen that missing values arise from various reasons and this is the predominant issue.

Table 3.2: Unusable data while extracting and cleaning the LIFELAB dataset variables for analysis

Reason unusable	**Example**	**# of variables**	**% of total variables**
Missing values	Data item was not collected	438	94.60
Free-text	Enters free text, remarks, description, or comment	61	13.17
Non meaningful	Link ID, update date	94	20.30
Homologous values	Same meaning or duplicated variabless	4	0.86

3.5.1 Missing values imputation for clinical datasets

In what follows now and in subsequent chapters, the following terms and notations will be used. The dataset is considered to be a matrix of predictors $X_i \in X \subseteq \mathbb{R}^n ; i = 1, \dots n$ where n is the number of patients and m is the number of attributes. The outcome Y is fully observed but some of the predictors have missing values. Thus the matrix of predictors can be partitioned as $X = (X_{obs}, X_{mis})$, where X_{obs} and X_{mis} are the observed and missing predictors, respectively.

Seven imputation methods have been discussed in this chapter. Of these methods, MCI, KNNI and EMI are commonly used. However, Luengo et al (Luengo *et al.*, 2011) suggest that FKMI and SVMI are the best methods. However, what is of interest is not only the manner of replacing missing values, but also the changes they bring about to the dataset, primarily the statistical properties. It should be noted that these properties directly affect the classification results and the performance of the models. In what follows these imputation schemes are analysed in terms of the changes to the statistical properties of the datasets.

3.5.2 Metrics for imputation

The importance of the imputation is justified by the imputed data obtained by applying the methods. In this thesis, we evaluate the performance by studying the effect of imputing data by considering the statistics, data distribution and performance indicators.

3.5.2.1 Statistics

In this section, the fundamental statistical values that have been used to describe the data are means and standard deviation. The mean or average, which obtains an average value and the standard deviation, which describes the range of variation of data items.

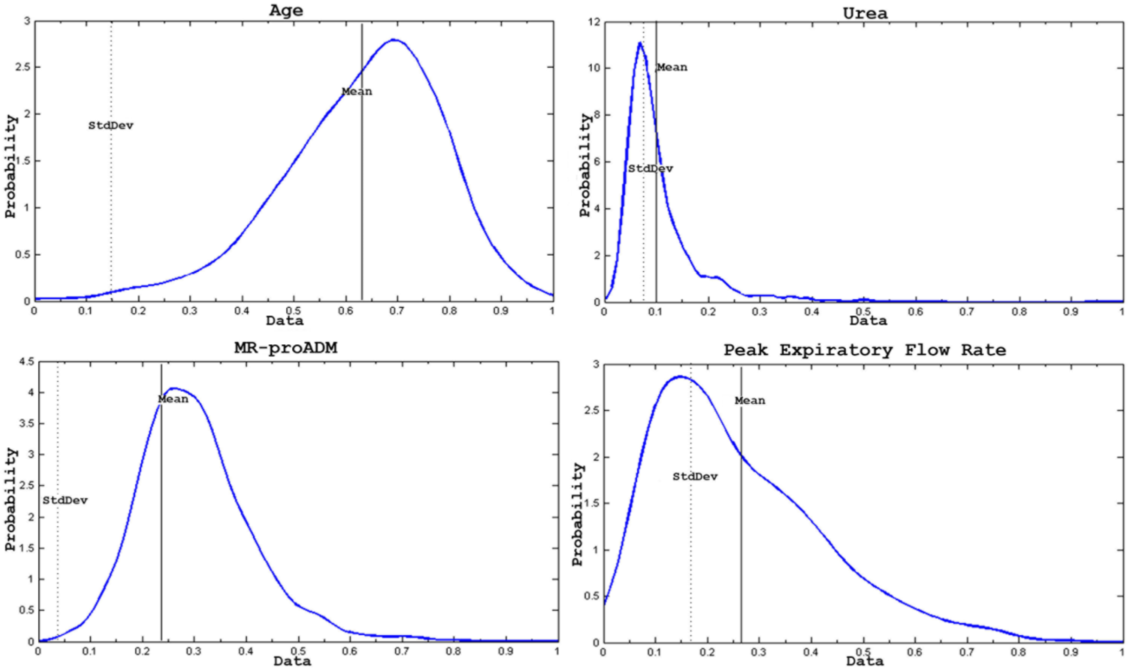


Figure 3.2: Data distribution of variables in clinical dataset

The LIFELAB dataset is used for an illustration to show the missing values problem in a clinical dataset. Fig. 3.2 shows selected variables that contain missing values and presents the data distributions with statistical values. The results show that these variables generate non-normal distributions. Non-normal distribution in the presence of the missing data is considered and this difficulty is handled by a missing value imputation.

Table 3.3 shows the variables with approximately 1-25% missing values. The table compares the statistical values between the original data and the data treated with different missing value imputation methods. Comparing the population mean (μ) and standard deviation (σ) before and after handling missing values with missing value imputation methods, these values are changed when compared with the original data variables containing missing values.

Table 3.3: The μ and σ before and after handling missing values by imputation techniques

Variable	Phosphate		Cholesterol		Uric acid	
Missing values (%)	5.77		10.38		14.04	
Imputation	μ	σ	μ	σ	μ	σ
Original (with missing values)	3.44	48.34	4.81	2.62	0.81	4.25
CMCI	3.26	47.01	4.68	1.20	0.62	2.32
EMI	3.49	47.26	5.75	2.38	0.67	2.32
FKMI	3.38	47.01	4.67	1.19	0.60	2.28
KMI	3.25	47.01	4.69	1.20	0.61	2.28
KNNI	3.25	47.01	4.67	1.20	0.68	2.31
MCI	3.25	47.01	4.67	1.21	0.60	2.28
SVMI	3.25	47.01	4.67	1.20	0.61	2.29
Variable	Iron		MCV		LVEDD	
Missing values (%)	17.5		22.31		24.62	
Imputation	μ	σ	μ	σ	μ	σ
Original (with missing values)	13.54	7.22	91.23	9.81	6.20	6.34
CMCI	13.38	5.69	91.31	8.22	5.86	1.08
EMI	23.76	12.28	100.86	12.44	9.25	1.73
FKMI	12.95	5.69	86.82	11.66	5.81	1.05
KMI	13.33	5.65	91.60	8.12	5.87	1.07
KNNI	13.35	5.68	91.42	8.16	5.88	1.06
MCI	12.92	5.84	91.96	8.43	5.95	1.17
SVMI	13.17	5.67	91.24	8.15	5.94	1.06

The results show that different missing values rates are affected to the statistical values. The variable that contains lower missing values, after imputation μ and σ were changed less than the variable with more missing values.

3.5.2.2 Data distribution

Data distribution is the basis of data analysis, which can describe the characteristics of data. In case of imputation, the data distribution also depicts the effect of the relationship between imputation methods and missing values rates.

Any single performance estimator suffers the risk of being fitted, if many classifiers based on the estimators are compared. Thus, we carefully used a confusion matrix (Chapter 5) and probability density function (*pdf*) (Rubin, 1976; Subramonian, 1998b; Poolsawad *et al.*, 2012b) to evaluate and investigate the performance of the classification and imputation techniques used in experiments. *Pdf* has to be approximated by counting the frequency of occurrence of the event whose probability is being estimated (Subramonian, 1998a). Probability density is simply the probability of a variable existing between two values that bound an interval. The area under the *pdf* is always 1 or 100%.

In a dataset of n records, let there be m attributes such that $X = [x_1, x_2, x_3, \cdots, x_m]$, and the dataset is a matrix of $n \times m$ dimensions. Let x_i^j ; $i \leq m, j \leq n$ be the number of missing values of an attribute x_i. The ideal *pdf* of the attribute x_i, is given by the distribution of $n - j$ values of the attribute $x_i,$ and let this *pdf* be given by $p_1(\bar{x}_i)$ and let $p_2(x_i)$ be the *pdf* of the same attribute once the imputation has been completed.

Definition 1: Given two probabilities p_1 and p_2, let $d(p_1,\ p_2) = \ |p_1 - p_2|$. $p_1 \neq \delta p_2$ *if* $D(p_1,\ p_2) > \delta;$ *else* $p_1 = \delta p_2$. (Zhang et al., 2013)

Definition 2: $p_1(x) \neq \delta p_2(x)$. *if* $(d(p_1(x)||p_2(x)) > \delta$ *; else* $p_1 = \delta p_2$. *d(·) is a metric which measure the distance between two pdfs.* (Subramonian, 1998a)

Pdf of variables, which are continuous would require the evaluation of the differences or distances over the whole range of the attribute. However, in most situations, data is obtained in a discrete manner and in modern tele- health situations, this is more the case. In such cases, a less computationally intensive approach can be used to establish the similarities. If we assume that the discretisation is uniform, then the following can be applied:

Table 3.4: A comparison of the *pdf* with different percentages of missing values

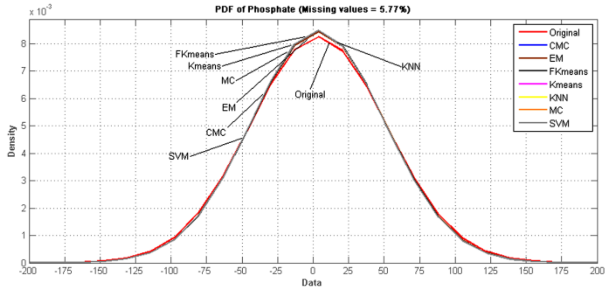
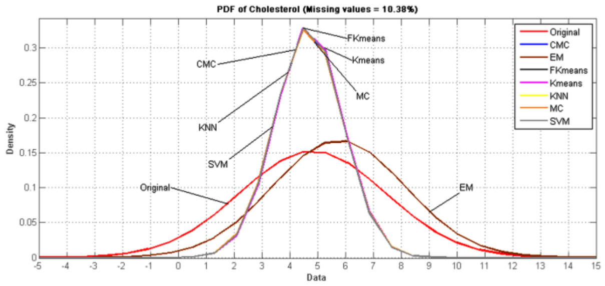
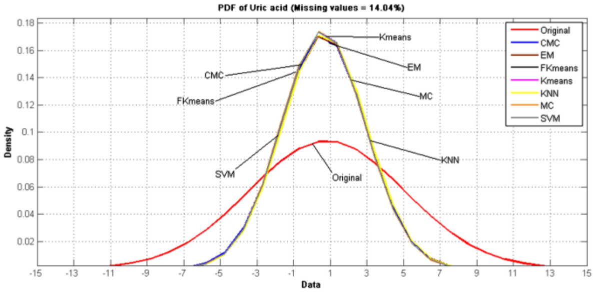
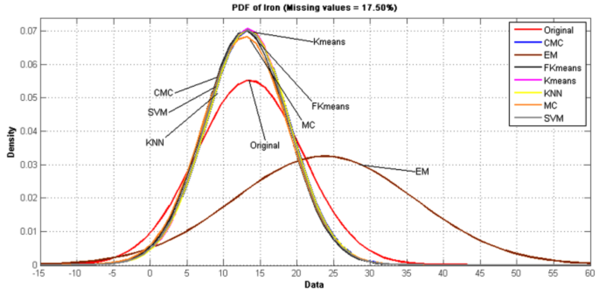
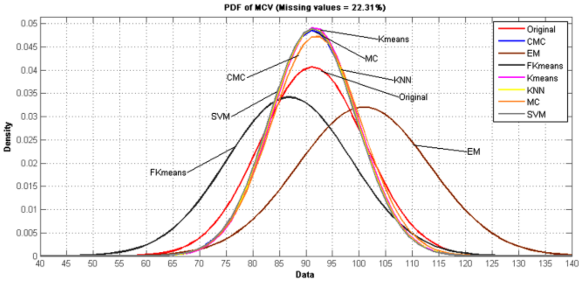
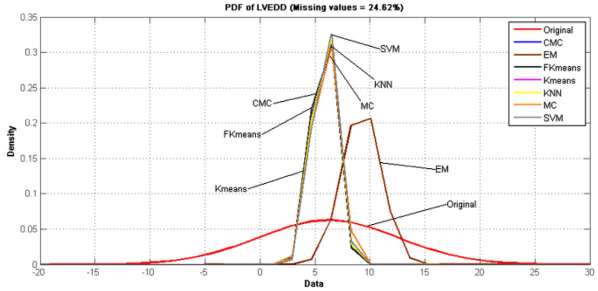
Phosphate (Missing values = 5.77%)	Cholesterol (Missing values = 10.38%)
	
Uric acid (Missing values = 14.04%)	Iron (Missing values = 17.50%)
	
MCV (Missing values = 22.31%)	LVEDD (Missing values = 24.62%)
	

Table 3.5: A comparison of the *pdf* on Phosphate with different percentages of missing values

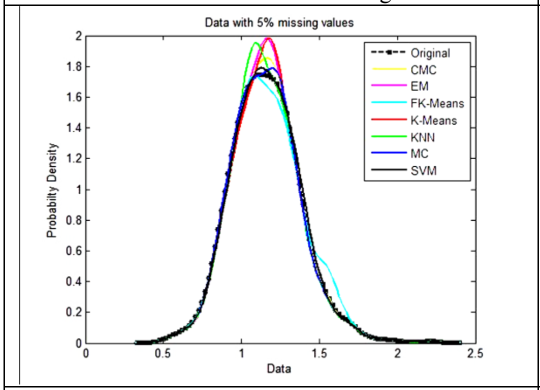
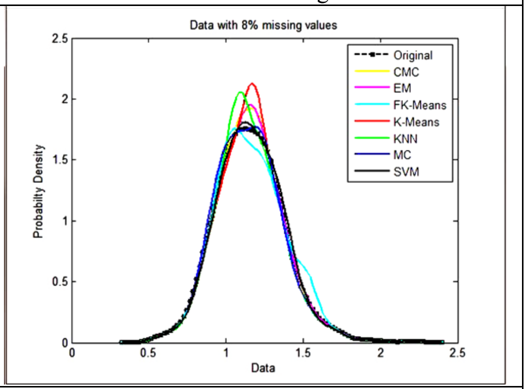
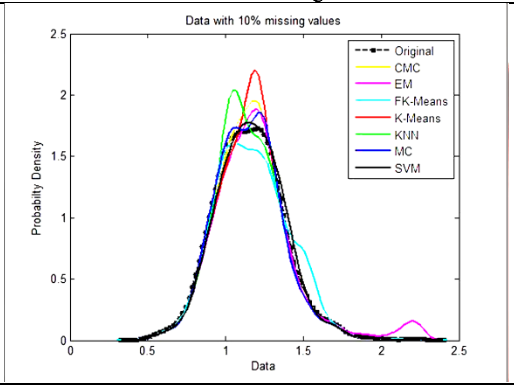
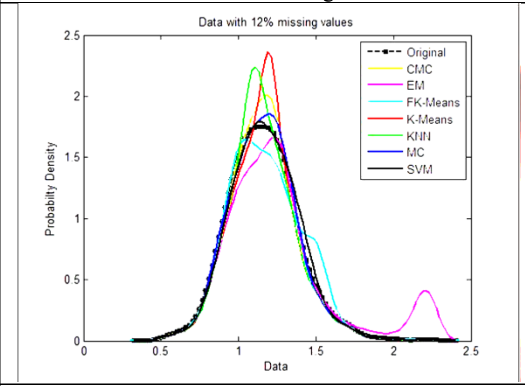
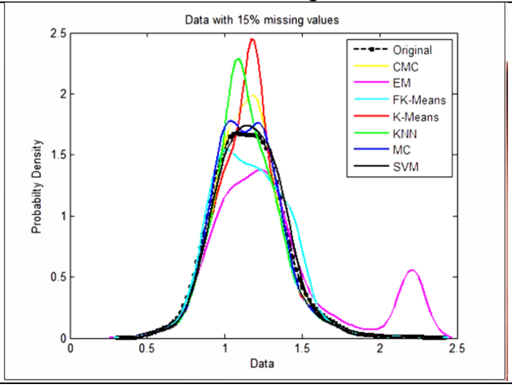
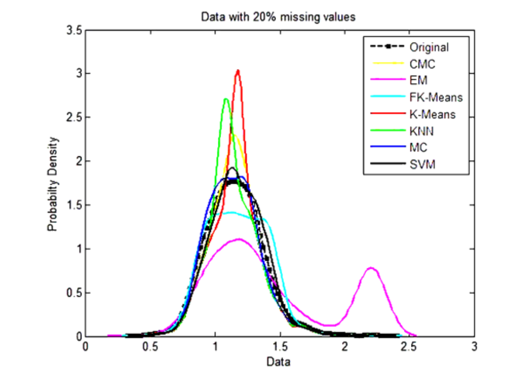
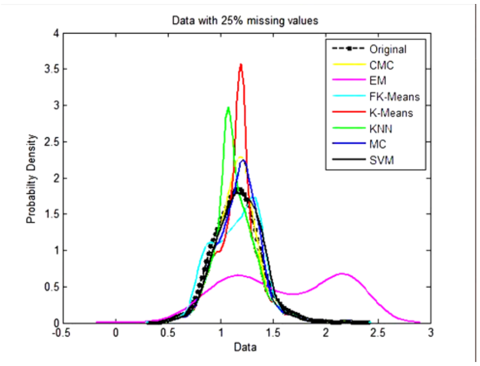
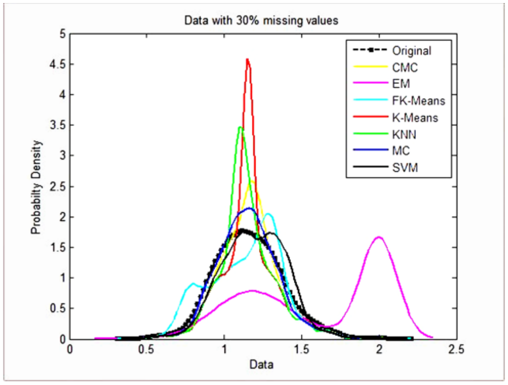
Data with 5% missing values	Data with 8% missing values
	
Data with 10% missing values	Data with 12% missing values
	
Data with 15% missing values	Data with 20% missing values
	
Data with 25% missing values	Data with 30% missing values
	

Table 3.4 presents the data distribution from different percentages of missing values along with different variables and Table 3.5 shows the data distribution (*pdf*) of the different missing value imputation techniques and different percentages of missing values on same variable. The different missing value percentages illustrate different *pdf* distributions; however, they still appeared to be normally distributed after the application of missing value imputation methods.

3.5.2.3 Performance indicators

The performance indicators were selected (N *et al.*, 2011) to evaluate the performance for imputing were 1) prediction accuracy, 2) coefficient of determination, 3) mean absolute error, and 4) root mean square error. The imputed and observed data were compared to select the best method for estimating missing values.

1) Predictive accuracy (PA)

PA is calculated by using

$$PA = \sum_{i=1}^{N} \frac{\left[(X_{mis_i} - \bar{X}_{mis})(X_{obs_i} - \bar{X}_{obs})\right]}{(N-1)\sigma_{X_{mis}}\sigma_{X_{obs}}} \tag{3.5}$$

where N is the number of missing values to be imputed, and $\overline{X}_{obs}$, $\overline{X}_{mis}$ are the averages, $\sigma_{X_{obs}}$ and $\sigma_{X_{mis}}$ their standard deviations. PA values range from 0 to 1, with higher values of PA indicating a better fit.

2) Coefficient of determination (R^2)

R^2 explains how much of the variability in the imputed data can be explained by the fact that they are related to the observed values or how close the points are to the line. It is given by

$$R^2 = \left[\frac{1}{N}\frac{\sum_{i=1}^{N}\left[(X_{mis_i} - \bar{X}_{mis})(X_{obs_i} - \bar{X}_{obs})\right]}{\sigma_{X_{mis}}\sigma_{X_{obs}}}\right]^2 \tag{3.6}$$

R^2 takes on values between 0 and 1, with values closer to 1 implying a better fit.

3) Mean absolute error (MAE)

MAE is the average difference between predicted and actual data values, and is given by

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|X_{mis_i} - X_{obs_i}\right| \tag{3.7}$$

MAE ranges from 0 to infinity and a perfect fit is obtained when MAE = 0.

4) Mean-squared error (RMSE)

RMSE is one of the most commonly used measures of success for numerical prediction. Its value is computed by

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left[X_{mis_i} - X_{obs_i}\right]^2} \tag{3.8}$$

The smaller the RMSE value, the better the performance of the model.

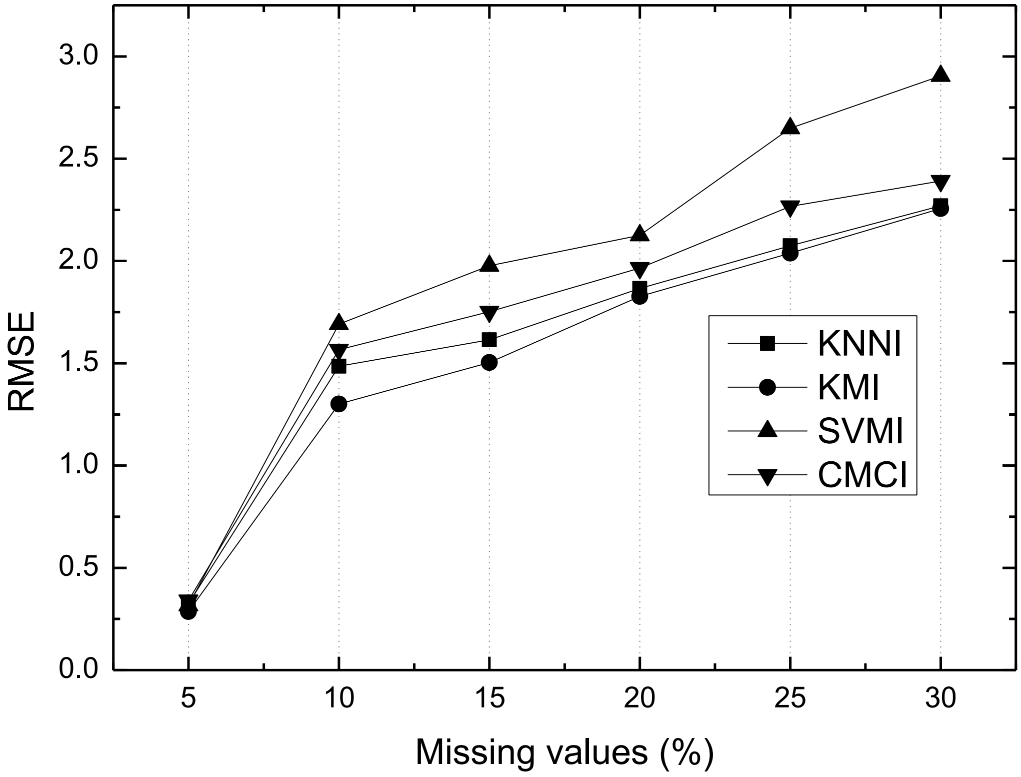


Figure 3.3: Mean-squared error (RMSE) of different numbers of missing value and different imputation methods

Due to RMSE is the most commonly used measures then Fig. 3.3 shows the results of the RMSE of different numbers of missing value and different imputation methods on one variable ('Phosphate' variable). It shows that the RMSE values of KNNI and KMI are lower, which means that the imputed data revealed better performance. With KNNI and KMI, a proximity (similarity and dissimilarity) measure between samples has to be defined for imputing the missing values. According to the imputation decision (Gilchrist *et al.*, 2008) if data is not MCAR and MAR, the imputations using data from similar patients are more appropriate than the methods using entire dataset. However, SVMI and CMCI use the information in whole data to assume the imputed data from the original values. These kinds of method are suitable for handling the MCAR and MAR missing values because there is no relationship between whether missing data and missing or observed data. The result shows that the missing values of this variable are dependent while higher numbers of missing values, the errors between imputed and observed data are increased. Thus, the nature of the data is the most important factor in finding the appropriate imputation method.

As a result, we suggest that the percentages of missing values should not be higher than 25-30 (Collins *et al.*, 2001). We observed that the imputation methods might not be recommended when considering the statistics values and data distribution. In addition, a large proportion of missing values will affect the performance of classification. The results will be shown in Chapter 7.

3.6 Summary

This chapter has shown that handling missing values using the underlined techniques is significant in data mining processes. Often a preliminary exercise is to discard variables with a large percentage of missing values to minimise the problem during mining the data, followed by imputing missing values. An alternative is to ignore missingness by analysing the incomplete data. This chapter looked at various methods for imputing missing values and it also suggested a multi-stage process for handling missing values. Imputation is a necessity, in order to keep the data rich. However, imputation merely imputes a value into the record and this value may not have any relationship to the real value. Thus, it often said that there is a limit to the degree of imputation. Missing values is a major issue as it affects the analytical part of the data process and missing value imputation methods have shown to solve this problem while retaining the original size of the dataset. This strongly suggests that imputed missing values are appropriate to analyse and implement analysis such as confusion matrix and probability distribution methods.

The imputations illustrated in this thesis have been applied to a heart failure dataset, and can be applied to various clinical datasets as these datasets present with similar issues. In theory, data would be precisely distributed but in the real world situation, data distribution is usually imprecise. The pre-processing step provides the story behind the data and tends to help in understanding the nature of the data. It also provides the opportunity to choose an appropriate technique. A key aspect to be kept in mind is that imputation is done before there is a reduction in the dimensionality of the problem, in order to ensure that whatever statistical method we use for imputation, the values inserted are correct as far as possible. In the next chapter, the issues around reduction of dimensionality are discussed.

.

CHAPTER 4

PRINCIPLES FOR DIMENSIONALITY REDUCTION TECHNIQUES

4.1 Introduction

In the previous chapter, the first of three discussing the challenges of missing values posed by clinical datasets was discussed. There are other challenges, for example dimensionality, class imbalance etc. In this chapter, the issues around dimensionality reduction will be discussed, while in Chapter 6 a new method for this will be developed.

There are a few questions that need to be posed with regard to dimension reduction; these are (a) Why do we need dimensionality reduction? (b) Why do clinical datasets need a reduction of dimensions? and (c) How is dimensionality reduction performed on datasets? Developing strategies for data mining can become very complex and time consuming. It should be noted here that dimensionality reduction refers to reduction of the datasets in terms of variables (in the remainder of the thesis, variables, attributes, features and dimensions will be used in an interchangeable manner). In this chapter, dimensionality reduction in a dataset is the same as feature extraction or feature selection.

In the following sections, different techniques for both feature extraction and selection are surveyed. A popular feature extraction algorithm is based on principal components and is known as Principal Component Analysis (PCA); it is extended to include nonlinearities in Nonlinear Principal Component Analysis (NL-PCA). These and their variations are discussed in this chapter. It should be noted that even though feature extraction algorithms reduce the numbers of dimensions, the features they provide are essentially synthetic. With regard to feature selection, five methods are examined, which all use a "feature importance ranking measure" for ranking each feature according to its discriminative capability, e.g. *t*-test (Zhou and Wang, 2007), entropy ranking (Fayyad and Irani, 1993; Liu *et al.*, 2002), Bhattacharyya distance (Theodoridis and Koutroumbas, 2006; Guo *et al.*, 2008), ROC (Fawcett, 2006; Theodoridis and Koutroumbas, 2006) and Wilcoxon test (Gibbons and Chakraborti, 2003; Liao *et al.*, 2006). The next stage investigates these procedures further. Based on a ranked list, different subsets of features (variables) are selected, and then tested on their ability to discriminate the classes present. In general, the optimal set of features is the one with the high classification accuracy and the minimum size.

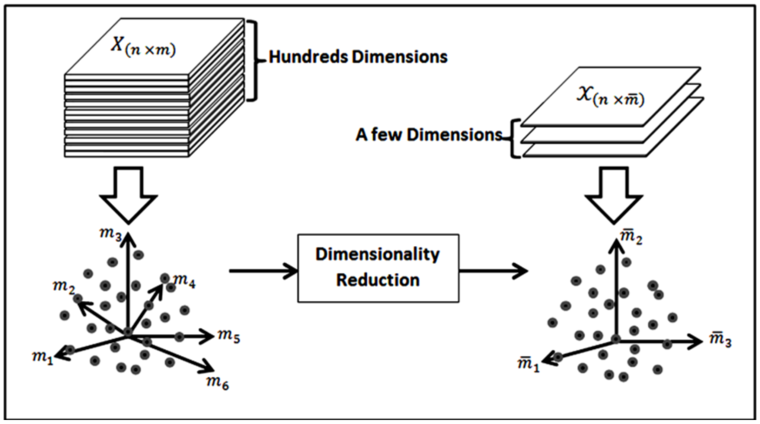


Figure 4. 1: Dimensionality reduction from high dimensionality to low dimensionality

4.2 Time complexity

A number of features contribute to classification complexity (Ho and Basu, 2002), so dimensionality reduction is important to improve the efficiency of classification. The O-notation, as used in algorithm analysis, is a simplification tool for complex expressions which arise due to the number of performed operations, assuming a given model of computation (Rutanen *et al.*, 2013). In other words, it describes the performance or complexity of an algorithm. Here, time complexity is expressed as the relationship between the number of features (the size of the input) and the amount of time required to execute an algorithm (run time for the algorithm).

The prevailing classification time of the feature selection will be changed according to the time complexity of several classification approaches. Suppose we have *n* data points in *m* dimensional space and a binary class variable. Consequently, examples of the average time complexity of classifiers are revealed. The k-nearest neighbour (kNN) classifier requires *O(nm)* (Kibriya and Frank, 2007), for the decision tree, the time is $O(nm^2)$ (Martin and Hirschberg, 1996), and for random forest it is $O(n^m)$ for computing a single tree (it depends on the number of trees) (Biau *et al.*, 2008), linear Support Vector Machine (SVM) has $O(nm)$, $O(nm\log(m))$ for regression problems (Joachims, 2006) and multi-layer perceptron (MLP) can be computed in $O(m)$ (Yang and Amari, 1998). These examples show how much each feature contributes to the time complexity. Therefore, when the numbers of features are changed this effect becomes more noticeable on the execution time of the algorithm. With approximate algorithms this can be brought down considerably at the expense of accuracy so that dimensionality reduction (Fig. 4.1) is used to deal with this problem. At the same time, the classification performance of the system is expected to improve, to yield an optimal set of features, a technique can be considered superior in terms of time complexity, if it provides a lower error rate with the same complexity. To evaluate the feature selection methods in this thesis, the trade-off between accuracy and redundancy rate is analysed in Chapter 7. This can identify the most relevant features that provide nearly the same classification performance as the full set of features.

4.3 Feature extraction

Feature extraction is the process of creating a representation by the transformation of the original data into a new set of synthetic features (or components/dimensions). The technique requires the original feature set in order to determine the transformations required for the new features. A set of new synthetic features are defined by functions over all the features; in other words, the original set of features are projected onto a lower dimensional space while preserving as much information as possible (Fig. 4.2).

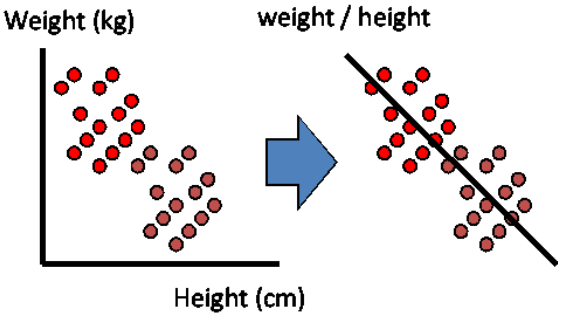


Figure 4. 2: Dimensionality reduction: feature extraction

Extracted features (Kramer, 1991)are a set of derived variables, functions of the original problem variables, which efficiently capture the information contained in the original data. Good features (Foldiak, 1989) will reduce dimensionality with only a minimal loss of information. The quality of the process is dependent on the number of features extracted; often, the larger the number, the better the eventual result. The trade-off here is between the desired accuracy, and the total operational (computational) cost. This trade-off is not an exact science, but essentially a design feature where the decision is made by the user. However, the primary interest is often to select the fewest possible features/components, in order to make these problems more tractable (Jirapech-Umpai and Aitken, 2005).

Feature extraction combines attributes into a new reduced set of features and creates new features based on transformation or combination of the original feature set. Principal Feature Analysis (PFA) (Lu *et al.*, 2007) is a feature grouping that chooses a feature from each group and this technique is based on the popular principal component analysis (PCA) (Jolliffe, 2005) technique. PCA is an unsupervised approach to project the original feature space onto a low-dimensional subspace. It obtains a set of transformed features rather than a subset of the original features.

There have been several successful methods (Yu and Liu, 2004; Wang *et al.*, 2007b), which have been developed and implemented. These methods are useful in applications where the labels associated with the reduced dimensions are not important. However, in clinical decision support systems, retention of labels is important from a clinical application point of view.

4.3.1 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a commonly used method for extraction of features (Tabachnick and Fidell, 1996; Jolliffe, 2005), and is the solution of choice for researchers who are primarily interested in reducing a large number of variables down to a small number of components. PCA is essentially an unsupervised pattern recognition technique and one of the many multivariable techniques available for data analysis. It describes the variation in multivariate data in terms of a set of uncorrelated variables by using singular value decomposition to find singular vectors and values of a centred version of the data matrix. Essentially the method creates new uncorrelated features that are linear combinations of the original features. Consider a data set which is represented by X, which is an nxm matrix, where *m* is the number of features (variables/attributes/dimensions) and *n* the number of records (samples).

$$X = \begin{bmatrix} x_{11} & x_{12} \cdots & x_{1m} \\ x_{21} & x_{22} \cdots & x_{2m} \\ \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & x_{nm} \end{bmatrix} \xrightarrow{reduce\ features} \Phi = \begin{bmatrix} \Phi_{11} & \Phi_{12} \cdots & \Phi_{1p} \\ \Phi_{21} & \Phi_{22} \cdots & \Phi_{2p} \\ \vdots & \vdots & \vdots \\ \Phi_{n1} & \Phi_{n2} & \Phi_{np} \end{bmatrix} (p < m)$$

The PCA (Zabiri *et al.*, 2009) allows a linear mapping of data from $\Re^m$ to $\Re^p$, with $p < m$. The optimal transformation of X via the PCA is shown in Eq. (4.1)

$$X = TP^T + e \tag{4.1}$$

where T is called the score matrix with dimension $n \times p$, p is the column indicating the principal components ($p < m$) of X, rows of SCORE correspond to observations. P represents the coefficient matrix, also known as loadings matrix with a dimension of $n \times p$. The Euclidean norm of the residuals $\|e\|$ must be minimised for a given number of principal components for the optimality condition to be satisfied. This is achieved if the columns of P are the eigenvectors corresponding to the λ largest eigenvalues of the covariance matrix of X. If $P^T P = I$, the linear mapping of the PCA is given by Eq. (4.2)

$$\underline{T} = \underline{X}P \tag{4.2}$$

where $\underline{X}$ represents a row of X, a single data vector, and $\underline{T}$ represents the corresponding row of T (the coordinates of $\underline{X}$ is the reduced p-dimensional variable space). The loadings matrix, P, are the coefficients for the linear transformation, and essentially define the orientation of the principal component plane with respect to original m-variables. The information lost in this mapping can be assessed by reconstruction of the measurement vector by reversing the projection back to $\Re^m$

$$\underline{X'} = TP^T \tag{4.3}$$

where $\underline{X'} = \underline{X} - e$ is the reconstructed measurement error. The eigenvalues λ of covariance matrix X are calculated.

$$det(X - I\lambda) = 0 \tag{4.4}$$

An eigenvector of a square matrix X is a scalar λ and a nonzero vector v is satisfactory.

$$Xv = \lambda v \tag{4.5}$$

Hotelling's T-squared is a multivariate approach to select principal components by computing the distance between samples means using the metric of covariation. Hotelling's T-squared statistic is the sum of squares of the standardised scores.

$$T^2(x) = \sum_{i=0}^{K} \frac{t^2(i)}{\lambda_i^2} \tag{4.6}$$

where $t(i)$ is the ith element in the vector t.

The principal components (PCs), that is, the eigenvectors, are shown by their associated eigenvalues. A cut-off percentage of the number of principal components is related to Hotelling’s; commonly it is in the range of 70% to 90% and this is determined by applying the cumulative percentage of total variation. It can be higher or lower depending on dataset (Jolliffe, 2002) and the size of variance of principal components takes into consideration Kaiser’s rule (Kaiser, 1960) which states that any PC with variances less than 1 contains less information and so is not worth retaining.

The power of PCA (Schwartzman *et al.*, 2001) resides in the fact that, as the data are decomposed into uncorrelated components that are arranged in order of decreasing variance, most of the population variance is contained in the first principal component. The effect is that of reducing the number of variables required to represent the data.

4.3.2 Nonlinear Principal Component Analysis (NLPCA)

NLPCA is an extension of principal component analysis. While PCA identifies only linear correlations between variables, NLPCA uncovers both linear and nonlinear correlations, without restriction of the characteristics of the nonlinearities present in the data. NLPCA, like PCA (Kramer, 1991), is used to identify and remove correlations among problem variables and is an aid in dimensionality reduction, visualisation, and exploratory data analysis. The key difference between PCA and NLPCA is that NLPCA allows arbitrary nonlinear mapping from $\Re^m$ to $\Re^p$ whereas PCA only allows linear mapping(Zabiri *et al.*, 2009).

The NLPCA method uses artificial neural network (ANN) training procedures to generate nonlinear features. Consider a mapping of the type in Eq. (4.7).

$$\underline{T} = \underline{G}(\underline{X}) \tag{4.7}$$

where $\underline{X}$ represents a row of an original data matrix $\underline{\underline{X}}$ with dimension $n \times m$ (n = number of observations, m = the number of variables). $\underline{T}$ represents the corresponding row of the scores of matrix $\underline{\underline{T}}$ with dimension $n \times p$ (n = number of observations, p = the number of principal components ($p < m$)). In this equation, $\underline{G}$ is a nonlinear vector function, composed of p individual nonlinear functions; $\underline{G} = \{G_1, G_2, G_3, \dots, G_p\}$, such that, if T_i represents the ith element of $\underline{T}$,

$$T_i = G_i(\underline{X}) \tag{4.8}$$

By similarity to the linear case, G_1 is referred to as the primary nonlinear principal component, and G_i is the ith nonlinear principal component of $\underline{X}$.

The inverse transformation, the reconstruction of the original data is accomplished by a second nonlinear vector function $\underline{H} = \{H_1, H_2, H_3, \dots, H_m\}$:

$$X'_j = H_j(\underline{T}) \tag{4.9}$$

The lost information is measured by:

$$\underline{e} = \underline{X} - \underline{X'} \tag{4.10}$$

where $\underline{e}$ is the resulting error, measured by $|\underline{e}|$ for individual measurement vectors, or $\|\underline{e}\|$ for the overall dataset. The function $\underline{G}$ and $\underline{H}$ are selected to minimise $\|\underline{e}\|$.

To generate $\underline{G}$ and $\underline{H}$, any nonlinear function to an arbitrary degree of precision:

$$nf_k = \sum_{j=1}^{N_2} w_{jk2}\sigma\left(\sum_{i=1}^{N_1} w_{ij1}u_i + \theta_{j1}\right) \tag{4.11}$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{4.12}$$

Eq. 4.11- 4.12 describe for a feedforward artificial neural network (ANN) with N_1 inputs, a hidden layer comprised of N_2 nodes with sigmoid transfer functions, and a linear output node for each k. In equation 4.11, w_{ijk} represents the weight on the connection from node i in layer k to node j in layer $k+1$, and θ are the bias nodes.

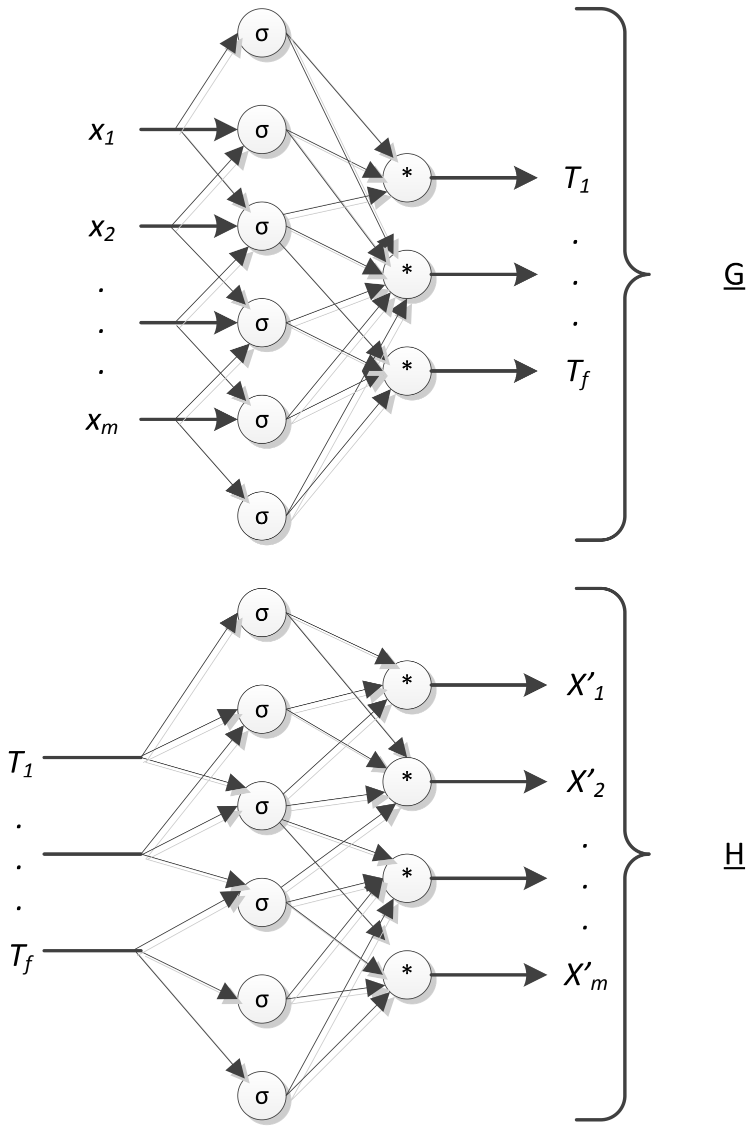


Figure 4. 3: Networks implementing mapping and demapping function

The ability of the neural network to fit arbitrary nonlinear functions depends on the presence of a hidden layer with nonlinear nodes. Therefore, the architecture for the networks shown in Fig. 4.3, is representing $\underline{G}$ and $\underline{H}$, is as follows. The network for $\underline{G}$operates on the rows of $\underline{\underline{X}}$ and has m inputs. The hidden layer $\underline{G}$ is called the ‘mapping’ layer. This layer contains M_1 nodes with sigmoidal transfer functions, where M_1>p. The output of the network is the projection of input vector into feature space, and therefore contains p nodes. The output nodes can have linear or sigmoidal transfer functions, without affecting the generality of $\underline{G}$. The functionG_i, the ith nonlinear principal component, is defined by the weights and biases on the connections from the input to the ith output.

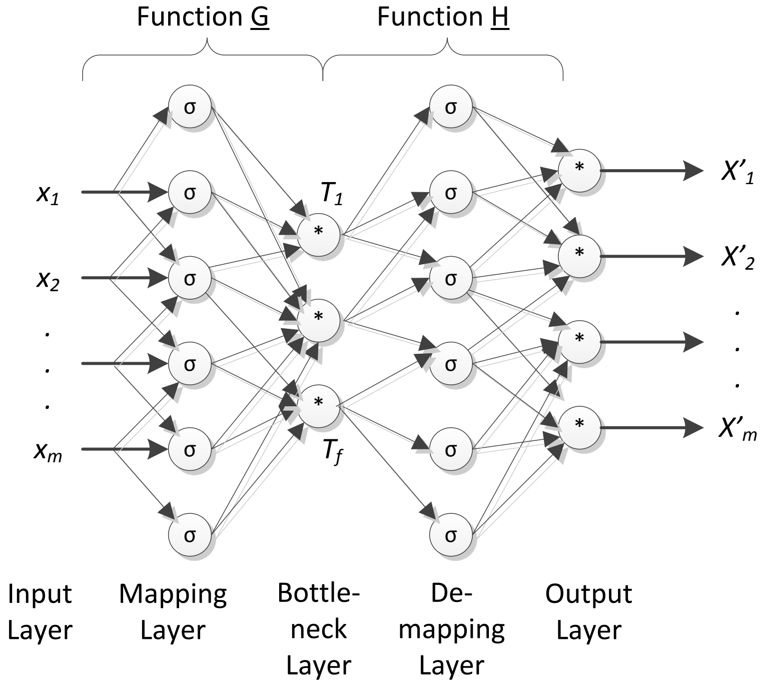


Figure 4. 4: Network architecture for simultaneous determination of f nonlinear components using an autoassociative network
(σ indicates sigmoidal nodes, * indicates sigmoidal or linear nodes)

The network represents the inverse mapping function $\underline{H}$ takes the rows of $\underline{\underline{T}}$ as inputs and accordingly has p inputs. The hidden layer, which is referred to as the ‘de-mapping’ layer, contains M_2 nodes with sigmoidal transfer functions, where $M_2 > p$.

The output layer yields the reconstructed data, $\underline{X'}$, and thus contains m nodes. The nodes of the output layer can have linear or sigmoidal transfer function. The weights and biases connecting the inputs to the jth output node define the function H_j. Combining the two networks in series so that $\underline{G}$ feeds directly into $\underline{H}$, a network is obtained whose inputs and desired output are known (Fig. 4.4).

Therefore, the performance of an autoassociative network with only one internal layer of sigmoidal nodes is often no better than linear PCA (Kramer, 1991). The reconstructed outputs $\underline{\underline{X'}}$ match the inputs $\underline{\underline{X}}$ as closely as possible. Training is complete when e, the sum of squared errors given in Eq. 4.13:

$$e = \sum_{p=1}^{n}\sum_{i=1}^{m}(X_i - X'_i)_p^2 \tag{4.13}$$

e is the square of $\|\underline{e}\|$, the optimality criterion used in PCA. The data are propagated through $\underline{G}$ to project the data into low-dimensional feature space.

In Chapter 7 it will be discussed in full how the results show that the NLPCA successfully reduces the dimensions and produces a feature space map resembling the actual distribution of the underlying system parameters. The NLPCA introduced here is a general purpose feature extraction algorithm producing features that retain the maximum possible amount of information from the original dataset for a given degree of data compression. Information preservation assures that the selected features will be useful in most situations, independent of the ultimate application. The NLPCA is commonly seen as a nonlinear generalization of the PCA. It generalizes the principal components from straight lines to nonlinear. Thus, the subspace in the original data space which is described by all nonlinear components is also curved.

NLPCA (Kramer, 1991) will describe the data with greater accuracy and/or by fewer features than PCA, provided that there are sufficient data to support the formulation of more complex mapping functions.

4.4 Feature selection

Feature selection, also known as subset selection, is a process that selects the most relevant attributes, in other words, it finds the best subset of the input feature (Fig. 4.5). In a sense it not only reduces the dimension of the system, but at the same time reduces the complexity and processing time while improving system performance on some dependent measure. A general feature selection algorithm is often composed of three components: an evaluation function, a performance function and a search algorithm. The evaluation function inputs a feature subset and outputs a numeric evaluation. The performance function provides the optimal subsets appropriate for classification. The search algorithm performs the search of an appropriate subset of features. These algorithms can be grouped into three categories, namely exponential, randomized and sequential.

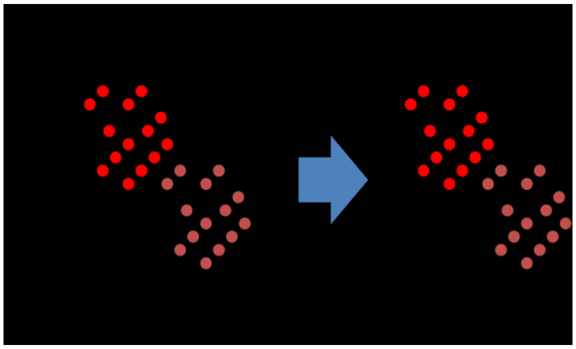


Figure 4. 5: Dimensionality reduction: feature selection

There are two general forms of feature selection procedures: (a) a wrapper model and (b) a filter model (Yu and Liu, 2004). The wrapper model uses the predictive accuracy of a pre-determined learning algorithm to determine the goodness of the selected subsets. The learning algorithm is run with various subsets of features and the learner that performs the best is chosen. In contrast, the filter model presents the data with the chosen subset of features to a learning algorithm. It separates feature selection from classifier learning and selects feature subsets that are independent of any learning algorithm (Yom-Tov and Inbar, 2002; Jirapech-Umpai and Aitken, 2005). In comparison to the wrapper model, the filter model is computationally efficient; however, the filter model is known to perform much worse than the wrapper model.

Feature selection selects the most relevant attributes and tries to find the best subset of the input feature set. Feature selection algorithms are divided into two categories, the filter model and the wrapper model. The filter model (Yu and Liu, 2004) relies on general characteristics of the data to evaluate and select the subset of features without involving any mining algorithm. The wrapper model (Kohavi and

John, 1997) requires a data mining algorithm to search for features. Its aim is to improve the performance of the subset of features but it also tends to be more computationally expensive than the filter model. Correlation-based Feature Subset Selection (CFS) (Dag *et al.*, 2012) uses a search algorithm along with a function to evaluate the merit of feature subsets. The heuristic by which CFS measures the "goodness" of feature subsets takes into account the usefulness of individual features for predicting the class label along with the level of inter-correlation among them (Hall and Smith, 1999). Good feature subsets contain features highly correlated with the class (predictive), yet uncorrelated with each other (not predictive). Examples include Information Gain Attribute Evaluation (Novakovic, 2009) and Symmetrical Uncertainty (Yu and Liu, 2003; Ali and Shahzad, 2012).

A key aspect, which needs to be considered when selecting a subset of features, is the metrics used for determining the relevance or redundancy of a particular feature. An optimal subset of features should contain a set of robust and relevant features along with a set of weak features (Omid, 2011). This allows for the selection of features with a positive Z-score (Steinbach *et al.*, 2000). It is possible to obtain different selection of subsets of features depending on the criterion used. Thus, the subset obtained using a statistical correlation criterion would be different from when mutual information is used. In the following sections of this thesis it will be demonstrated how five feature selection techniques, *t*-test, entropy ranking, Bhattacharyya distance, ROC and Wilcoxon test will be used.

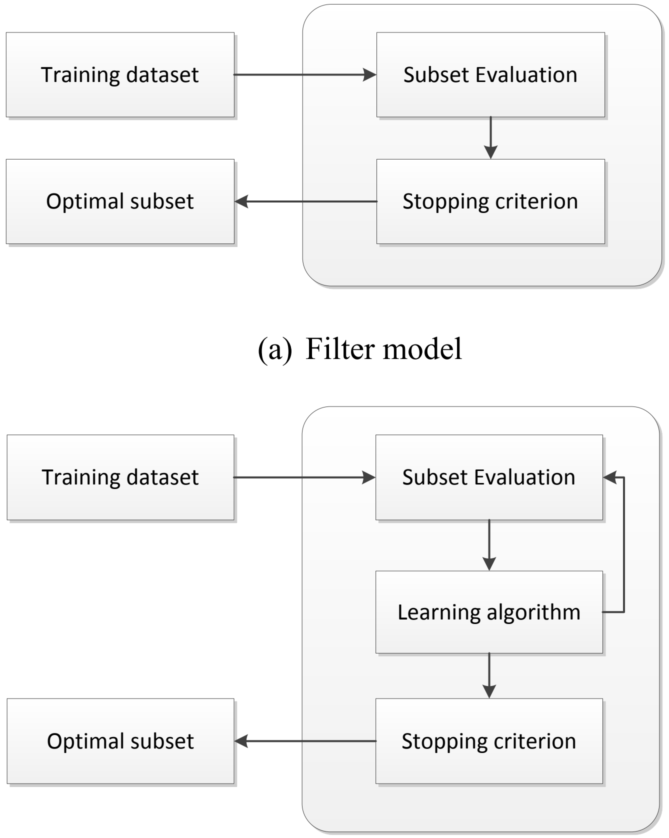
Algorithm 4. 1: Algorithm for feature selection

```
input:     $X_F(F_0, F_1, \dots, F_{m-1})$  // a  training dataset with $m$ features
           $S_0$                             // subset from which to start the search
           $E_F$                             // subset evaluation
           $\delta$                          // a stopping criterion
output: $S_{best}$                           // an optimal subset
begin
initialize: $\mathrm{S_{best}} = \mathrm{S_0}$ ;
           $\gamma_{best} = eval(S_0, X_F, E_F)$;
do begin
           $S = generate\ (X_F)$; // generate a subset for evaluation
           $\gamma = eval(S, X_F, E_F)$; // evaluate the current subset $S$ by $E_F$
if ($\gamma\ is\ better\ than\ \gamma_{best}$)
           $\gamma_{best} = \gamma$;
           $S_{best} = S$;
end until ($\delta\ is\ reached$);
return $S_{best}$ ;
end;
```

// E_F is mostly differed between filter and wrapper model. Filter usually uses criteria not involving any machine learning, whereas wrapper uses the performance of a learning machine (see Fig. 4.6).

1) Wrapper model: it proposes a learning task as the evaluation criterion to select a variable set. Here the aim is to select the variable set that yields the best results in the learning task. Unfortunately, the computational cost of these methods is high.

2) Filter model: it proposes evaluation functions called independent criteria. An independent criterion tries to evaluate the goodness of a feature subset by exploiting the intrinsic characteristics of the training data without involving any learning task.



(a) Filter model

(b) Wrapper model

Figure 4.6: The subset evaluation of filter and wrapper models

Table 4.1: Example algorithms for feature selection

Filter Model	**Wrapper Model**
• Correlation Based Feature Selection (CFS) (Dag *et al.*, 2012) • Fast Correlation Based Filter (FCBF) (Yu and Liu, 2003; Senliol *et al.*, 2008) • *t*-Test feature selection (Zhou and Wang, 2007) • Clearness-based feature scoring scheme (CBFS) (Seo and Oh, 2012) • Entropy ranking (Fayyad and Irani, 1993; Liu *et al.*, 2002) • Discrete Function Learning (DFL) algorithm (Zheng and Kwoh, 2011) • Info Gain (Dag *et al.*, 2012) • Gain Ratio (Dag *et al.*, 2012) • Mutual Information for Feature Selection (MIFS) (Battiti, 1994) • Unsupervised Feature Subset Selection (UFSS) (Søndberg-Madsen *et al.*, 2003) • Recursive Feature Selection Based on Minimum Redundancy Maximum Relevancy (RFS-MRMR) (Yuansheng *et al.*, 2010) • Unsupervised feature selection scheme for nominal data (UFSN) (Chow *et al.*, 2008)	• Artificial Neural Net Input Gain Measurement Approximation (ANNIGMA) (Chun-Nan *et al.*, 2002) • Sequential Floating Forward Selection (SFFS) (Ververidis and Kotropoulos, 2008; Ververidis and Kotropoulos, 2009) • Unsupervised Feature Subset Selection (UFSS) (Søndberg-Madsen *et al.*, 2003) • Feature selection using feature similarity (FSFS) (Mitra *et al.*, 2002) • Unsupervised Feature Selection for Relation Extraction (RLFS) (Jinxiu *et al.*, 2005)

The hybrid model, which was developed at a later time, attempts to take advantage of the two models by combining the different evaluation criteria. The aim here is to design a heuristic that exploits both concepts and produces a better accuracy in the learning task. A typical hybrid algorithm makes use of both models and independent measure and a learning algorithm to evaluate feature subsets. It uses the independent measure to decide the best subsets for a given cardinality and uses the learning algorithm to select the final best subset among the best subsets across different cardinalities.

4.4.1 *t*-Test method

The Student's *t*-test approach uses statistical tools to assess whether the means of two classes are statistically different from each other. It calculates a ratio between the difference of two-class means and the variability of two classes. The use of *t*-test is limited to two class challenges. This method has been found to be efficient in a variety of application domains, as shown in the following two examples:-

(a) Genotype research (Liu *et al.*, 2002; Coetzee, 2005; Zhou and Wang, 2007) where the problem is one of evaluating differential expressions of genes from two experimental conditions.

(b) The ranking of features for mass spectrometry (Wu *et al.*, 2003; Levner, 2005) and microarray data (Jaeger *et al.*, 2003; Su *et al.*, 2003).

For multi-class problems the procedure requires the computing of a *t*-statistic value for each feature corresponding to each class. This is done by evaluating the difference between the mean of one class and all the other classes, where the difference is standardized by within-class standard deviation (Eq. 4.16).

$$t(x_i) = \frac{(\bar{x}_1(x_i) - \bar{x}_2(x_i))}{\sqrt{\left({\sigma_1}^2(x_i)/_{n_1} + {\sigma_2}^2(x_i)/_{n_2} \right)}} \tag{4.16}$$

where $t(x)$ is the *t*-statistics value for the number of features and $\bar{x}_1$, $\overline{x}_2$ are means of classes 1 and 2, while ${\sigma_1}^2, {\sigma_2}^2$ are the within-class standard deviations of classes 1 and 2, n_1 is the number of all the samples in class 1 while n_2 is the number of samples in class 2.

4.4.2 Entropy ranking

The *t*-test approach utilizes some statistical properties to determine the required features. Entropy based approaches not only take into account the statistical properties of the features, but also the compactness and density of the data for variables. Entropy is a measure of the information conveyed by the probability distribution function of a particular variable/feature. Using this entropy, Fayyad (Fayyad and Irani, 1993) suggests a cut-off point selection procedure by using class entropy of subset. In general, if we are given a probability, (·) , then the information conveyed by this distribution, also called the Entropy of *P*, is:

$$Ent(X) = -\sum_{i=1}^{n} P(Y_i, X)\, log\big(P(Y_i, X)\big) \tag{4.17}$$

$$Ent(X) = -\sum_{i=1}^{n} \frac{Y_i}{X}\, log\frac{Y_i}{X} \tag{4.18}$$

where the $Ent(X)$ measures the amount of information required to specify the classes in X. X is a set of attributes and $P(Y_i, X)$ is the proportion of examples in X consisting of class Y in the jth feature. The entropy values are sorted in an ascending order and those features with the lowest entropy values are considered.

4.4.3 Bhattacharyya distance

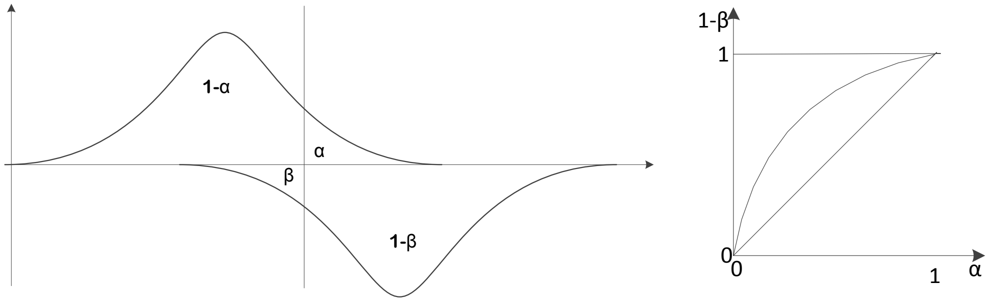
The probability of error is the measure that finds an optimum of feature effectiveness. Bhattacharyya distance is related to the upper bound of error probability (Xuan *et al.*, 1996; Reyes-Aldasoro and Bhalerao, 2006). The Bhattacharyya distance has been used as a *class separability measure* to evaluate the statistical dependence between two random variables and so can be used to measure the utility of selected features to classification (Theodoridis and Koutroumbas, 2006; Guo *et al.*, 2008). In other words, this method selects features by utilising an error estimation selection and finding the subset of features with the lowest classification error (Choi and Lee, 2003; Theodoridis and Koutroumbas, 2006). For two normally distributed classes, the Bhattacharyya distance is defined as follows:

$$B = \frac{1}{8}\left(\mu_i - \mu_j\right)^T \left(\frac{\Sigma_i + \Sigma_j}{2}\right)^{-1} \left(\mu_i - \mu_j\right) + \frac{1}{2}\ln\frac{\left|\frac{\Sigma_i + \Sigma_j}{2}\right|}{\sqrt{|\Sigma_i||\Sigma_j|}} \qquad (4.19)$$

where μ_i and Σ_i are the mean vector and covariance matrix of class i, respectively. Thus, the greater the difference of variances, the smaller the error bound (Theodoridis and Koutroumbas, 2006).

4.4.4 Receiver Operating Characteristics (ROC) Curve

ROC is commonly used to evaluate an algorithm (or classification accuracy) (Fawcett, 2006). This feature selection method uses the ROC curve to measure the individual significance of input variables. The hypothesis tests presented offer statistical evidence about the difference of the mean values of a single feature in the various classes (Theodoridis and Koutroumbas, 2006).



(a) (b)

Figure 4.7: The overlapping of two classes and the ROC curve

(a) overlapping *pdf*'s of the same feature in two classes

(b) the resulting ROC curve (Theodoridis and Koutroumbas, 2006)

Fig. 4.6(a) illustrates an example of two overlapping probability density functions (*pdf*) describing the distribution of a feature in two classes, together with a threshold. This decision is associated with an error probability, α, of reaching a wrong decision concerning 'class 1'. This is equal to the shaded area under the corresponding curve (similarly, β is associated with 'class 2'). If the two distributions have complete overlap, then for any position of the threshold, $\alpha = 1 - \beta$. Fig. 4.6(b) demonstrates a case corresponding to a straight line, where the two axes are α and $1-\beta$. Thus, the area varies between zero, for complete overlap, and ½ (the area of the upper triangle), for complete separation, and it is a measure of the class discrimination capability of the specific feature. Instead, the Area Under the ROC Curve (AUC) can be used for feature selection methods by utilising the requirement of AUC maximization (Wang and Tang, 2009), and Algorithm 4.3 presents the algorithm for selecting the feature by calculating AUC.

```
Algorithm 4.2: ROC feature selection by calculating AUC

$F = \{f_1, f_2, \ldots, f_m\}$
for $i$ = 1 to $m$ do
        AUC[i]←AUC score of the $f_i$;
        $AUC = \frac{\sum_{i=1}^{N_1}(r_1 - i)}{N_1 \times N_2} = \frac{\sum_{i=1}^{N_1}(r_1) - \frac{N_1(N_1+1)}{2}}{N_1 \times N_2}$
        //$r_1, r_2, \ldots, r_{N_1}$ be the rank of samples
end
sort(AUC);
//pick out $k$ features with highest AUC;
```

4.4.5 Wilcoxon rank sum test

The *Wilcoxon rank sum test* or *Mann-Whitney U test* or *Mann-Whitney-Wilcoxon test* is used to test whether two groups come from the same distribution;

values in the two different groups should have values somewhat equally distributed between the two. This test is applied for feature selection to decrease the dimensionality. The accuracy rates are ranked in descending order of features (Liao *et al.*, 2006). The *Wilcoxon rank-sum test* statistic is defined (Gibbons and Chakraborti, 2003) as follows:

$$W_N = \sum_{i=1}^{N} X_i \ , X = \ (X_1, X_2, \dots, X_N) \tag{4.20}$$

where X_i is the indicator random variable, the vector X , indicates the rank-order statistics of combined samples and in addition identifies the sample to which each observation belongs. Thus, the set of features can select as follows:

(1) Combine the samples into one sample of W_i's. Order data in the combined sample $W_{(1)} \leq \ W_{(2)} \ \leq \ldots \leq W_{(N_1+N_2)}$

(2) Assign rank i to the i^{th} smallest observation

(3) Let R_1^{obs} is sum of ranks in samples 1, and also R_2^{obs} is sum of ranks in samples 2

(4) $W_1 = \ R_1^{obs} - \frac{N_1(N_1+1)}{2}$, $W_2 = \ R_2^{obs} - \frac{N_2(N_2+1)}{2}$

(5) $W_1 + \ W_2 = \ \ R_1^{obs} - \frac{N_1(N_1+1)}{2} + \ \ R_2^{obs} - \frac{N_2(N_2+1)}{2}$

(6) By taking into account that $R_1^{obs} + \ \ R_2^{obs} \ = N(N \ + \ 1)/2 \ and \ N \ = N_1 \ + N_2$, then the sum is $W_1 + \ W_2 = \ N_1 N_2$

(7) The final value of W is taken as the maximum between W_1 and W_2, $W = max(W_1, W_2)$.

(8) High values of W shows the most different value between two samples. Therefore, the N first variables with W value are selected

Feature selection has been successfully applied to clinical datasets e.g., lymphoma, gene expression, cancers (Liu *et al.*, 2002; Li *et al.*, 2006; Wang *et al.*, 2007b; Qi and Li, 2009). Aha (Aha and Bankert, 1996) claimed that feature selection consistently increased accuracy, reduced feature set size, and provided better accuracy of classification. Liu (Liu *et al.*, 2002) said feature selection played an important role in classification and is effective in enhancing learning efficiency, increasing productive accuracy and reducing complexity of learning. Results reveal that learning can be achieved more efficiently and effectively with just relevant and non-redundant features.

4.5 Summary

This chapter has discussed principles for dimensionality reduction techniques, including feature extraction and feature selection techniques. There are useful for reducing the number of dimensions in high dimensionality datasets. Feature extraction is the process of transforming the original feature set into novel feature that are weighted combinations of the original features. It is applied to reduce the number of feature dimensions, so it is not useful for applications that need to use the meaningful labels of the features. Feature selection is a process of selecting a subset of the original features according to certain criteria. It selects meaningful features, which can be used in predictive modelling.

CHAPTER 5

METHODS FOR CLASSIFICATION

5.1 Introduction

In Chapter 2 a framework, for mining clinical datasets, was developed and discussed. In Chapter 4 one of the key processing steps, namely that of reducing the dimensionality was analysed and discussed. In this chapter, issues associated with classification are discussed. It should be kept in mind that in Real Live Clinical Datasets, there will always be imbalance, i.e. there will more live patients than dead patients. Therefore, it is important to discuss classification methods in the presence of class imbalance. According to proposed framework (Figure 5.1), classification is used to assess the pre-processed data (after handling data issues). In order to evaluate the behaviour of classifiers by relating their performance to the nature of data, pre-processing processes (imputation, resampling and feature selection) are used. In this thesis, classification is also used to compare the proposed feature selection scheme against some of the well-known algorithms.

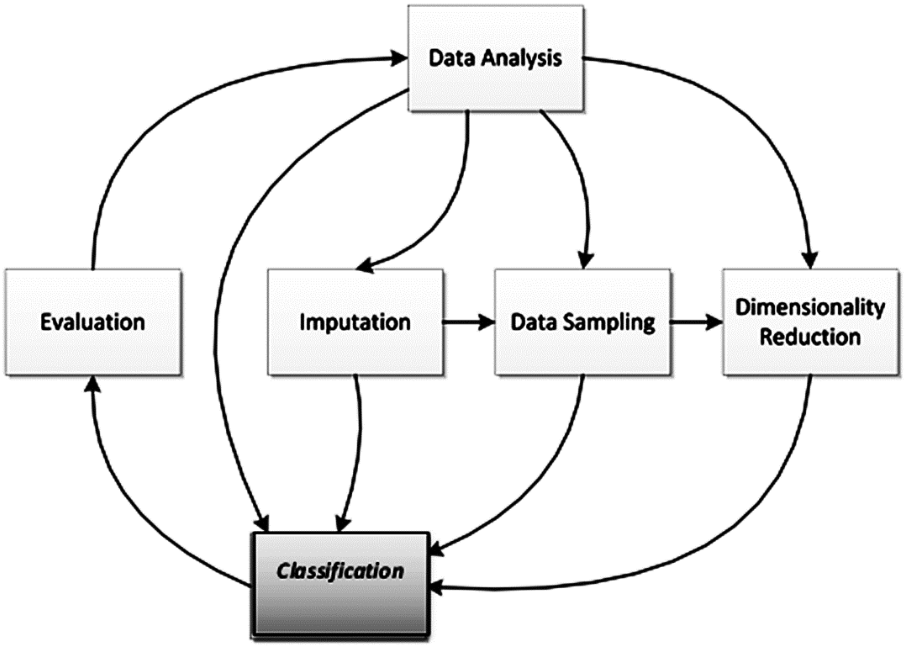


Figure 5.1: Classification in Handling Clinical Data Framework (HCDF)

The methods for classification are discussed in this chapter. Section 5.2 explains the definition of classification and gives the examples of classifiers that are applied in this thesis. Section 5.3 demonstrates the problem of imbalanced classes and presents two strategies, over-sampling and under-sampling, to handle this issue. Section 5.4 explains the normalisation methods and the evaluation of classification is discussed in section 5.5.

5.2 Classification

Classification is supervised learning and it defines the class of outputs. Of course, if target classes are not provided in the training set, unsupervised learning methods like clustering could be used for this purpose (Han et al., 2012). Data classification is a two-step process. In the first step a model is built describing a predetermined set of data classes or concepts. The model is constructed by analysing data samples (records) described by attributes, and each tuple belongs to a predefined class. In the context of classification, data tuples are also referred to as samples, examples or objects. The individual tuples making up the training set are referred to as training samples and are randomly selected from the sample population. Since the class label of each training sample is provided, supervised learning (i.e., the learning of the model is "supervised" in that it is told to which class each training sample belongs) is ideal. In the second step, the model is used for classification.

In the heart failure clinical dataset (LIFELAB) for this thesis, the two classes of patients who are alive and those who have dead are provided. In what follows, a selection of classification methods is surveyed, along with methods for dealing with class imbalances.

5.2.1 Multilayer Perceptron (MLP)

In Chapter 4, it was shown that a multi-layered neural network could be both structured and trained to provide a set of reduced features. In this chapter, a multilayer perceptron (MLP) will be used to learn the classification problem . In a multilayer perceptron (Gardner and Dorling, 1998; Autio *et al.*, 2007; Suzuki, 2011), the architecture is such that information flows from one layer to the next, and no information is passed on within a layer. Here, the first layer is considered as the input layer, and has as many inputs as there are variables (attributes). This information is then weighted and passed onto all the nodes in the next layer. After processing the information, each layer then passes on its outputs to the next layers till the output is obtained. Thus the design is based around a decision on the number of nodes in the intermediate layers and the number of intermediate layers (often called hidden layers and nodes). All the weights, connecting the nodes, are randomly initialized to a number and then updated by a back-propagation algorithm. A back-propagation algorithm is a typical learning algorithm which is used to train these networks. It consists of two phases, namely, a forward phase and a backward phase. In the forward phase the output value of each node is computed, layer by layer, using a weighted sum of its inputs. In the backward phase, the weights are updated after the error in the prediction and the target is evaluated, using a gradient descent algorithm. The algorithm essentially minimises the squared error between the network values and the target values.

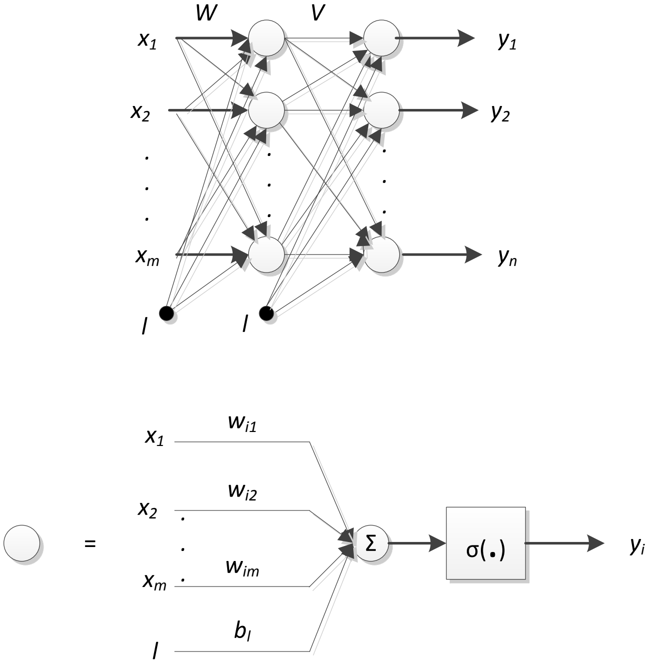


Figure 5. 2: A multilayer perceptron structure

Fig. 5.2 illustrates the architecture of multilayer perceptron. It shows the output y, which is a vector composed of n components. The n components are determined in terms of m components of an input vector x. The hidden layer is composed of l components. The mathematical representation is expressed as:

$$y_i(x) = \sum_{j=1}^{l} \left[v_{ij} g \left(\sum_{k=1}^{m} w_{ij} x_k + b_{wj} \right) + b_{vi} \right]; i = 1, \dots, n \tag{5.1}$$

where v_{ij} and w_{ij} are synaptic weights, x_k is the kth element of the input vector, $g(\cdot)$ is an activation function and b_{wj}, b_{vi} are the bias. The bias has the effect of increasing or decreasing the net input of the activation function depending on whether it is positive or negative, respectively.

5.2.2 Radial Basis Function Network (RBFN)

A radial basis function network (RBFN) (Suzuki, 2011) is an artificial neural network model that uses a RBF as an activation function. Fig. 5.3 presents the architecture of RBFN. It is composed of three layers: an input layer, a hidden layer and an output layer. Each hidden unit implements a radial activation function (a non-linear transfer function) and each output unit implements a weighted sum of hidden unit outputs.

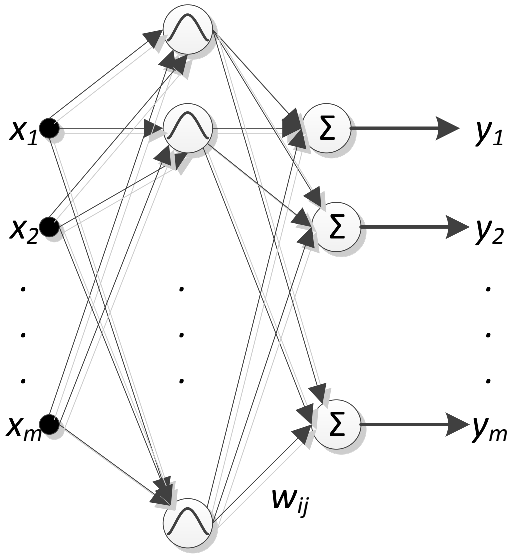


Figure 5. 3: A radial basis function network architecture

The output of ith neuron in the output layer of the RBF network is determined as shown in Eq. 5.2 below.

$$y_i(x) = \sum_{j=1}^{M} w_{ij}\varphi(\|x - c_j\|) \quad ; \; i = 1, \dots, m \tag{5.2}$$

where $\varphi(.)$ is the basis function, c_j is the centre vector for hidden neuron j and w_{ij} is the weight between the node j of the hidden layer and the node i of the output layer, and m is the number of nodes in the output layer. The norm is typically taken to be the Euclidean distance and the basic function is taken to be Gaussian:

$$\varphi(\|x - c_j\|) = exp\left\{\frac{\|x - c_j\|^2}{2\sigma_j^2}\right\} \tag{5.3}$$

where σ_j is the width parameter of the jth hidden unit in the hidden layer.

5.2.3 Support Vector Machine (SVM)

Support vector machines (SVMs) (Cortes and Vapnik, 1995) are supervised learning models. SVM's are essentially non-probabilistic binary linear classifiers, which use a representation of the key example points, which are mapped so that separate categories are divided by a gap that is as wide as possible. New data points are then mapped onto the same space and a prediction is made depending on which side of the divide they fall.

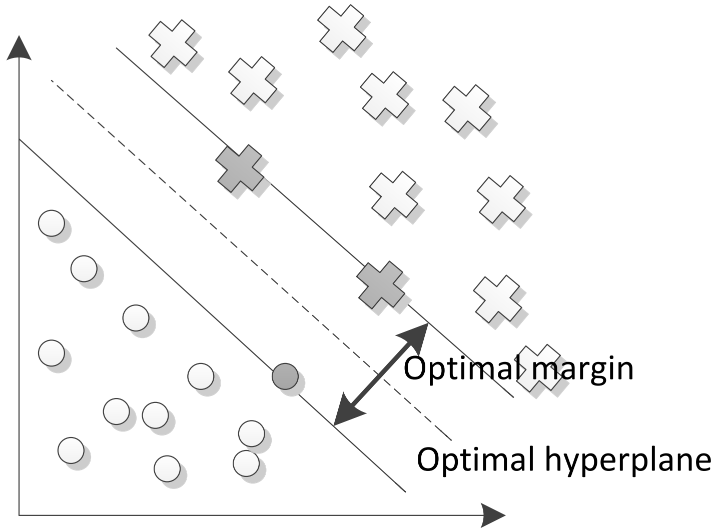


Figure 5. 4: A separable problem in a two dimensional space
(Cortes and Vapnik, 1995)

The learning in an SVM is the construction of a hyperplane which is used for classification. An ideal or optimal hyperplane can be defined as a linear decision function which provides the maximal margin between the vectors of the two classes (see Fig. 5.4). The support vectors define the margin of largest separation between the two classes. SVMs are a popular classification tool as they have excellent generalization properties. However, the training is slow and the algorithms are numerically complex (Platt, 1999). This thesis uses the SVM algorithm called sequential minimal optimization or SMO (Hastie and Tibshirani, 1998; Platt, 1999).

5.2.4 Decision Tree (DT)

Decision trees are popular and powerful methods to classify cases as well as to predict values (David and Balakrishnan, 2010). Their attractiveness is mainly due to the fact that the basic principles of tree-based methods as well as their outcomes are easy to understand. The basic principle is the hierarchical division of all observations into subcategories in such a way that the resulting subcategories differ from each other as much as possible, while the subcategories themselves are as homogenous as

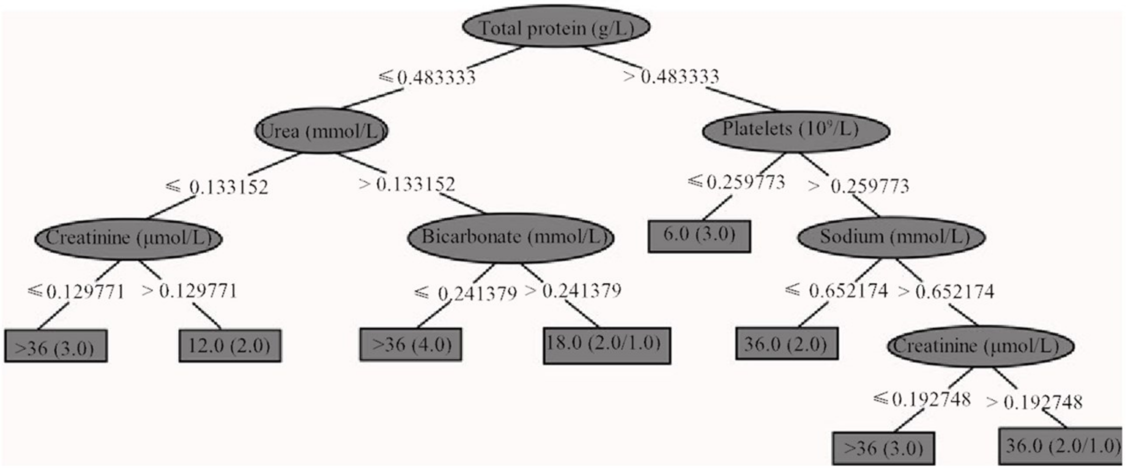


Figure 5.5: A typical example that shows the input and output of the decision tree

possible. The outcome of a decision tree is a rule that can be expressed in plain English as well as implemented as a database query in order to quickly and repeatedly apply it to new data.

Decision trees use a supervised learning technique, in which the input features are partitioned into regions, and where each assigned label is a value or an action to characterize its data points (Fig. 5.5). In this thesis, a decision tree obtained using the C4.5 algorithm is generated for classification. C4.5 is an algorithm developed by Ross Quinlan that generates Decision Trees (DT), which can be used for classification problems (Quinlan, 1996). It improves (extends) the Iterative Dichotomiser 3 (ID3) algorithm (Quinlan, 1986) by dealing with both continuous and discrete attributes, missing values and pruning trees after construction.

C4.5 builds decision trees from a set of training data in the same way as ID3, using the concept of information entropy. The training data is a set $S = s_1, s_2, \dots$ of already classified samples. Each sample $X = x_1, x_2, \dots$ is a vector where $x_1, x_2, \dots$ represent attributes or features of the sample. The training data is augmented with a vector $Y = y_1, y_2, \dots$ where $y_1, y_2, \dots$ represent the class to which each sample belongs. At each node of the tree, C4.5 chooses one attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. Its criterion is the normalized information gain (difference in entropy) that results from choosing an attribute for splitting the data. The attribute with the highest normalised information gain is chosen to make the decision. The C4.5 algorithm then recurses on the smaller sublists.

Algorithm 5.1: C4.5 algorithm to build decision tree

Choose attribute for root node

Create branch for each value of that attribute

Split cases according to branches

Repeat process for each branch until all cases in the branch have the same class

Choosing which attribute to be a root is based on highest gain of each attribute.

$$Gain(X,F) = Entropy(x_j) - \sum_{j=1}^{m} \frac{|x_j|}{|X|} \times Entropy(x_j)$$

where:

$x_1, \ldots, x_j, \ldots, x_m$ = partitions of X according to values of attribute

m = number of attributes

$|x_i|$ = number of cases in the partition x_i

$|X|$ = total number of cases in X

$$Entropy(X) = \sum_{i=1}^{n} -pi \times \log_2 pi$$

where:

X : Dataset

n : number of cases in the partition X

pi : Proportion of x_i to X

5.2.5 Random Forest (RF)

Random forest, as the name suggests, is a collection of trees: decision trees, in this case. An algorithm for classification using a random forest approach was developed by Leo Breiman (Breiman, 1996; Breiman, 2001). Here, a combination of tree predictors is used, such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The input class of the random forest for a given input is the mode of the classes predicted by individual trees.

Random Forests is a classification algorithm with a simple structure (Breiman, 2004). We examine the performance of this procedure when there are F_s strong variables and F_w weak ones. Choose *mtry* variables from among the *M* at random with replacement--then choose the one that gives the best split. Assume all strong variables give equal results on the split; the same for all weak variables. If the selection is all weak, then choose one at random to split on. If there is more than one strong variable selected in *mtry*, select one at random to split on. A forest of trees is grown as follows:

1) The training set is a bootstrap sample from the original training set.

2) An integer *mtry* is set by the user, where *m mtry* is less than the total number of variables. At each node, *mtry* variables are selected at random and the node is split on the best split among the selected *mtry*. The tree is grown to its maximal depth.

3) In regression, as a test vector *x* is put down each tree it is assigned the average values of the y-values at the node it stops at. The average of these overall

trees in the forest is the predicted value for x. The predicted value for classification is the class getting the majority of the forest votes.

At the optimal $mtry$, assume F_w is large compared to F_s, then

$$mtry \approx M/F_s(1 + (4/3)F_s) \qquad (5.4)$$

Unlike single trees, where consistency is proved by letting the number of cases in each terminal node become large (Breiman, 2001) RF trees are built to have a small number of cases in each terminal node.

5.3 Class imbalances

Learning classification methods generally perform poorly in the presences of imbalanced data. This is because learning classifiers attempt to reduce global quantities such as the error rate, and do not take the data distribution into consideration. As a result, examples from the dominant class are well-classified whereas examples from the minority class tend to be misclassified. Thus, failure to properly represent the distributive characteristics of the data results in inaccuracies across the classes of the data. This implies that either the learning classification algorithms are modified or the data presented to them is modified.

The current understanding of the imbalanced learning problem is that the number of records belonging to one class is much more or much less than that of all the other classes. Most machine learning algorithms are trained based on the assumption that the ratios of each class are almost equal and thus the errors associated with each class have the same cost. Since the cost gets skewed in favour of the majority class, learning classifiers are often biased towards them.

5.3.1 Sampling data

Building a classification model with imbalanced dataset will cause the underrepresented class to be overlooked or even ignored. One way to correct the imbalance is to train a cost sensitive classifier with the misclassification cost of the minority class greater than that of the majority class. There are two techniques for sampling data: (a) oversampling and (b) under-sampling. These are discussed in the following sections.

5.3.2.1 Over-sampling strategy

Synthetic Minority Over-sampling Technique (SMOTE), developed by Chawla, Hall, & Kegelmeyer in 2002 (Chawla *et al.*, 2002), is an over-sampling technique whereby synthetic minority examples are generated. It combines informed over-sampling of the minority class with random under-sampling of the majority class. Using the over-sampling approach the minority class is over-sampled by creating artificial examples of k nearest class neighbours as seen in figure 5.5. SMOTE currently yields the best results for re-sampling and modifying the probabilistic estimate techniques (Chawla *et al.*, 2002). This technique is a popular one, which creates artificial samples to increase the size of minority class. It balances the data by increasing the number of minority instances by over-sampling them. SMOTE generates synthetic examples to the minority class; the minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbours.

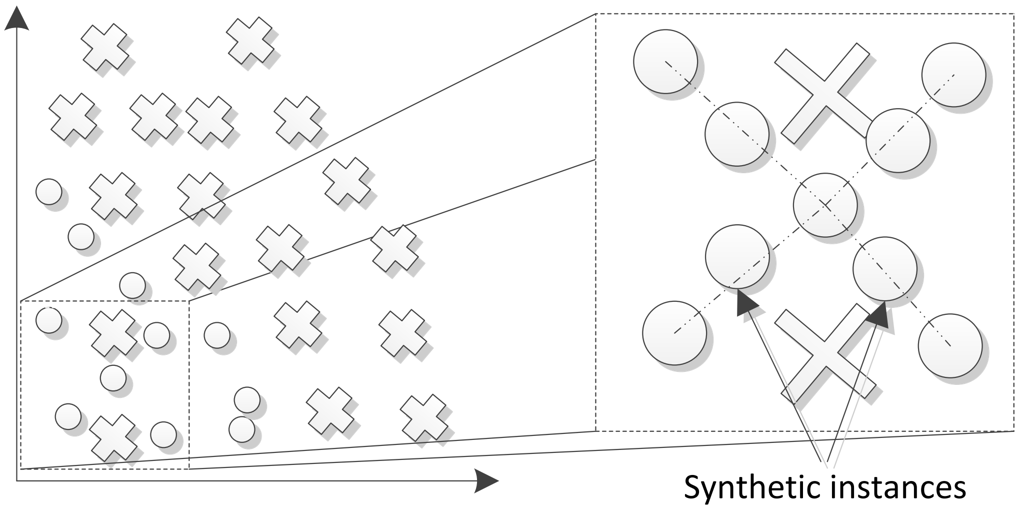


Figure 5.6: SMOTE - synthetic instances

Algorithm 5.2: Algorithm for SMOTE

For each minority sample

Find its *k*-nearest minority neighbours

- Randomly select *q* of these neighbours
- Randomly generate synthetic samples along the lines joining the minority sample and its *q* selected neighbours
- (*q* depends on the amount of oversampling desired)

5.3.2.2 Under-sampling strategy

Another strategy of sampling data is under-sampling that reduces the set of data examples (in this thesis means number of patients). The purpose of balancing data by using under-sampling is to achieve a high performance of classification and avoid the bias towards majority class examples (Garcia and Herrera, 2009). One simple method for under-sampling data is to select a subset of majority class samples randomly (Yen and Lee, 2006; Yan-ping *et al.*, 2010). However, many researchers proposed different methods to select the samples from majority class for example, Near-miss methods (Zhang and Mani, 2003), Cluster based method (Altınçay and Ergün, 2004; Yen and Lee, 2006; Rahman and Davis, 2013), and Distances between samples (Yen and Lee, 2006).

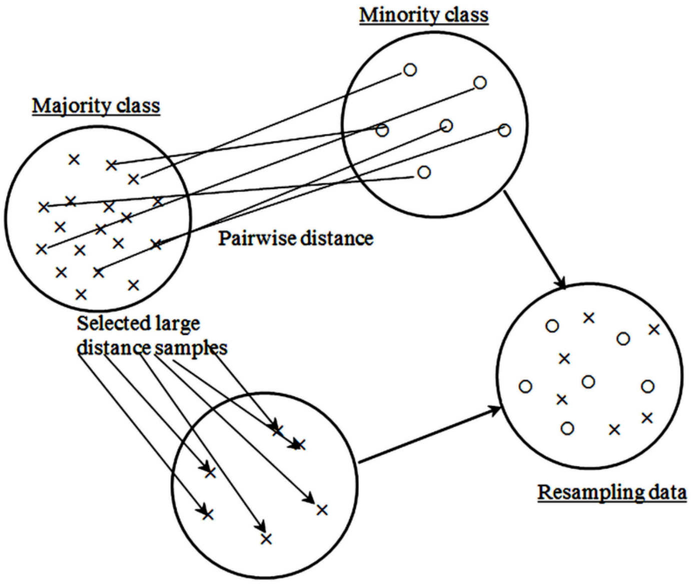


Figure 5.7: Under-sampling method

In this thesis, distance-based random under-sampling is proposed and used to compare the performance of classification between over-sampling and under-sampling. The majority data is selected by using the pairwise distance; Euclidian distance is used in this thesis but for other distances can also be applied. This strategy also uses the similarity between the minority class and majority class to find the greatest distance between them for selecting the instance from majority data to be balanced with minority data (Fig. 5.7).

Algorithm 5.3: Algorithm for distance-based random under-sampling

For sample data in majority class

- Apply Euclidian distance for the samples of majority and minority
- Select the samples by finding the largest distance between minority (D_i) and majority (A_i)

$$Large\ Dist\ (D_i, A_i)$$

- Randomly select data sample from majority class that tend to be balanced with minority data

According to the distance-based random under-sampling that shown in Algorithm 5.3, it selects the samples that belong to majority class by using the distance between samples that belong in different class. The balancing data from this method can reduce the bias from majority class and provide the appropriate dataset for data analysis. However, the relationship between training set size and improper classification performance for imbalanced data sets seems to be that on small imbalanced data sets the minority class is poorly represented by an excessively reduced number of examples that might not be sufficient for learning. For larger data sets, the effect of these complicating factors seems to be reduced, as the minority class is better represented by a larger number of examples.

5.4 Evaluation

Performance evaluation is probably the most critical of all the steps in the data mining process that have shown in framework (Fig. 5.2). Commonly, the accuracy is assessed to evaluate the performance of classification. In classification, supervised learner models are designed to classify, estimate, and/or predict future outcome. For some applications the desire is to build models showing consistently high predictive accuracy. Classification correctness is best calculated by using unseen data in the form of a test set to evaluate the model. Test set model accuracy can be summarized in a table known as a confusion matrix (Table 5.1).

Table 5.1: Confusion matrix

Performance Measure		Predict	
		Positive	Negative
Actual	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

The confusion matrix can be used to identify the performance of classification. The Positive Predictive Value (PPV) and the Negative Predictive Value (NPV) are both measures of the accuracy. True positive (TP) is equivalent to a hit. True negative (TN) is equivalent to correct rejection. False positive (FP) is equivalent to a false alarm. False negative (FN) is equivalent to a miss. These are used to calculate the positive predictive value, negative predictive value, sensitivity and specificity as seen by the equation below. The PPV is equivalent to precision. In using population-based data for risk factor analyses it is important that identified cases are true cases (high PPV) (Ford et al., 2007).

$$PPV\ (precision) = \frac{TP}{(TP + FP)} \tag{5.4}$$

$$NPV = \frac{TN}{(FN + TN)} \tag{5.5}$$

$$Sensitivity\ (recall) = \frac{TP}{(TP + FN)} \tag{5.6}$$

$$Specificity = \frac{TN}{(FP + TN)} \tag{5.7}$$

$$Accuracy = \frac{TP + TN}{(TP + FP + FN + TN)} \tag{5.8}$$

The confusion matrix in Table 5.1 shows that TP and TN denote the number of positive and negative examples that are classified correctly. FN and FP denote the number of misclassified positive and negative examples respectively.

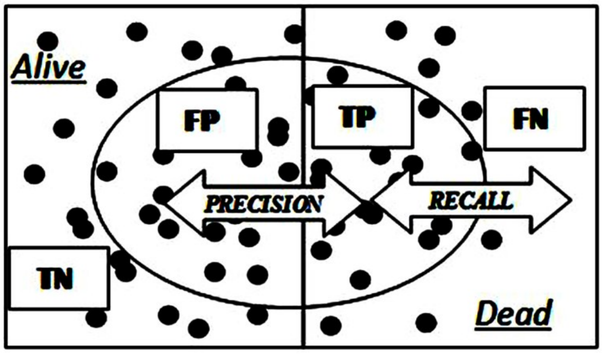


Figure 5. 8: The performance indicators on target class

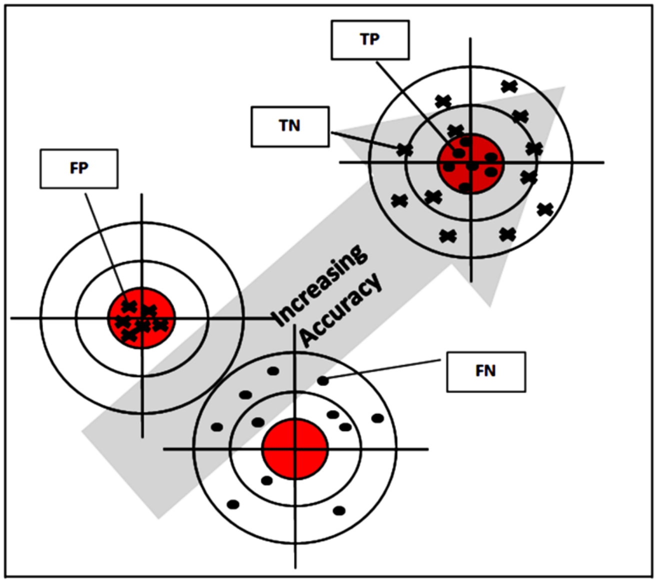


Figure 5.9: Relationship of performance indicators

Any single performance indicator suffers the risk of not being suitable; Fig. 5.8 and Fig. 5.9 show that the relationship of performance indicators.

Precision is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved. It is usually expressed as a percentage.

Recall is the ratio of the number of relevant records retrieved to the total number of relevant records in the database. It is usually expressed as a percentage.

Thus, it should be more carefully used a confusion matrix to investigate and evaluate the performance of the classification.

Table 5.2: The classification accuracy of different classifications using the 'Original' LIFELAB dataset

		Class	Original	
			Accuracy	
			Precision	Recall
Classification	MLP	Dead	46.5	41.4
		Alive	81.2	84.2
	RBFN	Dead	56.4	28
		Alive	79.5	92.8
	SVM	Dead	68.4	32.6
		Alive	80.9	95
	DT	Dead	43.4	36.1
		Alive	79.9	84.4
	RF	Dead	61.2	23.1
		Alive	78.8	95.1

Table 5.2 is an example to demonstrate the results on the different classifiers; it shows the accuracy of the different classifiers when used on the 'Original' (unpre-processing data) LIFELAB dataset, which consists of 60 variables, and contains missing values and imbalanced classes. In this thesis precision and recall are measured the effective of classification. In medical diagnosis, the default assumption of equal misclassification costs underlying machine learning techniques is most likely violated. Precision is important that identified cases are true cases (high precision). A false negative prediction that is used for recall may have more serious consequences than a false positive prediction (Yang *et al.*, 2009). For example, consider prediction task, where we are predicting for patient who has a high probability of dead. Suppose that we are given a list of patients to classify as "relevant" or "non-relevant" for dead case, and then the cost of mistakenly assigning a relevant patient to the non-relevant patient class depends on whether there are any other relevant patents that we have

correctly classified. Recall tends to be neglected or averaged away in machine learning and computational linguistics where the focus is on how confident we can be in the rule or classifier (Powers, 2007). Consequently, in this paper both precision and recall are evaluated. From the results, it can be seen that as a classifier on the dead class (positive class), the SVM algorithm gives a better precision than other classifiers, at 68.4%, and the MLP gives a better recall than others at 41.4%. However, these results are assessed without pre-processing. The results with pre-processing methods will be presented in Chapter 7.

5.5 Summary

In this thesis, classification accuracy was selected as the criteria to assess the effectiveness of the data mining methods. The classifier used were: multilayer perceptron (back-propagation), J48 (decision tree), RBFN (neural network), SVM and Random Forest. However, the fundamental factor here is to understand the nature of the dataset in order to choose a suitable technique. In addition, imbalanced class is an issue that does occur naturally in clinical datasets. Resampling of data sampling is one way to deal with this problem and is essentially a process, which enables the balancing of the proportions of majority and minority class in a dataset, such that they both have similar sizes in terms of number of samples in each class. A sampling strategy, which is applied, has to be such that reliable results are obtained, and is also, statistically representative of the full detail data. A key reason for this resampling is that most data mining and classification algorithms often show a strong bias towards the majority class, and for purposes of clinical applications a goal is to minimise the overall prediction error rate especially the minority class (positive case) are minimized. It should be noted that the size of samples for each class should be big enough to contain the significant information whether or be not too small to represent the data.

CHAPTER 6

FEATURE SELECTION BY PROJECTING ONTO PRINCIPAL COMPONENT

6.1 Introduction

The problem of high dimensionality in datasets was discussed in Chapters 1 and 4. High dimensionality results from the ability to gather data with a large number of variables, often without knowing whether they are suitable, required or significant. Number of features has an effect on the complexity of the algorithms that can be developed and used within any framework for data mining. In this chapter, a new methodology for reducing dimensions is proposed. This methodology combines elements from both feature extraction and selection, and is essentially part of feature selection. It is a filter model combined with a non-linear feature extraction. The advantage of this method is the retention of feature labels (gain in interpretability) while maintaining performance through the combining of the strengths of feature extraction and feature selection. This allows for accuracy in the data mining process to be maintained, while at the same time also keeping the computational overheads low.

Real-world data, such as speech signals, digital photographs, bioinformatics, multimedia, economic and consumer transactions usually have high dimensionality (Fodor, 2002; Cunningham, 2008; van der Maaten *et al.*, 2009). One of the key issues with high dimensional datasets is that, in many cases, not all the measured variables are "important" for understanding the underlying phenomenon of interest (Fodor, 2002). Thus, dimensions can be eliminated, thereby reducing complexity while retaining the performance of data mining algorithms. Essentially, dimension reduction is a process of projecting a high dimensional data space onto a space with fewer dimensions (see Fig. 6.1, repeated from Chapter 4).

As discussed in earlier chapters (Chapters 1 and 4), techniques for reduction of dimension can be classified into two broad categories (a) feature extraction and (b) feature selection. In the main, feature extraction can give high accuracy for both classification and prediction problems. However, the reduced dimensionality of the data set is essentially one which yields new (and fewer) dimensions than before. These new dimensions do not necessarily carry any meaning, nor can they be directly associated with the variables of the dataset. On the other hand, feature selection, also

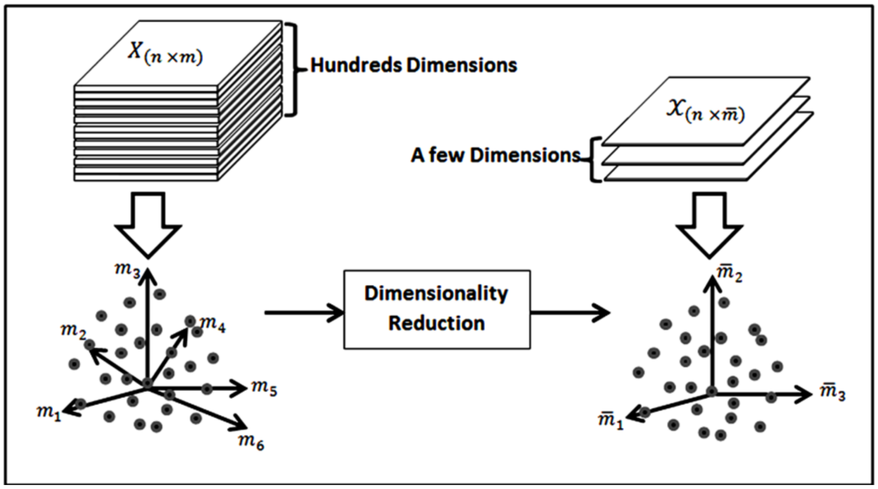


Figure 6.1: Dimensionality reduction from high dimensionality to low dimensionality

reduces the dimensions, and retains the labels associated with the variables; in a sense the new set of features is a subset of the original set of features. Both the categories of techniques are used frequently, for example in image processing feature extraction is a popular technique, while where it is important that labels are retained for the features (e.g. clinical systems) feature selection is the dominant technique for reduction of dimensions.

The key performance indicators of a data mining process are predictive accuracy, speed of the data mining algorithm, and ability to provide a greater insight into both the data and the application in order to develop good decision support systems. These indicators become even more important when dealing with clinical data, where such systems are often used to help support decision making and diagnosis at a lower level and also to support tele-health/tele-medicine systems (Sittig *et al.*, 2008; Fox *et al.*, 2010). For example in clinical systems, one of the key uses of prediction is to be able to both predict hospitalisation of patients, and develop systems for planned care of patients, thus enabling better control over costs and greater efficiency. The performance of prediction (or the accuracy of prediction) is highly dependent on the ability to select the most appropriate or relevant variables (dimensions) from a list of variables available in the dataset (Ramaswami and Bhaskaran, 2009). The most important question is which of the features is the most influential in determining the classification and hence should be chosen first. This thesis develops a new methodology for the selection of the right variables through the use of efficient dimension reduction for datasets with a large number of variables.

The proposed methodology combines both feature extraction and selection. In this new methodology, the advantages of feature extraction and feature selection are retained and combined. The proposed method, called Feature Selection by Projecting onto Principal Component (FS-PPC) is an integrated filter combined with a non-linear feature extraction. This allows for accuracy in the data mining process to be maintained, while at the same time also keeping the computational overheads low. The FS-PPC has three components

- Principal component generation: Here principal components are generated by implementing the PCA (Kramer, 1991; Scholz, 2012) to determine the most significant principal components.
- Features subset evaluation process: Here each candidate subset of features is evaluated using a Symmetrical Uncertainty (SU) measure.
- Subset selection criterion: Here irrelevant and redundant features are removed from within the candidate set of features using Mutual Information (MI).

This is an iterative process, and is carried out until all the features in the subset have been ranked, with the elimination of low-ranking features from the subset of features. If a subset turns out to be better, it replaces the current best one (a small example is used in this chapter to illustrate the method, whilst a more detailed set of results will be presented in Chapter 7).

The chapter is structured as follows: section 6.2 outlines the preliminaries by providing some definitions, background and notations, which are then used later in the chapter. Section 6.3 describes the steps of the proposed feature selection method, which has all the components of the methods. Finally, in section 6.4 we draw some concluding remarks, and summarize the methods.

6.2 Background and notation

Consider a dataset $X = \{x_{i,j}\}, i = 1, 2, 3, \dots, n\,;\ j = 1, 2, 3, \dots, m$. We can define $X_i = \{x_{i,j}\}$ consisting of a set of n observations (records) of data while $F_j = \{x_j\} \in F$ is the m set of variables (attributes) and also introduce a class variable $Y = \{y_i\}$ as output. Collecting the variables together the data set can be represented as $X(F, Y)$. It should be noted here that the term input variables is used to represent features presented to the algorithms, whilst observations is used to represent the data that is collected.

6.2.1 Dimensionality reduction

The problem can be stated as follows: given the m dimensional variable $X = (X_1, X_2, X_3, \dots, X_m)^T$, find a representation of it with fewer dimensions, i.e. $\Phi = (\Phi_1, \Phi_2, \Phi_3, \dots, \Phi_p)^T$ with $p \leq m$, that captures the content in the original data. The components of Φ are mostly called “variables” or “features” or “attributes” in computer science and machine learning literature. Dimensionality reduction is categorised into the following two techniques, feature extraction and feature selection.

6.2.2 Feature extraction

The goal of feature extraction is to yield a new feature set with fewer dimensions by defining a mapping function $M(\cdot)$ such that $\Phi = M(F_j)$

In general, the map $M_F(\cdot)$ could be any function, linear or non-linear. The result of applying $M_F(\cdot)$ is to create $\Phi = \{F_j\}$, $(j = 1, 2, 3, \dots, p)$, where p is the number of a new feature of fewer dimensions ($p \leq m$).

The classification task of learning algorithms is introduced by $h_x : \Phi \rightarrow Y$, $\varepsilon_\Phi(h_x) = |\{h_x(\Phi) \neq y_i\}|/n$ for each classifier, where $h_x(\Phi)$ is the predicted class label of y_i by h_x. $\varepsilon_\Phi(h_x)$ is a classification error from Φ that is under control at a range ε, i.e., $|\varepsilon_F(h_x) - \varepsilon_\Phi(h_x)| \leq \varepsilon$. It shows that the loss information, ε is measured by resulting error between an original subset of features $\varepsilon_F(h_x)$ and a new subset of features $\varepsilon_\Phi(h_x)$. By understanding this, it can be concluded that a smaller error can be achieved by reducing the new subset of feature errors.

6.2.3 Feature selection

It should be noted that in feature extraction the labels associated with the final set of features are meaningless. However, feature selection allows for the retention of labels of the variables. Given a criterion $M_F(S)$, and a set of features F, the problem is to find a set S subset of F where $S \in F$.

Since the dataset has limitations, in that all instances of the relationship between observations and class cannot be collected, it is expected that there will be an error associated with the classification using the reduced set of features.$\varepsilon_S(h_s) =$

$|\{h_s(S) \neq y_i\}|/n$ for each classifier, where $h_s(S)$ is the predicted class label of y_i by h_s. $\varepsilon_S(h_s)$ is a classification error from a feature subset S of h that should be a small error.

6.2.4 Relevance and redundancy

Let P denote the conditional probability of the class label Y given a feature set. The statistical relevance and redundancy of a feature can be defined as:

Definition 6.1: Relevance

A feature F_j is relevant *if*

$\exists\, S_j \subseteq S,$ such that $P\left(Y\middle|F_j,\, S_j\right) \neq P(Y|S_j)$

Otherwise, the feature F_j *is said to be irrelevant.*

Definition 6.2: Redundancy

A feature F_j is redundant *if*

$$P\left(Y\middle|F_j, S_j\right) = P\left(Y\middle|S_j\right), \text{but } \exists\, S_j \subseteq S, \text{such that } P\left(Y\middle|F_i, S_j\right) \neq P(Y|S_j)$$

Otherwise, the feature F_j *is said to be non-redundant.*

6.2.5 Mutual information

Mutual information (or Information Gain (IG)) is used to quantify how much information is shared by two variables X and Y, $MI(X;Y)$ is defined as

$$MI(X;Y) = \sum_{y}\sum_{x} p(x,y) log \frac{p(x,y)}{p(x)p(y)} \tag{6.1}$$

From this definition, the value of $MI(X;Y) > 0$, if X and Y are closely related to each other; otherwise, $MI(X;Y)$= 0 denotes that these two variables are totally unrelated. Additionally, $MI(X;Y)$ can be rewritten as

$$MI(X;Y) = H(X) - H(X|Y) \tag{6.2}$$

Conditional mutual information of X and Y, denoted as

$$MI(X;Y|Z) = H(X|Z) - H(X|Y,Z) \tag{6.3}$$

represents the quantity of information shared by X and Y when Z is known. That is to say, $MI(X;Y|Z)$ implies Y brings information about X which is not already contained in Z.

6.2.6 Symmetrical uncertainty

The Symmetrical Uncertainty (SU) coefficient normalises the information gain by dividing the sum of the entropies of X and Y. Both the gain ratio and the symmetrical uncertainty coefficient lie between 0 and 1. A value of 0 indicates that X and Y have no association; the value 1 for the gain ratio indicates that knowledge of Y completely predicts X; the value 1 for the symmetrical uncertainty coefficient indicates that knowledge of one variable completely predicts the other. Both display a bias in favour of attributes with fewer values.

$$SU = 2 \times \left[\frac{IG}{H(Y) + H(X)}\right] \tag{6.4}$$

Eq. (6.5-6.8) are used to find the MI and SU. If X and Y are discrete random variables, equations 6.5 and 6.6 give the entropy of Y before and after observing X.

$$H(Y) = \sum_{y} p(y) log_2(p(y)) \tag{6.5}$$

$$H(Y|X) = \sum_{x} p(x) \sum_{y} p(y|x) log_2\big(p(y|x)\big) \tag{6.6}$$

Eq. 6.7 gives the amount of information gained about Y after observing X (and vice versa—the amount of information gained about X after observing Y). Information gain (IG) within the framework of data mining and in feature selection provides very valuable information, however, it should be noted that it is biased in favour of attributes with more values; in other words, attributes with a low percentage of missing values. Thus, missing value imputations schemes, discussed in Chapter 3, become important. There are three ways of determining information gain (IG):

$$IG = \begin{cases} H(Y) - H(Y|X) \\ H(X) - H(X|Y) \\ H(Y) + H(X) - H(X,Y) \end{cases} \tag{6.7}$$

The gain ratio (Eq. 6.8) is a non-symmetrical measure that tries to compensate for this bias. If Y is the variable to be predicted, then the gain ratio normalises the gain by dividing by the entropy of X.

$$gain\, ratio = \frac{IG}{H(X)} \tag{6.8}$$

6.3 Feature Selection by Projecting onto Principal Components (FS-PPC)

Feature extraction is a useful method for reduction of dimensions; however, it is not useful for the applications that need to use meaningful (and perhaps the original) labels of the features. It has a distinct advantage in that accuracy in classification is high. On the other hand, feature selection is a process of selecting a subset of original features based on a desired criterion. It reduces the number of features, removes variables that are irrelevant and redundant, and also the computational complexity is lower than that of feature extraction. Both these methods for reducing dimensions are extensively used, and the choice is dependent on the application and the problem to be solved. What is important to note is that often feature extraction has a higher degree of accuracy when compared to feature selection (Kotani *et al.*, 1999; Addison *et al.*, 2003; Cruz-Barbosa *et al.*, 2011). Thus, it would be interesting to create a methodology which is a combination of the two methods. In this thesis, we develop a new feature selection algorithm, in which nonlinear principal feature selection is proposed for selecting the significant optimal subset of features. This method combines principal component and feature correlations (symmetrical uncertainty and mutual information), in order to obtain the exact values of symmetrical uncertainty of candidate features and the most significant principal component (1st PC) that it is produced by PCA (or NLPCA). Before identifying a desired feature, the mutual information of candidate features is applied to find irrelevant and remove redundant features.

The proposed methodology consists of the following steps (see Fig. 6.2):

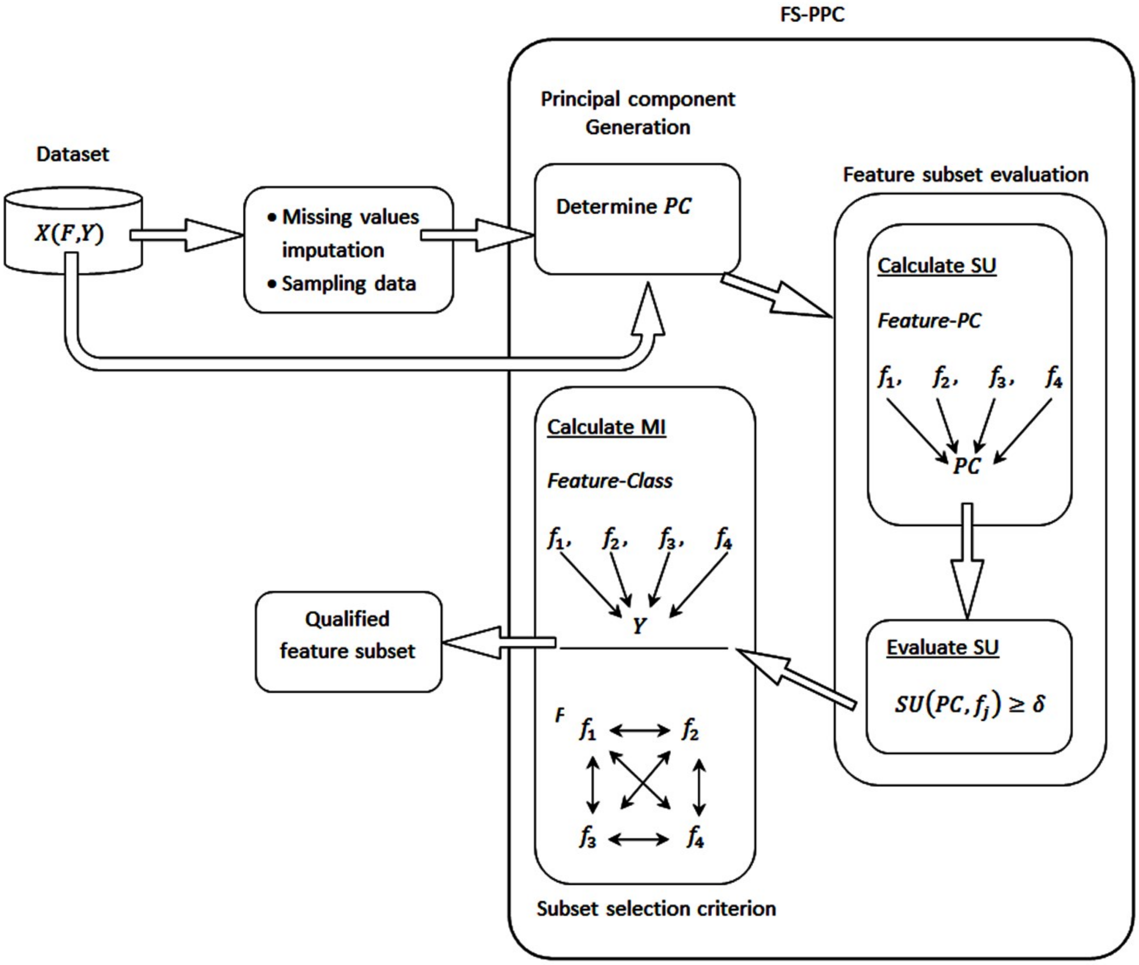


Figure 6.2: The components of FS-PPC

6.3.1 Principal component generation

Principal component generation is used to generate the principal component by using PCA or NLPCA (in the results in Chapter 7, PCA is used to generate PC). PCA is a powerful multivariate data analysis method (for further details in Chapter 4). Its main purpose is to summarise large datasets by removing any redundancy in the data. Principal component (Kramer, 1991; Wang *et al.*, 2007a; Zabiri *et al.*, 2009) is used to identify and remove correlations among problem variables and is an aid in dimensionality reduction, visualisation, and exploratory data analysis. In this process, the first principal component represents the most significant principal component.

6.3.2 Features subset evaluation

Features subset evaluation is a process in which each candidate subset of feature is evaluated by calculating SU (see Section 6.2.5) with the most significant principal component. Here, candidate feature subsets are generated for evaluation based on a SU strategy. Each candidate subset is evaluated by calculating SU with the most significant feature component and compared with the threshold (δ) to find an appropriate feature subset. The subset of features is selected by determining the threshold (δ) of SU, and thus the subset with the highest relation to the most significant principal component.

6.3.3 Subset selection criterion

In general, it is widely recognised that a good subset of features should not only be individually relevant, but also should not be redundant with respect to each other features (Brown, 2009). The selection criterion is used to remove irrelevant and redundant features (Hall and Smith, 1999; Novakovic, 2009; Ali and Shahzad, 2012) from this subset by using MI measures (Yu and Liu, 2003) (see Section 6.2.4). The features are eliminated by a ranking criterion based on MI measures. The process is repeated until all the features in the subset have been ranked, and then all low-ranking features are removed from the subset of features. This evaluation process is critical because if an inappropriate or unsuitable threshold, δ, is selected, the ranking of the features will be affected.

6.3.4 FS-PPC Algorithm

With the discussion and analysis above, we develop a new feature selection algorithm using PCA and feature correlations (symmetrical uncertainty and mutual information), in order to obtain the exact values of symmetrical uncertainty of candidate features and the most significant principal component (1st PC) that it is produced by PCA. Before identifying a desired feature, the mutual information of candidate features is applied to find irrelevant and remove redundant features. Explicitly, the details of our algorithm are shown as Algorithm 6.1.

Algorithm 6.1: Algorithm for Feature Selection by Projecting onto Principal Components (FS-PPC)

```
Input: A training dataset $T = D(F, C)$.
Output: Selected features $S$.

Initialize relative parameters:
        $PC; \delta; S = \emptyset;$
//PC = Significant principal component
// $\delta$ = threshold to find appropriate feature subset
// $S$ = subset of selected features

//Determine significant principal component
                PC = PCA(F);

//Calculate its symmetrical uncertainty
        For each feature $F_j \in F$ do
                $SU(PC, F_j)$ on $F$;
        Until $F = \emptyset$;

//Choose the feature $F_j$ with the highest $SU(PC, F_j)$;
                $SU(PC, F_j) \geq \delta$;
                $S = S \cup \{F_j\}$;

//Rank relevant $(MI(Y, F_j))$ and redundant $(MI(F_i, F_j))$; features $F_j$ from $S$ by
  calculating its mutual information
        For each feature $F_j \in S$ do
              $MI_F = MI(Y, F_j) - \sum_{j=1}^{i-1} MI(F_i, F_j)$
        Until $S = \emptyset$;
```

It can be seen that the algorithm estimates MI for each candidate feature in F with the label Y and for each candidate feature F and a feature will be immediately discarded from F if its SU is more than δ. This procedure will be repeated until there are no more candidate features in F. After that, the feature with the highest MI of feature-class and the lowest MI of feature-feature will be chosen (Fig. 6.1).

Example 6.1 Consider the Heart Disease dataset consisting of 270 observations (instances) and 13 variables (features) with 2 classes of output. In order to demonstrate the Feature Selection by Projecting onto Principal Component (FS-PPC), this example follows the steps of the FS-PPC (as stated above) in order to draw comparisons between the theoretical method and its use in practice.

Table 6.1: Example of the Heart Disease dataset

no.	age	sex	chest pain	blood pressure	cholesterol	blood sugar	ecg	heart rate	angina	oldpeak	ST segment	vessel	thak	class
1.	70	1	4	130	322	0	2	109	0	2.4	2	3	3	presence
2.	67	0	3	115	564	0	2	160	0	1.6	2	0	7	Absence
3.	57	1	2	124	261	0	0	141	0	0.3	1	0	7	presence
4.	64	1	4	128	263	0	0	105	1	0.2	2	1	7	Absence
5.	74	0	2	120	269	0	2	121	1	0.2	1	1	3	Absence
6.	65	1	4	120	177	0	0	140	0	0.4	1	0	7	Absence
7.	56	1	3	130	256	1	2	142	1	0.6	2	1	6	presence
8.	59	1	4	110	239	0	2	142	1	1.2	2	1	7	presence
9.	60	1	4	140	293	0	2	170	0	1.2	2	2	7	presence
10.	63	0	4	150	407	0	2	154	0	4	2	3	7	presence
11.	59	1	4	135	234	0	0	161	0	0.5	2	0	7	Absence
12.	53	1	4	142	226	0	2	111	1	0	1	0	7	Absence
13.	44	1	3	140	235	0	2	180	0	0	1	0	3	Absence
14.	61	1	1	134	234	0	0	145	0	2.6	2	2	3	presence
15.	57	0	4	128	303	0	2	159	0	0	1	1	3	Absence
16.	71	0	4	112	149	0	0	125	0	1.6	2	0	3	Absence
17.	46	1	4	140	311	0	0	120	1	1.8	2	2	7	presence
18.	53	1	4	140	203	1	2	155	1	3.1	3	0	7	presence
19.	64	1	1	110	211	0	2	144	1	1.8	2	0	3	Absence
20.	40	1	1	140	199	0	0	178	1	1.4	1	0	7	Absence

$$F = \left\{ \begin{matrix} age, sex, chest\ pain, blood\ pressure, cholesterol, blood\ sugar, ecg, \\ heart\ rate, angina, oldpeak, ST\ segment, vessel, thak \end{matrix} \right\}$$

$$Y = \{Absence, Presence\}$$

<u>Step 1</u>: Consider the significant principal component

$$PCA(F) = PC$$

After applying $PCA(F)$, the set of principal components, PC is shown as below:

PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12	PC13
1.332	-0.129	-0.020	0.022	-0.133	0.045	0.005	0.015	-0.005	-0.014	0.017	0.007	0.046
-0.279	-0.272	-0.057	-0.034	0.010	-0.024	-0.006	0.028	-0.011	0.041	-0.017	0.010	0.008
-0.331	0.214	-0.060	-0.016	0.021	-0.006	0.019	0.021	-0.021	0.003	-0.003	-0.003	0.001
-0.387	0.106	0.095	-0.005	-0.019	0.003	-0.003	0.015	-0.020	0.019	0.009	-0.004	0.019
1.403	-0.231	0.064	-0.001	0.041	0.017	0.060	0.019	-0.037	0.016	-0.014	-0.007	0.034
-0.336	0.187	-0.029	-0.026	-0.003	0.025	-0.033	0.037	-0.016	-0.005	-0.001	-0.002	0.000
0.007	-0.123	0.024	0.091	0.063	-0.013	0.001	-0.024	-0.005	0.012	0.009	-0.003	0.012
-0.436	-0.125	0.032	-0.033	0.013	0.008	0.000	-0.023	-0.013	0.013	0.005	-0.002	0.007
-0.391	-0.129	-0.069	-0.018	-0.055	0.032	-0.006	0.003	0.016	0.017	0.011	0.004	0.016
-0.301	-0.355	-0.027	-0.001	-0.082	0.035	0.009	0.012	0.017	-0.010	-0.008	0.011	0.034
-0.338	0.152	-0.024	-0.028	-0.027	-0.012	-0.036	0.023	0.007	0.016	0.008	0.001	-0.003
-0.430	-0.080	0.022	-0.045	0.077	0.026	-0.008	0.021	-0.010	-0.009	0.006	-0.007	0.000
1.347	0.009	-0.077	-0.035	0.028	0.019	-0.013	0.002	0.011	0.001	0.006	-0.003	-0.006
1.393	0.182	-0.017	0.033	-0.088	-0.028	0.076	-0.004	0.000	-0.013	0.012	0.001	0.040
1.441	-0.222	-0.016	-0.023	-0.002	0.049	-0.017	0.006	0.002	0.008	-0.012	0.002	0.012
1.482	-0.004	0.052	-0.013	-0.055	-0.021	-0.034	0.023	-0.022	-0.015	-0.006	0.004	0.013
-0.393	0.084	0.097	0.000	-0.034	0.012	0.011	-0.021	0.009	-0.002	0.011	-0.001	0.021
-0.443	-0.169	0.032	0.065	0.049	-0.068	-0.037	-0.021	0.016	-0.013	0.017	0.003	-0.004
1.308	-0.038	0.016	-0.012	0.032	-0.056	0.067	-0.010	-0.025	0.005	0.009	-0.005	0.018
-0.374	0.208	0.028	-0.021	0.104	-0.029	0.075	-0.019	0.007	-0.009	-0.003	-0.010	0.000

Consider a significant principal component, PC

Feature ID	SU>0.25
5	0.926181
8	0.865926
1	0.771938
4	0.731095
10	0.681432
3	0.351744
12	0.324462
11	0.274706
13	0.262114
7	0.231119
9	0.203428
2	0.201875
6	0.139412

Step 2: Calculate Symmetrical Uncertainty between a significant principal component and each feature, $\mathrm{SU}(\mathrm{PC}, F_{\mathrm{j}})$:

Set the threshold, δ = 0.25,

then $S = \{5, 8, 1, 4, 10, 3, 12, 11, 13\}$ or

$$S = \left\{ \begin{matrix} cholesterol, heart\ rate, age, \\ blood\ pressure, oldpeak, chest\ pain, \\ vessel, ST\ segment, thak \end{matrix} \right\}$$

Step 3: Calculate Mutual Information $MI(Y, F_j)$ *and* $MI(F_i, F_j)$ to retain the relevant features and remove the redundant features, features F_j from S by calculating its mutual information. The features are eliminated by a calculating as follows:

$$MI_F = MI\big(Y, F_j\big) - \sum_{j=1}^{i-1} MI(F_i, F_j)$$

Ranking	Feature
1	*blood pressure*
2	*cholesterol*
3	*age*
4	*oldpeak*
5	*chest pain*
6	*ST segment*
7	*heart rate*
8	*vessel*
9	*thak*

The low-ranking features are removed from the subset of features, S. In this example, *vessel* and *thak* should be eliminated, and the optimal subset will be:

no.	age	chest pain	blood pressure	cholesterol	heart rate	oldpeak	ST segment
1.	70	4	130	322	109	2.4	2
2.	67	3	115	564	160	1.6	2
3.	57	2	124	261	141	0.3	1
4.	64	4	128	263	105	0.2	2
5.	74	2	120	269	121	0.2	1
6.	65	4	120	177	140	0.4	1
7.	56	3	130	256	142	0.6	2
8.	59	4	110	239	142	1.2	2
9.	60	4	140	293	170	1.2	2
10.	63	4	150	407	154	4	2
11.	59	4	135	234	161	0.5	2
12.	53	4	142	226	111	0	1
13.	44	3	140	235	180	0	1
14.	61	1	134	234	145	2.6	2
15.	57	4	128	303	159	0	1
16.	71	4	112	149	125	1.6	2
17.	46	4	140	311	120	1.8	2
18.	53	4	140	203	155	3.1	3
19.	64	1	110	211	144	1.8	2
20.	40	1	140	199	178	1.4	1

$$S = \left\{ \begin{matrix} cholesterol, heart\ rate, \\ age, blood\ pressure, \\ oldpeak, chest\ pain, \\ ST\ segment \end{matrix} \right\}$$

6.4 Summary

This thesis has presented a new approach to feature selection for machine learning. The proposed FS-PPC combines correlation features and PCA as an available method for feature selection. The FS-PPC uses principal component's performance and feature correlations to guide its selection of a good subset of features. The relevance chooses the principal component that contains the most information. It uses correlation features to delete the features that contain less relevant information.

CHAPTER 7

RESULTS AND DISCUSSIONS

7.1 Introduction

The underlying theme in this thesis has been the challenges posed by real-life clinical data and the development of methodologies for the extraction of information from this data. These challenges were discussed in Chapter 1, along with a framework (HCDF) in Chapter 2. In the later chapters (3 – 6) algorithms which would be incorporated in the HCDF framework were discussed. These chapters correspond to the main objectives highlighted in Chapter 1. This chapter assesses the methods already described for (a) handling missing values (section 7.3.1), (b) class imbalance (section 7.3.2) and the effect of high dimensionality on classification (Section 7.3.3). The framework for the discussion of the results is based around the following: (a) the use of the original unmodified dataset(s) and (b) in the case of Hull-LIFELAB the additional use of a variable from the Seattle Heart Failure Model (SHFM), which is an expert driven model. When discussing the new feature selection algorithm, the performance is also evaluated based on the results from a set of clinical datasets present in the UCI repository (Blake and Merz, 1998).

7.2 Framework for assessment

The results were obtained using software provided within MATLAB[1] (The MathWorks Inc.), Weka (Witten and Frank, 2005), and KEEL (Alcalá-Fdez *et al.*, 2009; Fernández *et al.*, 2009). Where required additional functionality was provided within the environment. The performance was measured based on how well the algorithms contributed towards the required binary classification task, i.e. the Alive class and the Dead class. As mentioned above, all the algorithms were tested on the data from Hull-LIFELAB (discussed in Chapter 1). When the proposed feature selection algorithm was tested, the tests were carried out on additional clinical datasets obtained from the UCI repository. The performance was measured not only on how well the classification was carried out, but also on a "redundancy measure" (Chapter 5, and section 7.3.3).

7.2.1 Datasets

The LIFELAB dataset (Table 1.1(1-2)) (Poolsawad *et al.*, 2011; Poolsawad *et al.*, 2012b; Poolsawad *et al.*, 2012a; Poolsawad and Kambhampati, 2014) is the main dataset used for this thesis. Table 7.1 provides for further details of class distribution. After a discussion of this dataset in Chapter 1, the size of classes of target output, as shown in Table 7.1, is imbalanced.

[1] MATLAB technical documentation - http://www.mathworks.co.uk/help/

Table 7. 2: Target classes' distribution on LIFELAB

No. of features	463	
No. of samples	1944	
Target output	Mortality	
Class	Alive	Dead
Frequency	1459	485
Ratio	3	1

Table 7. 1: Selected clinical datasets from UCI repository

Dataset	No. of features	No. of instances	No. of classes
Breast Cancer	30	569	2
Parkinson's	22	197	2
Heart Disease	13	270	2

In addition, clinical datasets (Breast cancer, Parkinson's and Heart disease) drawn from the UCI repository of machine learning[2] were applied in the feature selection experiments (Table 7.2). These contain continuous features; the rest contain only nominal features on two classes. These datasets were chosen because of the prevalence of continuous features and their predominance in the literature.

7.2.2 Missing values imputation

The missing values imputation algorithms that are provided in KEEL (Alcalá-Fdez *et al.*, 2009; Fernández *et al.*, 2009). The trend of most of the suites

[2] UCI Machine Learning Repository - http://archive.ics.uci.edu/ml/datasets.html

is to offer a good feature selection and discretization set of methods, but they overlook specialized methods of missing values imputation. Usually, the contributions included are basic modules of replacing or generating null. We used seven missing values imputations are included in the KEEL: *Data Preprocessing (Family), Missing Values (Subfamily)*.

There is a wide selection of methods available for missing value imputations. However, of these seven have been found to be most useful for clinical data (Zhang *et al.*, 2012), These are: most common value imputation (MCI); concept most common value imputation (CMCI); *K*-nearest neighbour imputation (KNNI); *K*-means clustering imputation (KMI); Fuzzy K-means clustering (FKMI); expectation-maximization imputation (EMI); and support vector machine imputation (SVMI). Of these seven, MCI, KNNI and EMI are the most commonly used imputation methods. It should be noted that CMCI is an extension of MCI, while FKMI and SVMI are recommended in literature (Luengo *et al.*, 2011).

7.2.3 Re-sampling techniques

There is always in imbalance in real clinical datasets. The reason for this is that it is always the norm that healthy (or live) patients are more numerous than patients with ill-health (or dead). Thus, any framework for clinical datasets has to deal with this reality. There are two approaches that can be used, namely (a) over-sampling the minority class e.g. SMOTE, or (b) under sampling the majority class.

7.2.3.1 Over-sampling by SMOTE

Over sampling is essentially a process of generating new samples given an imbalanced dataset. An adhoc process is simply to replicate the minority class n-number of times so that there is no major or minor class. A more systematic approach is to select some exemplars from the minority class, and then select extra samples by using nearest neighbours; often these are 3, 5, or 7 depending on the ratio of the classes. For this thesis this ratio is approximately three and thus the number of neighbours was set to be equal to 3 This is the principle behind SMOTE (section 5.3.2.1 from Chapter 5). In the experiment design, the positive examples were oversampled by using nearest neighbours = 3 that the size of positive class is 1459, roughly equal to the size of negative class 485. The SMOTE technique is embedded in the Weka package: weka.fiters.supervised.instance.SMOTE.

7.2.3.2 Under-sampling by computing the distance values

The under-sampling that is used in this thesis selects samples from the majority class (‘Alive’ class) that are furthest from the minority class (‘Dead’ class) (section 5.3.2.2 from Chapter 5). This is done using a Pairwise distance measure between the two classes (‘Dead’ and ‘Alive’ classes) of samples. For the purposes of this thesis, the Euclidean distance measure has been used. However, if the data set was of a mixed type, other measures like the Mahalanobnis distance could be used.

7.2.4 Feature selection methods

The experiments and results are based on the LIFELAB dataset and clinical datasets obtained from the UCI repository for machine learning databases. The Seattle Heart Failure Model (SHFM) (Levy *et al.*, 2006)is also used in order to compare both the selected features and the classification, since SHFM consists of the expert's selection of features without the help of a data mining algorithm. SHFM is a multivariate risk model that incorporates obtainable clinical data and laboratory variables, heart failure medications, and devices. SHFM consists of a large number of different predictors, including NYHA classification (Levy *et al.*, 2006). However, this study focuses on the variables obtained during routine blood tests, ECG, Echo and pulmonary function tests, which are all not present within the SHFM. The common variables between the two are the following eight variables Age, Sodium, Creatinine, White blood cell count, heart rate, blood pressure, albumin, BMI, Urea. All feature selection methods are based on some form of ranking of features. These rankings are obtained using '*Bioinformatics Toolbox*' from MATLAB, one of the following measures: '*t*-Test', 'Entropy', 'Bhattacharyya', 'ROC', and 'Wilcoxon' (Chapter 4). Features obtained from these ranking schemes are then compared and evaluated along with the proposed new Feature selection algorithm (FS-PPC).

7.2.5 Building the classifiers

The key to any algorithm within a data mining framework is its ability to provide correct information to the classification algorithms. In other words the "goodness" of any imputation scheme, or feature selection algorithm, or methods for handling skews in classes, is judged on how well the resultant dataset is classified. The classifiers used to assess the performance are (a) Feed Forward Networks (MLPs) (b) Radial Basis Function Networks (RBFN) (c) Support Vector machines (SVMs) (d) Decision Trees (DT) and (e) Random Forest (RF) and were discussed in Chapter 5. All of these methods are present in the software packages already mentioned. In all cases a 10-fold validation process was employed (discussed later in the next section).

- **Multilayer perceptron**

Three-layer feed forward neural networks (one hidden layer) were trained using the new data sets. Results experimented with different number of hidden units and selected the one with the best accuracy. There are used the default learning rate 0.3 and momentum rate 0.2. The training algorithm is weka.classifiers.functions.neural.NeuralNetwork.

- **Radial Basis Function Networks**

Results are used weka.classifiers.functions.RBFNetwork in Weka package to implement a normalized Gaussian radial basis function network. It uses the *k*-means clustering algorithm to provide the basis functions, and also use the default values, e.g. clustering seed, number of clusters. Symmetric multivariate Gaussians are fit to the data from each cluster.

- **Support Vector Machine**

Support Vector Machine (linear, polynomial and RBF kernel) with Sequential Minimal Optimization Algorithm, weka.classifiers.functions.SMO.

This implementation globally replaces all missing values and transforms nominal attributes into binary ones. It also normalizes all attributes by default.

- **Decision Tree**

Decision Tree classifiers were trained using each of the three rebalanced training sets. Results use weka.classifiers.trees.j48.J48 in Weka package. When building the tree, these results selected the default pruning option.

- **Random Forest**

Random forest is also available in weka.classifiers.trees. RandomForestClass is used for constructing random. In Weka allows for selecting the number of trees and controlling the number of random attributes to be chosen for each node, in this thesis 10 trees is the number of trees in the forest.

7.2.6 Assessment of the data mining process

The performance of all algorithms within a data mining framework are assessed, individually or in combination, on how well the classification is carried out. This is shown in Fig. 7.1.

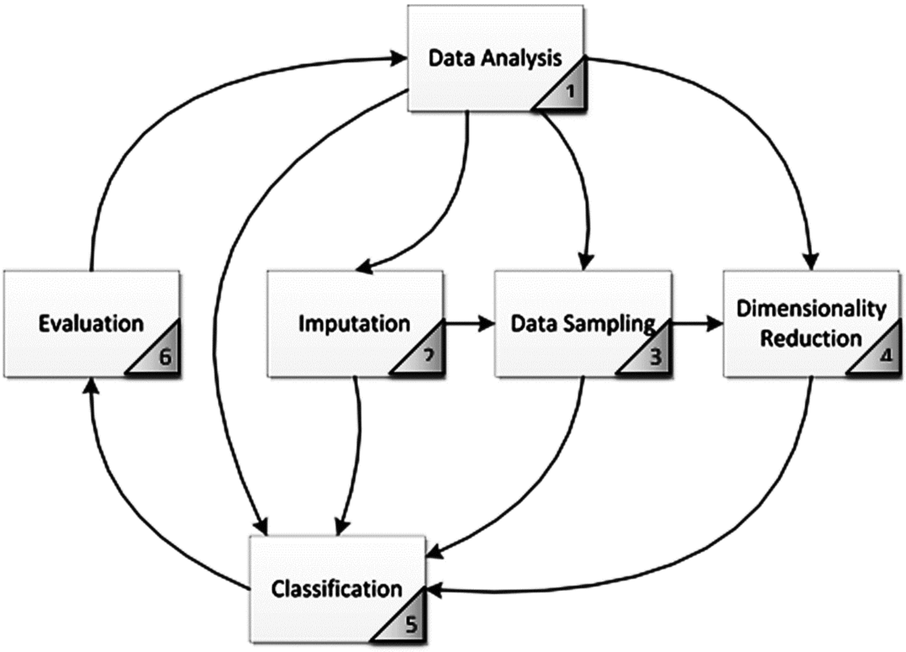


Figure 7. 1: HCDF for handling complexities of clinical dataset

For clinical datasets, apart from the ability to predict the correct class, what is crucial is the number of false positives and false negatives and the amount of redundant information present within the dataset. Thus, the evaluation in this thesis is carried out using two types of metrics (1) classification accuracy and (2) redundancy rate (Quinlan, 1992; Ramaswami and Bhaskaran, 2009). Here, redundancy rate is used for assessing the subset of features from different feature selection methods (Zhao and Wang, 2010).

Accuracy: Both Precision and Recall are used to assess the accuracy of the classifiers. These can be obtained from the data available in a Confusion matrix (Chapter 5). Both precision and recall are associated with false positives and false negatives. Thus for clinical datasets these two measure are significant (Forman, 2003; Powers, 2007). Two types of classification results are presented: 1) one with the 10-fold cross-validation and 2) a training set. The outcomes of classification which are used to form the confusion matrix are

True positive (TP):

A sample X_i is predicted to be in class Y_j, and is actually in it.

False positive (FP):

A sample X_i is predicted to be in class Y_j, but is actually not in it.

True negative (TN):

A sample X_i is not predicted to be in class Y_j, and is actually not in it.

False negative (FN):

A sample X_i is not predicted to be in class Y_j, but is actually in it.

In this thesis *precision* and *recall* are measured the effectiveness of subset of features from different feature selection schemes (Turney, 2000; Forman, 2003; Cheong Hee *et al.*, 2004; Powers, 2007). In medical diagnosis, the default assumption of equal misclassification costs underlying machine learning techniques is most likely violated. A false negative prediction may have more serious consequences than a false positive prediction (Yang *et al.*, 2009). For example, consider prediction task, where we are predicting for patient who has a high probability of being dead. Suppose that we are given a list of patients to classify as “relevant” or “non-relevant” for the dead class case, and then the cost of mistakenly assigning a relevant patient to the non-relevant patient class depends on whether there are any other relevant patents that we have correctly classified. Recall tends to be neglected or averaged away in machine learning and computational linguistics where the focus is on how confident we can be in the rule or classifier (Powers, 2007).Consequently, in this thesis both precision and recall are evaluated.

Redundancy rate: The redundancy rates is obtained by calculating the averaged correlation among the selected features returned by different feature selection algorithm (Zhao). To decide the appropriate subset of selected features; the redundancy rate is measured to assess which optimal subset will be a suitable subset of features for the dataset. In a case where the different methods appear to give a similar performance of classification, measuring the redundancy can provide a better measure of confidence on the subset of features selected. A low redundancy rate is an indication of a good set of independent variables, and an indication of low bias in the classification. Thus, if F is the subset of selected features, and X_F is the data containing features in F. The redundancy rate measured by:

$$RED(F) = \frac{1}{m(m-1)} \sum_{f_i, f_j \in F, i>j} |\rho_{i,j}| \tag{7.1}$$

where $\rho_{i,j}$ returns the correlation between the ith and the jth features, m is number of features. A large value of $RED(F)$ indicates that many selected features are strongly correlated and thus redundancy is expected to exist in F.

7.3 Results and discussions

The results presented here illustrate the data mining methods for handling the clinical data complexities that were introduced in previous chapters. Data were pre-processed for analysis and then explored to discover data characteristics.

A set of initial benchmark results were obtained, using the "Original" data (unpre-processed data). Table 7.3 shows the accuracy of the different classifiers when used on the "Original" LIFELAB dataset, which consists of 60 variables, and contains missing values and imbalanced classes. From the results, it can be seen that the classifier based on the Random forest (RF) algorithm gives better accuracy than other classifiers, with more than 90% precision and recall for both classes. This is of course on the training set. RF is a versatile classification algorithm suited for the analysis of these large datasets and a suitable classification for clinical data (Diaz and Alvarez, 2006; Pang *et al.*, 2006; Strobl *et al.*, 2008; Chen *et al.*, 2013) because RF classification models provide information on the importance of variables for the classification, leading to its superior performance on high-dimensional data (Breiman, 2004; Touw *et al.*, 2013).

On the other hand, when checked with cross validation it can be seen that the performance is not as good. For example RF precision and recall with the RF algorithm drops to 61.2% and 23.1%, for the dead class, and is at 78.8% and 95.1% for the live class. A similar drop in precision and accuracy for all classes is exhibited by all the classifiers. For example SVM shows only a marginal improvement with precision of 68.4% and recall of 32.6% for the 'Dead' class, and 80.9% and 95% for the 'Alive' class. These differences are marginal at best. However, what is significant is that the accuracies associated with the 'Alive' class are higher than those for the 'Dead' class, and also the recall values on the 'Dead' class are significantly lower than precision values. This indicates that the

‘Alive’ is better learnt than the dead class. This is a result of the existence of a far greater number of ‘Alive’ samples than ‘Dead’ samples.

Table 7. 3: The classification accuracy of different classifications using ‘Original’ LIFELAB dataset

		Test option	Class	Original	
				Accuracy	
				Precision	Recall
Classification	MLP	Cross-validation	Dead	46.5	41.4
			Alive	81.2	84.2
		Training set	Dead	98.7	93.4
			Alive	97.8	99.6
	RBFN	Cross-validation	Dead	56.4	28
			Alive	79.5	92.8
		Training set	Dead	57.9	30.1
			Alive	80	92.7
	SVM	Cross-validation	Dead	68.4	32.6
			Alive	80.9	95
		Training set	Dead	72.3	33.4
			Alive	81.2	95.8
	DT	Cross-validation	Dead	43.4	36.1
			Alive	79.9	84.4
		Training set	Dead	93.5	74
			Alive	91.9	98.3
	RF	Cross-validation	Dead	61.2	23.1
			Alive	78.8	95.1
		Training set	Dead	99.8	99.4
			Alive	99.8	99.9

From the results in Table 7.3 it can be seen that that the recall values for the ‘Dead’ class are relatively low compared to the ‘Alive’ class. This could be the result of the presence of large amount of missing values and the imbalance of classes. Missing values could be compounding the class imbalance more for the dead class than the ‘Alive’ class.

Table 7. 4: The classification accuracy of selected variables from SHFM compared with 'Original' data

		Test option	Class	Original		SHFM	
				Accuracy		Accuracy	
				Precision	Recall	Precision	Recall
Classification	MLP	Cross-validation	Dead	46.5	41.4	52.5	30.5
			Alive	81.2	84.2	79.7	90.8
		Training set	Dead	98.7	93.4	66.8	34
			Alive	97.8	99.6	81.1	94.4
	RBFN	Cross-validation	Dead	56.4	28	57.1	17.3
			Alive	79.5	92.8	77.7	95.7
		Training set	Dead	57.9	30.1	61.2	16.9
			Alive	80	92.7	77.7	96.4
	SVM	Cross-validation	Dead	68.4	32.6	0	0
			Alive	80.9	95	75.1	100
		Training set	Dead	72.3	33.4	0	0
			Alive	81.2	95.8	75.1	100
	DT	Cross-validation	Dead	43.4	36.1	52	24.3
			Alive	79.9	84.4	78.6	92.5
		Training set	Dead	93.5	74	73.5	34.2
			Alive	91.9	98.3	81.4	95.9
	RF	Cross-validation	Dead	61.2	23.1	53.8	19.2
			Alive	78.8	95.1	77.9	94.5
		Training set	Dead	99.8	99.4	99.6	97.7
			Alive	99.8	99.9	99.3	99.9

Remarks: SHFM – Seattle Heart Failure Model

Table 7.4 shows the classification from selected variables from the Seattle Heart Failure Model (SHFM). It is seen from the results that there is not much improvement in the overall performance using the features from SHFM. For example the 'Dead' class from RF (with cross-validation), precision with 'Original' data (61.2%) was significantly higher than the precision with 'SHFM' (53.8%), while the recall with 'Original' data (23.1%) was higher than with the

'SHFM' data (19.2%). What can be read from the results is that, in both cases, the accuracy (precision and recall) is very low for the dead class. This is an indication that, apart from issues associated with missing values, the imbalance in classes has a significant effect on the overall performance of the classification algorithms.

7.3.1 Missing values imputation and classification

The integrity of data is crucial in getting correct results from any analysis. Imputation schemes often predict a value that is missing. This value is often not an exact value which could have been obtained using a measurement or an observation. However, given the situation where it is missing, what can be done is to ensure that the value is as close as possible to a real value and that the overall distribution characteristics of the data are maintained. These aspects were discussed in Chapter 3 (section 3.6). The results presented here illustrate the imputation methods introduced in that chapter. The results shown in Table 7.5 are for the various imputation schemes using different classification algorithms. This table also gives the results for the same classification algorithms using the "Original" dataset. The results show that the combination of imputation based on SVM with Random Forest classification gives the best improvement in results. However, what cannot be ignored is that all imputation schemes, with any of the classification methods, give better results. The differences in the accuracy for the dead class are still lower than for the 'Alive' class, irrespective of the combination of imputation scheme and the classification algorithm. At the same time it can be seen that for the 'Dead' class, for all combinations, there is a significant difference between precision and recall. This is to do with the ability to learn the underlying characteristics associated with the class, and is a result of the small number of samples for that class. The results (Table 7.6) for the SHFM based data also confirm that the variables within the model are not sufficiently informative in that the classification performance does not significantly improve. This does not mean that the SHFM is not clinically important, but that it has to be enhanced in terms of the features available to it.

Table 7. 5: The classification accuracy of different classifications using different imputation schemes on 'Original' data

		Test option	Class	Original		Imputed					
				Accuracy		MCI		CMCI		SVMI	
				Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Classification	MLP	Cross-validation	Dead	46.5	41.4	51.2	44.3	52.2	44.9	53.2	46.6
			Alive	81.2	84.2	82.3	85.9	82.5	86.3	82.9	86.4
		Training set	Dead	98.7	93.4	98.2	88.9	98	90.7	96.1	81
			Alive	97.8	99.6	96.4	99.5	97	99.4	94	98.9
	RBFN	Cross-validation	Dead	56.4	28	57.1	27.4	59.9	29.3	60.9	32.4
			Alive	79.5	92.8	79.4	93.1	79.9	93.5	80.5	93.1
		Training set	Dead	57.9	30.1	60.3	28.9	62.2	30.9	63.4	32.2
			Alive	80	92.7	79.8	93.7	80.3	93.8	80.6	93.8
	SVM	Cross-validation	Dead	68.4	32.6	66.7	31.3	67.5	35.5	68.9	36.1
			Alive	80.9	95	80.6	94.8	81.5	94.3	81.7	94.6
		Training set	Dead	72.3	33.4	72.2	34.8	73	38.6	74.2	39.8
			Alive	81.2	95.8	81.5	95.5	82.3	95.3	82.7	95.4
	DT	Cross-validation	Dead	43.4	36.1	38.3	36.9	46.8	45.2	55.9	53
			Alive	79.9	84.4	79.3	80.3	82	82.9	84.6	86.1
		Training set	Dead	93.5	74	97	93	95	94.8	97.6	92.8
			Alive	91.9	98.3	97.7	99	98.3	98.4	97.6	99.2
	RF	Cross-validation	Dead	61.2	23.1	53.8	40.4	65.7	49.1	69.3	56.3
			Alive	78.8	95.1	81.7	88.5	84.4	91.5	86.3	91.7
		Training set	Dead	99.8	99.4	99.8	99.8	99.8	99.6	99.6	100
			Alive	99.8	99.9	99.9	99.9	99.9	99.9	100	99.9

		Test option	Class	Imputed							
				KNNI		KMI		FKMI		EMI	
				Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Classification	MLP	Cross-validation	Dead	50.7	44.7	52.2	47.2	46.9	43.7	48.7	41.6
			Alive	82.3	85.5	83	85.6	81.7	83.6	81.5	85.4
		Training set	Dead	98.4	89.5	95.4	80.8	98.9	89.3	85.7	75.3
			Alive	96.6	98	93.9	98.7	96.5	99.7	92.1	95.8
	RBFN	Cross-validation	Dead	58.1	27.2	55.8	28.9	58.4	27.8	52.3	37.3
			Alive	79.4	93.5	79.6	92.4	79.6	93.4	81	88.7
		Training set	Dead	55.6	46.4	60.5	29.7	62.4	29.5	55.4	39.2
			Alive	83.1	87.7	80	93.6	80.1	94.1	81.6	89.5
	SVM	Cross-validation	Dead	69.6	32.2	68	30.7	66.4	30.1	67.5	27.8
			Alive	80.9	95.3	80.5	95.2	80.3	94.9	79.9	95.5
		Training set	Dead	73	35.7	71.9	32.8	71.4	33	71.5	32.6
			Alive	81.7	95.6	81.1	95.8	81.1	95.6	81	95.7
	DT	Cross-validation	Dead	39.5	39	38.5	36.7	40	38.4	36.8	33.2
			Alive	79.8	80.1	79.3	80.5	79.8	80.9	78.5	81
		Training set	Dead	96.6	89.1	97.1	89.9	97.5	94.6	93.8	80.8
			Alive	96.5	99	96.7	99.1	98.2	99.2	93.9	98.2
	RF	Cross-validation	Dead	52.9	39.2	54.3	39	51.8	37.7	49	34
			Alive	81.4	88.4	81.5	89.1	81	88.3	80.1	88.2
		Training set	Dead	99.8	99.6	99.8	99.8	100	99.8	99.6	99.8
			Alive	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9

Table 7. 6: The classification accuracy of different classifications using different imputation schemes on 'SHFM' selected variables

		Test option	Class			Imputed					
				SHFM		MCI		CMCI		SVMI	
				Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Classification	MLP	Cross-validation	Dead	52.5	30.5	56.4	29.9	56.7	27.8	62	30.3
			Alive	79.7	90.8	79.8	92.3	79.5	92.9	80.2	93.8
		Training set	Dead	66.8	34	63.5	32.6	68.5	30.1	67.7	35.1
			Alive	81.1	94.4	80.7	93.8	80.4	95.4	81.4	94.4
	RBFN	Cross-validation	Dead	57.1	17.3	56	16.3	57.5	17.3	57.7	21.6
			Alive	77.7	95.7	77.5	95.8	77.7	95.8	78.4	94.7
		Training set	Dead	61.2	16.9	61.1	16.5	61.8	17.3	60.5	21.4
			Alive	77.7	96.4	83.5	77.7	77.8	96.4	78.5	95.3
	SVM	Cross-validation	Dead	0	0	0	0	0	0	0	0
			Alive	75.1	100	75.1	100	75.1	100	75.1	100
		Training set	Dead	0	0	0	0	0	0	0	0
			Alive	75.1	100	75.1	100	75.1	100	75.1	100
	DT	Cross-validation	Dead	52	24.3	49.5	22.3	50.9	24.1	53.7	32.8
			Alive	78.6	92.5	78.2	92.5	78.5	92.3	80.2	90.6
		Training set	Dead	73.5	34.2	76.3	37.1	82.1	35.1	77.2	41.9
			Alive	81.4	95.9	82.1	96.2	81.9	97.5	83.2	95.9
	RF	Cross-validation	Dead	53.8	19.2	44.6	35.5	44.8	34.8	48.9	40.4
			Alive	77.9	94.5	79.9	85.3	79.8	85.7	81.3	85.9
		Training set	Dead	99.6	97.7	99.4	100	99.2	99.6	99.4	99.8
			Alive	99.3	99.9	100	99.8	99.9	99.7	99.9	99.8

		Test option	Class			Imputed					
				KNNI		KMI		FKMI		EMI	
				Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Classification	MLP	Cross-validation	Dead	56.2	30.7	55	28.5	53.6	27.6	47.1	21.4
			Alive	80	92	79.5	92.3	79.3	92	77.9	92
		Training set	Dead	65.5	27.4	65.7	28.9	67.3	27.6	65.9	18.4
			Alive	79.8	95.2	80.1	95	79.9	95.5	78.1	96.8
	RBFN	Cross-validation	Dead	55.8	15.9	55.8	15.9	56.7	15.7	54.8	14.2
			Alive	77.4	95.8	77.4	95.8	77.4	96	77.1	96.1
		Training set	Dead	61.7	16.3	60.9	16.7	60.7	18.1	55.7	19.2
			Alive	77.6	96.6	77.7	96.4	77.9	96.1	77.9	94.9
	SVM	Cross-validation	Dead	0	0	0	0	0	0	0	0
			Alive	75.1	100	75.1	100	75.1	100	75.1	100
		Training set	Dead	0	0	0	0	0	0	0	0
			Alive	75.1	100	75.1	100	75.1	100	75.1	100
	DT	Cross-validation	Dead	52.4	22.7	49.8	21.6	51.1	23.3	44.7	17.3
			Alive	78.4	93.1	78.1	92.7	78.4	92.6	77.2	92.9
		Training set	Dead	83.1	36.5	79.8	36.7	76.3	44.5	71.7	17.7
			Alive	82.2	97.5	82.2	96.9	83.8	95.4	78.1	97.7
	RF	Cross-validation	Dead	44.7	35.9	47.4	35.1	45	33.6	39.6	29.3
			Alive	80	85.3	80.1	87	79.6	86.4	78.4	85.1
		Training set	Dead	99.8	99.6	99.2	99.8	99.6	99.6	99.2	99.4
			Alive	99.9	99.9	99.9	99.9	99.9	99.9	99.8	99.7

From both the tables it can be seen that there is an improvement in accuracy when missing values are imputed. What can be seen within the details is that precision improves significantly but recall does not improve at the same level. It should be noted that “recall” is associated with false positive classification (i.e. the ‘Dead’ class in this case), and thus for clinical application recall becomes important. Given the lack of a sufficient number of samples in this class, imputation can only improve it by small amounts. It is also evident that the imputation scheme based on SVMs provides greater improvements in the performance of classification algorithms. This method reflects the hidden information in the whole data in contrast to other methods, such as by assuming that the missing points are the same as their nearest neighbours, where local information is taken into account, resulting in bigger errors. (Honghai *et al.*, 2005). SVMI would also be useful to attempt to find heuristics to characterize the data that would act as a guide for choosing the most appropriate imputation method (Honghai *et al.*, 2005; Mallinson and Gammerman, 2005) and also it is recommended for the processing of clinical data (Wang and Wang, 2010; Zhang *et al.*, 2012). Given the performance of SVM based imputation, it is only natural to use this scheme for all future results and analysis.

7.3.2 Class balancing and classification

There are often three key components at the core of any data mining framework. These are (a) missing value (b) class imbalance and (c) dimensionality reduction and finally classification. Intuitively, it can be seen that a reduction of dimensions with the presence of imbalanced classes may not yield the right subset of features required for an improvement in performance. Thus, carrying out a balancing of classes before reduction of dimensions would be preferable.

Typically, the proportion of positive and negative cases in a dataset is not equal (usually there are many more negative cases ('Alive' in our instance) than positive cases ('Dead' class)). This imbalance affects the learning process (He and Garcia, 2009). There are two approaches which can be applied here namely over- and under-sampling. These two sampling approaches change the number of positive or negative cases in the dataset to balance their proportions; Table 7.7 shows the result of these two sampling methods.

Table 7. 7: The LIFELAB with different resampling methods

Resampling	No. of patient	Class	No. of patient
Original	1944	Alive	1459
		Dead	485
Over-sampling	2429	Alive	1459
		Dead	970
Under-sampling	1009	Alive	524
		Dead	485

What is clear from the table is that both methods change the number of samples available. Over-sampling increases the 'Dead' class and thus increases the total number of sample, while under-sampling decreases the 'Alive' class sample and thus decreases the number of the total sample. It should be noted that under-sampling can result in the removal of important examples/exemplars from the dataset, whereas over-sampling can lead to overfitting (Mease *et al.*, 2007).

Table 7.8 compares three different sets of results. In all cases, a SVM-based imputation scheme was used. The first set is the classification performance using the original, imbalanced, data set and the next two are based on the balancing approaches taken. It can be seen that balancing the classes greatly improves the performance of the algorithms. The key indicator of recall shows a significant improvement with all classification algorithms. Thus balancing of classes does lead to better performance in all indicators but shows significant improvement in the key indicators. For example, with the RF classification, precision on 'Dead' Class rises from 69.3% to 77.6% using oversampling, and 75.3% with under-sampling, while recall changes from 56.3% to 79.8% and 82.6%. A similar situation exists with data based on SHFM (Table 7.9). However, this table also illustrates the issue of reducing dimensions before balancing is carried out. Although it can be argued that the variable set is not an optimal one; it is nevertheless one used by expert clinicians. What can be concluded is that both the sampling methods improve classification (Burez and Poel, 2009; Hunt *et al.*, 2011; Liang and Zhang, 2012; Wang *et al.*, 2013), since classifiers are often biased towards the majority class (Afzal *et al.*, 2013). A key focus should be the effect of the individual strategy on rates of recall, and it can be seen that under-sampling provides marginally better recall rates.

Table 7. 8: The classification accuracy on imbalanced and balanced data

		Test option	Class	Imbalanced data		Sampling Method			
						Over-sampling		Under-sampling	
				Precision	Recall	Precision	Recall	Precision	Recall
Classification	MLP	Cross-validation	Dead	53.2	46.6	70.2	70	73	70.8
			Alive	82.9	86.4	80.1	80.3	69.5	71.8
		Training set	Dead	96.1	81	81	96.2	98.3	98.1
			Alive	94	98.9	97.1	85	97.9	98.1
	RBFN	Cross-validation	Dead	60.9	32.4	67.2	66.8	70.9	71
			Alive	80.5	93.1	78	78.3	68.6	68.5
		Training set	Dead	63.4	32.2	68.8	67.8	74.8	71.8
			Alive	80.6	93.8	78.8	79.6	70.8	73.8
	SVM	Cross-validation	Dead	68.9	36.1	74.8	66.3	73.9	74.6
			Alive	81.7	94.6	79.2	85.1	72.3	71.5
		Training set	Dead	74.2	39.8	76.6	67.5	76.8	76.5
			Alive	82.7	95.4	80	86.3	74.7	75.1
	DT	Cross-validation	Dead	55.9	53	70	69.9	74.4	75.4
			Alive	84.6	86.1	80	80.1	73	72
		Training set	Dead	97.6	92.8	97.8	98.2	97.2	98.5
			Alive	97.6	99.2	98.8	98.6	98.3	96.9
	RF	Cross-validation	Dead	69.3	56.3	77.6	79.8	75.3	82.6
			Alive	86.3	91.7	86.3	84.6	79	70.7
		Training set	Dead	99.6	100	100	99.9	99.6	100
			Alive	100	99.9	99.9	100	100	99.6

Remarks: SVM imputation

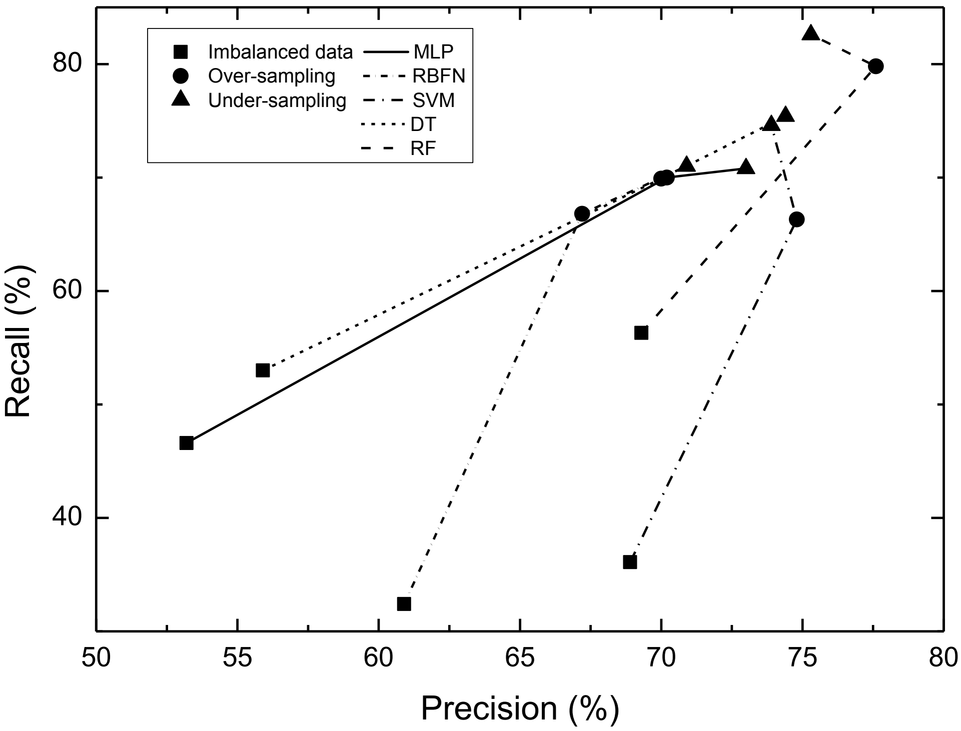


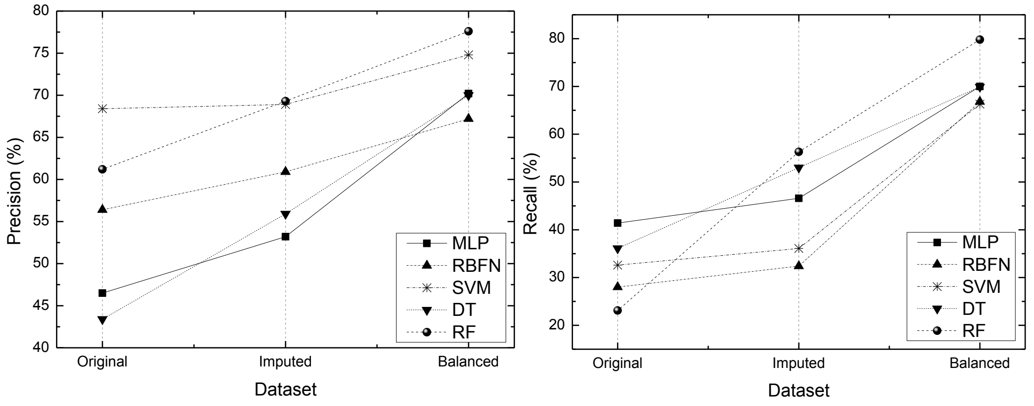
Figure 7. 2: The classification accuracy on imbalanced and balanced data

Fig. 7.2 represents the results from Table 7.8 that comparing the precision and recall of different classifiers on imbalanced and balanced (over-sampling and under-sampling) data. The results illustrate that the balanced data after applying sampling methods, greatly improves the performance; especially recall (steep slopes) values. As a result, the sampling strategies were validated by comparison of different classifiers reveal that an under-sampling may be more suitable for clinical datasets, as it reduces the proportion of negative cases and keeps the positive cases, at the same time the error rates of minority class (positive case) are minimised.

Table 7. 9: A comparison of the accuracies on different set of variables on imbalanced and balanced data

		Test option	Class	Original						SHFM					
				Imbalanced		Sampling Method				Imbalanced		Sampling Method			
						Over-sampling		Under-sampling				Over-sampling		Under-sampling	
				Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Classification	MLP	Cross-validation	Dead	53.2	46.6	70.2	70	73	70.8	62	30.3	64.1	58.5	71.8	62.2
			Alive	82.9	86.4	80.1	80.3	69.5	71.8	80.2	93.8	73.9	78.3	64.3	73.6
		Training set	Dead	96.1	81	81	96.2	98.3	98.1	67.7	35.1	57.8	84.2	71.6	68.9
			Alive	94	98.9	97.1	85	97.9	98.1	81.4	94.4	84.9	59.1	67.7	70.5
	RBFN	Cross-validation	Dead	60.9	32.4	67.2	66.8	70.9	71	57.7	21.6	64.6	47	70.6	61.5
			Alive	80.5	93.1	78	78.3	68.6	68.5	78.4	94.7	70.2	82.9	63.5	72.4
		Training set	Dead	63.4	32.2	68.8	67.8	74.8	71.8	60.5	21.4	65.7	49.1	73.6	61.1
			Alive	80.6	93.8	78.8	79.6	70.8	73.8	78.5	95.3	71	82.9	64.5	76.3
	SVM	Cross-validation	Dead	68.9	36.1	74.8	66.3	73.9	74.6	0	0	67.5	42.3	67.5	64.7
			Alive	81.7	94.6	79.2	85.1	72.3	71.5	75.1	100	69.3	86.5	63.5	66.4
		Training set	Dead	74.2	39.8	76.6	67.5	76.8	76.5	0	0	68.2	43.9	69.5	66
			Alive	82.7	95.4	80	86.3	74.7	75.1	75.1	100	69.8	86.4	65.2	68.7
	DT	Cross-validation	Dead	55.9	53	70	69.9	74.4	75.4	53.7	32.8	65.3	61.4	65	67.6
			Alive	84.6	86.1	80	80.1	73	72	80.2	90.6	75.3	78.3	63.4	60.6
		Training set	Dead	97.6	92.8	97.8	98.2	97.2	98.5	77.2	41.9	75.9	69.6	74.2	76.9
			Alive	97.6	99.2	98.8	98.6	98.3	96.9	83.2	95.9	80.8	85.3	74	71.1
	RF	Cross-validation	Dead	69.3	56.3	77.6	79.8	75.3	82.6	48.9	40.4	64.9	68.6	66	71.8
			Alive	86.3	91.7	86.3	84.6	79	70.7	81.3	85.9	78.3	75.4	66.3	60
		Training set	Dead	99.6	100	100	99.9	99.6	100	99.4	99.8	99.7	99.9	99.6	100
			Alive	100	99.9	99.9	100	100	99.6	99.9	99.8	99.9	99.8	100	99.6

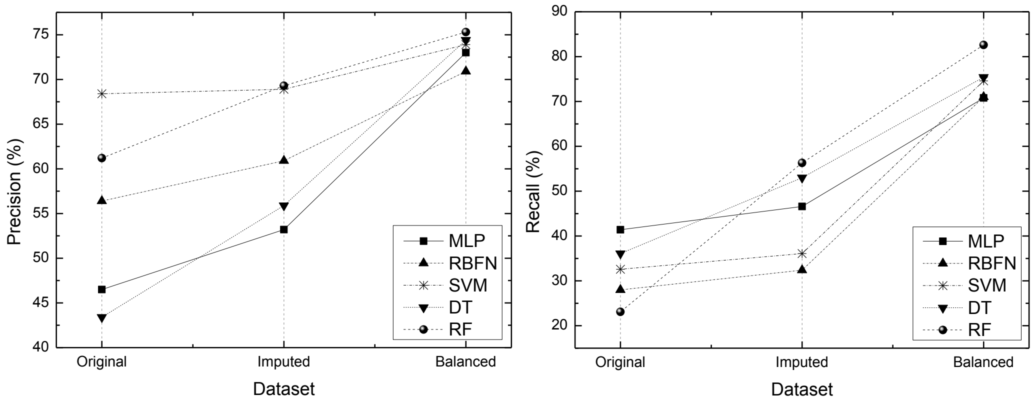
Remarks: SVM imputation



(a) Precision

(b) Recall

Figure 7. 3 (a-b): The classification accuracy on data mining process for over-sampling



(a) Precision

(b) Recall

Figure 7. 4 (a-b): The classification accuracy on data mining process for under-sampling

The graphs in Fig. 7.3 (a-b) and Fig. 7.4 (a-b) illustrate the above analysis further. These graphs show the changes to precision and recall, under three different conditions, namely: original data set, dataset with imputation and dataset with different sampling strategies. It can be seen that improvements are made progressively at each stage. It can be seen that there are sharp increases after sampling the data post imputation. Fig. 7.3 (a) and Fig. 7.4 (a) show that precision from both sampling have a slightly different improvement. Fig. 7.3 (b) and Fig.

7.4 (b) show that under-sampling provides improved marginally better recall rates than over-sampling. This further reinforces the evidence for the proposed framework, and also for the steps described earlier in this section. However, a key component of the framework, i.e. the high dimensionality, has not been examined as yet.

7.3.3 Feature selection and classification

Reduction of dimensions is essentially carried out not to improve performance but to increase the numerical tractability of the data mining problem. The reason for this expectation is that a reduction in dimensionality often leads to a reduction in the overall information present in the reduced dimensions. What is important is to keep this loss to a minimum, and thus there is always a compromise to be made between loss of information and numerical complexity.

Often clinical data consists of variables, which are useful, marginally useful and not useful at all, from an information content point of view. Therefore, it is possible that there are variables which are important for a clinician but do not necessarily have much information present within them for the classification algorithm. To surmount the difficulties caused by high dimensionality, various approaches have been proposed and developed. The problem is one of computationally selecting a small set of relevant variables from a large set (high dimensionality) such that the selected variables are representative of the whole dataset. Often the reduction in dimensions, and thus selection of features, is carried out by ranking them in order of importance or information content. By selecting an appropriate threshold, different numbers of features can be selected at any given time. Thus, the results shown in this section illustrate the effect of selecting different numbers of features (variables/dimensions). These are also compared to feature extraction methods. Often, it is considered that feature extraction carries more information to the new dataset. However, for clinical datasets, the meaning of the new variables is important. In terms of information content present, the data set obtained using feature extraction is more than that with feature selection. Based on these results a new feature selection algorithm has been suggested in Chapter 6. The method is very similar to the feature selection methods, and is an integration of both extraction and selection, using mutual information and redundancy tools. As a benchmark, Table 7.10 has a set of results, which illustrates the performance of feature extraction. In all cases, the dataset used SVM for imputation, and over-sampled the dead-class, in order to show changes within an already low performance. Two types of feature extraction schemes (a) PCA and (b) NL-PCA were tested. The performance does not change much; indeed in the odd situation there is an improvement in the accuracy. This could be a result of removing non-independent features from the data set. It should be noted that in general, a larger number of features should give better classification performance, however, it has been found that in practice, fewer features are required to retain or improve the performance (Janecek *et al.*, 2008).

Table 7. 10: The classification accuracy of different selection methods on imputed missing values by SVMI and balanced data by over-sampling

		Test option	Class	Original		Feature Extraction							
						PCA				NLPCA			
				60 features		33 features		20 features		33 features		20 features	
				Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Classification	MLP	Cross-validation	Dead	70.2	70	70	69.9	68.1	67.4	64.5	64.7	64.7	58.9
			Alive	80.1	80.3	80	80.1	78.1	78.7	76.5	76.3	74.2	78.7
		Training set	Dead	81	96.2	88	93.7	77.1	90.1	96.6	94.1	67.5	73.8
			Alive	97.1	85	95.6	91.5	92.6	82.2	96.2	97.8	81.4	76.4
	RBFN	Cross-validation	Dead	67.2	66.8	67.6	60.7	67.9	59.1	64.6	52.8	63.5	43.2
			Alive	78	78.3	75.5	80.7	75	81.4	72	80.8	68.9	83.5
		Training set	Dead	68.8	67.8	71.7	62	69.3	62.3	68	53.6	64.8	44
			Alive	78.8	79.6	76.8	83.8	76.5	81.6	73	83.2	69.3	84.1
	SVM	Cross-validation	Dead	74.8	66.3	73	64.2	72.6	64.5	70.8	61.1	70.3	46.8
			Alive	79.2	85.1	78	84.2	78	83.8	76.3	83.3	71.1	86.8
		Training set	Dead	76.6	67.5	74.5	65.3	73.3	64.9	72.3	63.1	70.8	47.8
			Alive	80	86.3	78.7	85.1	78.3	84.2	77.4	84	71.5	86.9
	DT	Cross-validation	Dead	70	69.9	58.5	58.7	61.2	58.6	59.6	57.9	61.8	54.6
			Alive	80	80.1	72.5	72.4	73.2	75.3	72.5	73.9	72	77.5
		Training set	Dead	97.8	98.2	94.1	83.5	85.2	91.3	86.7	91.9	83.9	64.3
			Alive	98.8	98.6	89.8	96.5	94	89.4	94.4	90.6	79.5	91.8
	RF	Cross-validation	Dead	77.6	79.8	65.8	68.6	56.6	57.3	61.8	65.9	65.2	69
			Alive	86.3	84.6	78.5	76.3	71.4	70.7	76.3	72.9	78.5	75.5
		Training set	Dead	100	99.9	99.9	100	100	100	99.4	100	100	99.8
			Alive	99.9	100	100	99.9	100	100	100	99.6	99.9	100

For clinical data, the key is the retention of the variable labels. Feature extraction provides reduced dimensionality but does not retain the labels and hence is often not of much use in the development or design of diagnostic or prognostic tools. Feature selection, on the other hand, retains the labels and is preferred to extraction for this reason, in spite of a reduction in information content. (See chapter 4 for more details). All the feature selection methods selected use ranking of features based on one of the following measures: '*t*-Test', 'Entropy', 'Bhattacharyya', 'ROC', and 'Wilcoxon'. Features obtained from these ranking schemes are then compared and evaluated along with the proposed new feature selection algorithm (FS-PPC). These results are shown in Tables 7.11−7.13. These tables show the performance of these selection methods under different conditions. For all results the data had missing values imputed using SVM. The tables show how the feature selection works under three different conditions namely: (a) original set of classes (b) under-sampled and (c) over sampled. These results would further reinforce the framework procedure outlined earlier.

It can be seen from Table 7.11 shows that the results of 20 selected features on the unbalanced dataset, almost all the feature selection algorithms give a similar set of poor results as compared to the full dataset. Indeed in some cases the performance drops; this is more pronounced with the dead class and also with the recall for both classes. Both Tables 7.12 and 7.13 illustrate the importance of balancing the data. However, it should be noted that the Random Forest Classification algorithm outperformed all algorithms under the different conditions. The results illustrate what is possible with the different classification algorithms. The overall performance is greatly improved for all selection algorithms and a choice between under-sampling or over-sampling cannot be made from these results. However, under-sampling may discard potentially important cases from the majority class of the sample and can lead to overfitting of similar instances (Crone and Finlay, 2012). Therefore, under-sampling tends to overestimate the probability of cases belonging to the minority class, while over-sampling tends to underestimate the likelihood observations belonging to the minority class (Weiss, 2004). As both sampling methods can potentially reduce accuracy in generalising for unseen data, authors have presented viewpoints on the gains in accuracy derived over-sampling versus under-sampling (Drummond and Holte, 2003; Maloof, 2003; Chawla *et al.*, 2004; Prati *et al.*, 2004; Crone and Finlay, 2012), indicating that the results are not universal and depend on the dataset properties and the application domain.

Table 7. 11: The classification accuracy of different feature selection methods on imbalanced data

		Test option	Class	Feature Selection									
				t-Test		Entropy		Bhattacharyya		ROC		Wilcoxon	
				Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Classification	MLP	Cross-validation	Dead	50.5	41.2	56.6	46	49	40.8	52.1	37.5	53.6	38.4
			Alive	81.6	86.6	83.1	88.3	81.4	85.9	81	88.6	81.3	89
		Training set	Dead	76.8	60.6	79.9	45.8	78.4	55.3	76.5	61.9	79	53.6
			Alive	87.8	93.9	84.2	96.2	86.5	94.9	88.1	93.7	86.1	95.3
	RBFN	Cross-validation	Dead	62.9	31.8	61.8	31.8	59.8	30.1	63.4	27.8	63.4	27.8
			Alive	80.5	93.8	80.5	93.5	80.1	93.3	79.8	94.7	79.8	94.7
		Training set	Dead	64	33.4	63.6	29.9	64.4	29.1	64.5	28	64.5	28
			Alive	80.9	93.8	80.2	94.3	80.1	94.7	79.9	94.9	79.9	94.9
	SVM	Cross-validation	Dead	67.3	22.1	65.6	20.4	69.5	15.1	66.3	21.9	66.3	21.9
			Alive	78.8	96.4	78.5	96.4	77.6	97.8	78.8	96.3	78.8	96.3
		Training set	Dead	68.9	23.3	69	22.5	71.1	17.7	67.1	22.3	66.7	21.9
			Alive	79.1	96.5	78.9	96.6	78.1	97.6	78.9	96.4	78.8	96.4
	DT	Cross-validation	Dead	55	49.7	54.9	47.4	50.1	44.7	58	50.9	58.1	50.9
			Alive	83.8	86.5	83.3	87	82.3	85.2	84.3	87.7	84.3	87.8
		Training set	Dead	95.3	75.3	89.9	76.7	91.4	80.6	96.9	77.7	96.9	77.7
			Alive	92.3	98.8	92.6	97.1	93.8	97.5	93.1	99.2	93.1	99.2
	RF	Cross-validation	Dead	63.8	56.7	66.8	57.3	54.5	44.5	64.5	57.7	61.1	55.1
			Alive	86.1	89.3	86.5	90.5	82.6	87.7	86.4	89.4	85.5	88.3
		Training set	Dead	99.8	99.8	99.8	99.8	99.8	99.8	99.6	99.6	99.6	99.8
			Alive	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9

Remarks: SVM imputation

Table 7. 12: The classification accuracy of different feature selection methods on imbalanced data by over-sampling

		Test option	Class	Feature Selection									
				t-Test		Entropy		Bhattacharyya		ROC		Wilcoxon	
				Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Classification	MLP	Cross-validation	Dead	65.8	66.8	66.5	66.9	63.9	62.6	65.7	63.9	64.8	63.3
			Alive	77.7	76.9	77.9	77.6	75.5	76.5	76.4	77.9	76	77.2
		Training set	Dead	69.8	86.6	70.6	85.4	73.1	80.3	73.8	75.6	69.8	82.9
			Alive	89.4	75.1	88.7	76.4	86	80.3	83.5	82.2	87	76.1
	RBFN	Cross-validation	Dead	66	55.6	66.7	61.9	65.3	56.8	66	57.3	65.9	54.9
			Alive	73.3	80.9	75.8	79.5	73.6	79.9	73.9	80.4	73	81.1
		Training set	Dead	67	56.9	67.4	62.2	66.9	58.1	66.9	57.7	66.3	55.9
			Alive	74	81.4	76.1	80	74.4	80.9	74.2	81	73.4	81.2
	SVM	Cross-validation	Dead	70	62.5	70.2	60.1	69.1	58.5	69	59.7	69.1	60.2
			Alive	76.7	82.2	75.8	83.1	75	82.7	75.4	82.2	75.6	82.1
		Training set	Dead	70.6	63.1	70.5	60.6	70.2	60.1	70.2	60.9	70.3	61.3
			Alive	77.1	82.5	76.1	83.1	75.8	83	76.1	82.8	76.3	82.8
	Decision Tree	Cross-validation	Dead	68.5	66.7	69.2	70.3	67.6	65.1	66.6	66.8	66.1	66.5
			Alive	78.2	79.6	80	79.2	77.3	79.2	77.9	77.7	77.6	77.3
		Training set	Dead	88.6	95.5	88.9	94.5	91.9	92.7	89.5	91.8	95.4	96.2
			Alive	96.8	91.8	96.2	92.2	95.1	94.6	94.4	92.9	97.5	96.9
	Random Forest	Cross-validation	Dead	72.2	76.7	75	79.6	71.3	76.2	75	77.3	73.3	78.1
			Alive	83.8	80.4	85.9	82.4	83.4	79.6	84.6	82.9	84.8	81.1
		Training set	Dead	99.7	99.8	99.7	99.8	99.8	99.9	99.7	100	99.8	100
			Alive	99.9	99.8	99.9	99.8	99.9	99.9	100	99.9	100	99.9

Remarks: SVM imputation

Table 7. 13: The classification accuracy of different feature selection methods on imbalanced data by under-sampling

		Test option	Class	Feature Selection									
				t-Test		Entropy		Bhattacharyya		ROC		Wilcoxon	
				Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Classification	MLP	Cross-validation	Dead	67.3	66.4	70.3	67.2	67.7	58.4	66.7	68.3	69.3	67.7
			Alive	64.2	65.2	66.1	69.3	60.9	69.9	64.8	63.1	66	67.6
		Training set	Dead	89.1	82.6	87.9	83	82.8	88.4	82.5	84.7	88	78.6
			Alive	82.6	89.1	82.7	87.6	86.4	80.2	83	80.6	79.3	88.5
	RBFN	Cross-validation	Dead	67.6	67.2	72.5	69.1	70.7	58.4	68.4	68.1	67.3	66.8
			Alive	64.8	65.2	68.2	71.8	62.2	73.8	65.7	66	64.4	64.9
		Training set	Dead	70	68.7	73.3	70.2	70.4	63.5	69.2	69.8	68.5	67.6
			Alive	66.9	68.2	69.2	72.4	64.4	71.1	67.1	66.4	65.4	66.4
	SVM	Cross-validation	Dead	72.5	68.3	73.2	68.3	72.3	68.3	70.9	69.7	71.1	67.9
			Alive	67.8	72	68.1	73	67.7	71.8	67.8	69.1	66.9	70.1
		Training set	Dead	73.8	68.7	74.2	69.3	73.3	69.3	72.1	71.9	72.7	68.7
			Alive	68.5	73.6	69	74	68.7	72.8	69.8	69.9	68.1	72.2
	Decision Tree	Cross-validation	Dead	71.5	70.2	75.2	66.4	64.9	66.6	72.6	75.8	71.7	72.9
			Alive	68.4	69.7	67.8	76.3	62.8	61	72.5	69.1	70.2	68.9
		Training set	Dead	94.8	94.8	96.6	87.8	96.8	92.7	93.3	92.6	95.6	95.6
			Alive	94.4	94.4	88	96.7	92.5	96.7	92	92.8	95.3	95.3
	Random Forest	Cross-validation	Dead	72.9	79.4	76.6	81.1	68	76.1	74.5	80.2	73.5	77.3
			Alive	75.3	68	78.2	73.2	70.4	61.2	76.6	70.3	74	69.9
		Training set	Dead	100	100	100	100	100	99.8	99.8	100	99.6	99.8
			Alive	100	100	100	100	99.8	100	100	99.8	99.8	99.6

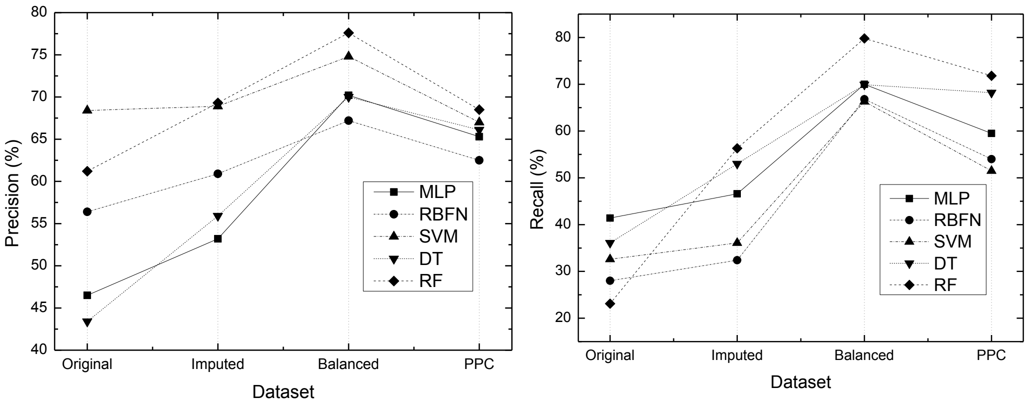
Remarks: SVM imputation

Feature extraction is a useful method for reduction of dimensions; however, it is not useful for applications that need to use meaningful (and perhaps the original) labels of the features. It has a distinct advantage in that accuracy in classification performance is high. On the other hand, feature selection is a process of selecting a subset of original features based on a desired criterion. It reduces the number of features, removes variables that are irrelevant and redundant, and also the computational complexity is lower than that of feature extraction. Both these methods for reducing dimensions are extensively used, and the choice is dependent on the application and the problem to be solved. The proposed methodology combines the two, feature extraction and selection. In this new methodology, the advantages of feature extraction and feature selection are retained and combined. FS-PPC selects an optimal subset of features by projecting a principal component from feature extraction and removing redundant features. Table 7.14 shows the performance of classification of different classifiers on imbalanced and balanced data by using the FS-PPC method; these results are a little lower than for 'Original' data, correspondingly with other feature selection methods. For example, RF showed the results on the 'Dead' class with cross-validation; the precision was 68.5% and recall was 71.8% on balanced data by using over-sampling, and also the precision was 62.5% and recall was 72.5% on balanced data by using under-sampling. Fig. 7.3 (a-b) and Fig. 7.4 (a-b) show the effect of selecting features on the performance of the classification. It can be seen from Tables 7.10 and 7.11 that feature extraction provides better performance compared to feature selection. However, it was argued in Chapter 6 that a combination of feature extraction and selection would do two things: (a) improve on the accuracy of feature selection and (b) retain the labels of the variables. Indeed, it can be seen from Table 7.14 that improved performance is obtained using the new feature selection algorithm. However, this does not mean that the performance is at the same level as when all the features are used (Fig. 7.5 (a-b) and Fig. 7.6 (a-b)).

Table 7. 14: The classification accuracy of FS-PPC on LIFELAB dataset

		Test option	Class	PPC					
						Sampling Method			
						Over-sampling		Under-sampling	
				Precision	Recall	Precision	Recall	Precision	Recall
Classification	MLP	Cross-validation	Dead	47.3	38.6	65.3	59.5	63.8	64.1
			Alive	80.8	85.7	74.6	79	61	60.6
		Training set	Dead	74.9	54.2	66.1	81.6	76.9	84.7
			Alive	86.1	94	85.5	72.1	81.5	72.6
	RBFN	Cross-validation	Dead	54.8	31.5	62.5	54	63.2	70.6
			Alive	80.1	91.4	72	78.5	63.7	55.7
		Training set	Dead	59.8	34.6	64.5	57.4	65.4	71.2
			Alive	80.9	92.3	73.6	79	65.6	59.4
	SVM	Cross-validation	Dead	58.6	3.5	67	51.5	66.8	72.5
			Alive	75.6	99.2	72.1	83.1	67.3	61
		Training set	Dead	66.8	4.5	68.7	52.4	68.7	74.2
			Alive	75.8	99.3	72.7	84.2	69.5	63.5
	DT	Cross-validation	Dead	46.8	36.5	66.1	68.2	65.8	60.9
			Alive	80.3	86.2	78.4	76.7	60.9	65.8
		Training set	Dead	90.2	55.1	86	91.9	91.4	89.3
			Alive	86.8	98	94.3	90.1	88.7	90.9
	RF	Cross-validation	Dead	56.4	43.7	68.5	71.8	62.5	72.5
			Alive	82.6	88.8	80.6	78.1	64.1	53
		Training set	Dead	99.6	99.8	99.7	99.9	99.4	99.8
			Alive	99.9	99.9	99.9	99.8	99.8	99.4

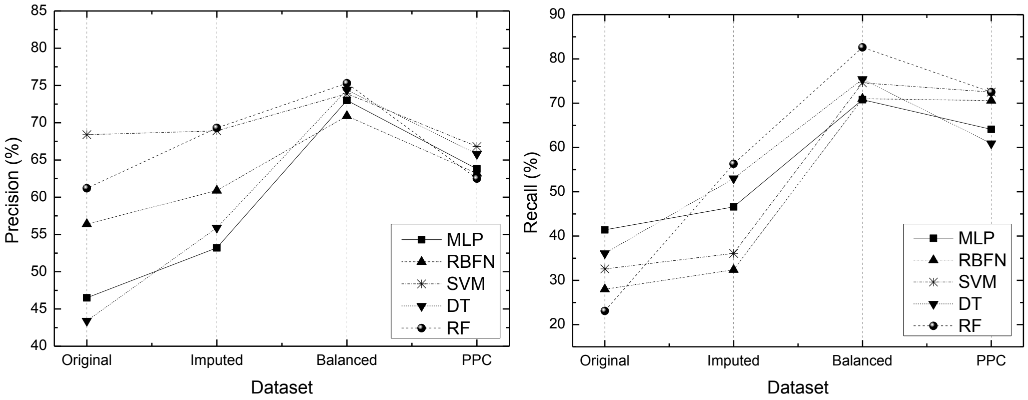
Remarks: SVM imputation



(a) Precision

(b) Recall

Figure 7. 5 (a-b): The classification accuracy on data mining process for over-sampling



(a) Precision

(b) Recall

Figure 7. 6 (a-b): The classification accuracy on data mining process for under-sampling

Fig. 7.5 (a-b) and Fig. 7.6 (a-b) summarise the classification performance under the following conditions: missing imputation (SVMI), data sampling (over- and under-sampling), and feature selection (FS-PPC). It can be concluded as:

- *With imputation of Missing values* there is an improved classification performance.
- *With Data sampling* the recall (sensitivity) of minority class is improved at a better rate

- *With Feature selection* the performance of classification is nearly retained at the same level as before.

It has been mentioned before that all dimensionality reduction techniques use a ranking scheme. The number of dimensions retained is dependent on the threshold being used. If this is varied, then the number of dimensions is also varied. It should also be recalled that for modelling, the variables should be independent of each other. This is done in an automatic manner with feature extraction; however with feature selection what is needed is a test of redundancy, i.e. relevant and non-redundant features (Yu and Liu, 2004) for effective selection with an optimal set of features.

Table 7. 15: The redundancy rate of feature extraction methods

Feature Extraction	Redundancy Rate	
	Over-sampling	Under-sampling
PCA	1.15E-16	1.03E-16
NLPCA	0.1133	0.1117

Table 7. 16: The redundancy rate of different selection methods on different data

Feature Selection	Redundancy Rate		
	Imbalanced classes	Balancing Methods	
		Over-sampling	Under-sampling
t-Test	0.1005	0.1029	0.1190
Entropy	0.0857	0.0872	0.0964
Bhattacharyya	0.0737	0.0750	0.0739
ROC	0.0961	0.0991	0.1016
Wilcoxon	0.0961	0.1022	0.1050
PPC	**0.0651**	**0.0621**	**0.062**

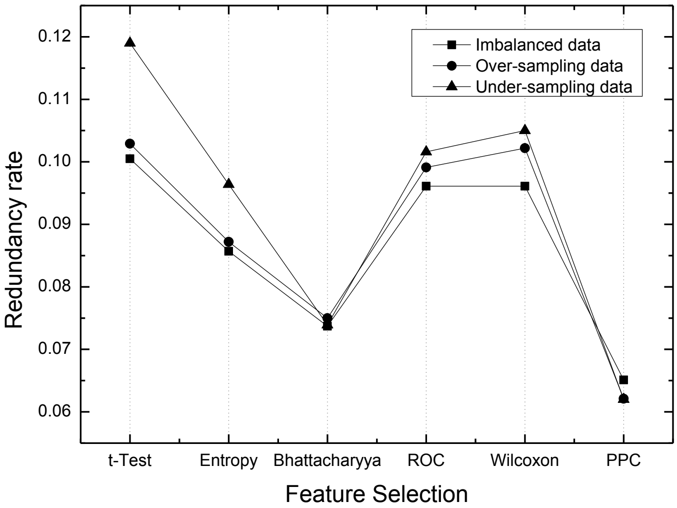


Figure 7. 7: The redundancy rate of different selection methods on different data

Tables 7.15−7.16 show the redundancy rate (see Eq. (7.1)); Table 7.16 and Fig. 7.7 show the redundancy rates with different feature selection methods, and with both sampling techniques. It can be seen that of the five, the Bhattacharyya distance method has the lowest redundancy rate. However, the new feature selection technique (FS-PPC) has an even lower redundancy rate. The reason for this is that the new method has a feature extraction process coupled with a selection algorithm which utilizes mutual information and this reduces the redundancy within the selected features. Hence, the features are more suitable for the purposes of classification.

To evaluate the FS-PPC further, three datasets drawn from the UCI repository of Machine Learning (Breast cancer, Parkinson's and Heart disease) are used in this thesis. These datasets were chosen because of the prevalence of variables (features) which are continuous and are representative of clinical datasets. Table 7.2 shows the key characteristics of the selected datasets from the UCI repository. The performance of a particular dimension reduction technique is judged by how well the new set of dimensions are able to retain the key properties of the dataset, and the relationship with the outcomes. In the case of the above datasets the problem is one of classification. It was decided to select the most popular method of classification, the decision tree (DT), as this was found to provide a good uniform set of results, when compared to a range of other algorithms.

Table 7. 17: The classification accuracy on UCI balanced dataset

Dataset	Class	*t*-Test			Entropy			Bhattacharyya		
		No. of features	Precision	Recall	No. of features	Precision	Recall	No. of features	Precision	Recall
Breast Cancer	Malignant	10	91.7	88.2	10	91.7	89.2	10	91.7	89.2
	Benign		93.2	95.2		93.7	95.2		93.7	95.2
Parkinson’s	Parkinson’s	10	87.4	89.8	10	87.1	96.6	10	89.9	90.5
	Healthy		65.9	60.4		84.4	56.3		70.2	68.8
Heart Disease	Presence	7	79.1	75.8	7	80.7	73.3	7	80.7	73.3
	Absence		81.3	84		80.1	86		80.1	86
Dataset	Class	ROC			Wilcoxon			PPC		
		No. of features	Precision	Recall	No. of features	Precision	Recall	No. of features	Precision	Recall
Breast Cancer	Malignant	10	90.3	88.2	10	89.5	88.7	10	91.3	88.7
	Benign		93.1	94.4		93.3	93.8		93.4	95
Parkinson’s	Parkinson’s	10	85.7	89.8	10	92.3	89.1	10	90.5	90.5
	Healthy		63.4	54.2		69.8	77.1		70.8	70.8
Heart Disease	Presence	7	79.1	75.8	7	78.8	74.2	7	80.2	70.8
	Absence		81.3	84		80.3	84		78.7	86

Table 7. 18: The redundancy rate of different selection methods on different data

Feature Selection	Redundancy Rate		
	Breast Cancer	Heart Disease	Parkinson’s
t-Test	0.4094	0.1415	0.3698
Entropy	0.4030	0.1197	0.4142
Bhattacharyya	0.4030	0.1197	0.2332
ROC	0.4163	0.1415	0.3651
Wilcoxon	**0.1760**	0.1067	0.2109
PPC	0.2161	**0.0940**	**0.1696**

Table 7.18 shows the redundancy rates of the optimal subset of features. It can be seen that the redundancy rate for FS-PPC is good. In particular, it performed well with the Heart Disease dataset, where its RED was 0.0940. However, although it reduced the rate with the Parkinson’s dataset (0.1695), it was not the case with the Breast Cancer dataset. In the Breast Cancer dataset, Wilcoxon (0.1760) had its redundancy rate lower than others (FS-PPC was 0.2162) but considering the accuracies in Table 7.17, was also lower than others.

7.4 Summary

According to the HCDF framework, process flexibility depends on the problems and datasets that need to be solved. In this thesis, the LIFELAB was used and the experiments were set-up correspondingly this framework. Hence, we suggest the data mining procedure as follows:

1. *Analysis of the characteristics of dataset to identify the data problems that needs to be solved.*
2. *For datasets with missing values; imputation techniques are applied to impute the missing values.*
3. *For imbalanced classes, over-sampling and under-sampling are applied to treat the imbalanced classes.*
4. *Feature selection is applied to reduce dimensionality.*
5. *A classifier is built to evaluate the performance of classification.*

To evaluate the data mining techniques, accuracy and redundancy rate are measured.

The results shown in this chapter used data primarily from LIFELAB, and in the case of FS-PCC were further validated with the help of three additional clinical datasets (Breast cancer, Parkinson's and Heart disease). In these experiments the interest was to identify general properties and differences in the methods employed in classification to evaluate the handling of complexity – including missing values, class imbalance and dimensionality – of clinical data. The framework helped in this and also showed that an improvement in accuracy and confidence of disease diagnosis and prognosis can be achieved. The removal of irrelevant – or even misleading, and also selected features (predictors/variables) was crucial in the application of any classifier available in literature and in this study. In Chapter 6, it was proposed to combine the best of both feature extraction and feature selection approaches, and the resultant method was evaluated. The new method of feature selection was based on a recursive feature elimination process using the MI and SU together with principal component(s) from feature extraction. Here, because the principal component was projected on the set of features, it resulted in a better classification performance. It outperformed the direct application of the random forest classifier, or the direct application of the regularised classifiers on the full set of features.

In theory, a larger number of features should give a better classification performance (Janecek *et al.*, 2008). However, it was found that in practice, fewer features are required to retain or improve the performance. The results show that even though the FS-PPC selects fewer features it can maintain the performance of classification (precision and recall). In addition, FS-PPC produces a lower the redundancy rate of features is lower than other methods. Thus, this method is suitable to use for selecting an “optimal” feature set. FS-PPC allows for accuracy in the data mining process to maintain or reduce the redundancy rate, while at the same time also keeping the computational overheads low.

.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

8.1 Introduction

Data mining techniques can solve the problems of extraction of information from data but, like any statistical technique, they also have the power to reveal results that do not occur naturally. Specifically, this thesis has investigated issues with clinical datasets, such as missing values, class imbalance and high dimensionality. The research in this thesis was motivated by these challenges to minimise the problems whilst, at the same time, maximising classification performance of data. As such, this led to the proposal of a data mining framework and feature selection method. The proposed framework has a simple algorithmic framework and makes use of a modified form of existing frameworks to address a variety of different data issues. This framework, called the Handling Clinical Data Framework (HCDF) that was discussed in section 2.3.3 from Chapter 2. Next, the proposed feature selection method, was introduced; it involves projecting onto principal component method (FS-PPC) and draws on ideas from both feature extraction and feature selection to select a significant subset of features from the data. This method selects features that have high correlation with the principal component by applying SU. However, irrelevant and redundant features are removed by using MI (see the details in Chapter 6). This method provides confidence in the selected subset of features that will yield realistic results with less time and effort. Since principal component from feature extraction reflects non-redundant features, while feature selection selects meaningful features. FS-PPC integrates both methods, so that the optimal subset of features from this method is able to retain classification performance and meaningful features while consisting of non-redundant features. The optimal subset of features from this method reduces misclassification because the final set of features will avoid redundant features. The assessment of data mining techniques reveals that missing values imputation and resampling data for class balancing can increase the performance of classification and the proposed feature selection method produces acceptable performance and a confident subset of selected features for classification of clinical datasets. This chapter concludes the thesis with a summary of the main contributions of the thesis and gives a summary and some suggestions for future research.

8.2 Contributions of the research

Chapter 1 introduced the problem issues of data mining in clinical datasets and indicated a list of key issues that are of concern in this field. From this, a list of motivation and research problems was formulated (section 1.2) to outline specific goals of the research (section 1.3). To assess and support this, the research aim and objectives were defined. In the following discussion, we revisit these objectives and summarise how, and to what extent, they have been achieved. This section will discuss the research findings in light of these problems, referring to each objective by its corresponding problems.

Objective 1: To develop a data mining framework for classification based on the underlying statistical properties of the datasets and the existing frameworks

A data mining framework for clinical datasets was proposed, and explained in section 2.3.3 from Chapter 2. This framework is called Handling Clinical Data Framework (HCDF). It was developed by modifying by existing frameworks e.g. CRIPS (Wirth and Hipp, 2000) and SEMMA (SAS Institute Inc.). This approach is used to address data mining problems, specifically the classification problem in clinical datasets. HCDF is expected to be efficient for clinical datasets (Chapter 1, section 1.1) and it is used for assessment of data mining techniques in Chapter 7. This framework consists of six main processes: (1) data analysis (2) missing values imputation (3) dimension reduction using feature selection techniques (4) data sampling (5) classification and (6) evaluation. This framework is effective for handling complexities in the dataset and also demonstrated its importance as a flexible procedure for coping with different issues in datasets. The main focus of the framework is to reduce problem of data mining and increase the performance and the reliability of classification in clinical data.

Objective 2: To investigate the relationship between the methods for imputation and the statistical properties of the datasets

The characteristics of the clinical dataset were investigated and data mining issues were revealed. Chapter 1 discussed the issues in clinical data sets. Chapter 2 presented a data mining framework which can cope with the issues of missing values, imbalanced class, high dimensionality, and classification. The performance of classification can be used to evaluate the performance of the predictive model. In our study, we found that missing values handling and class balancing were able to improve the classification performance. In Chapter 3, missing values imputation techniques were demonstrated for imputing missing values for a relatively complex data structure such as a clinical dataset when the data contain missing values. The several types of imputations generalise values of variables being imputed. However, the data need to be adjusted to take into account imputed values. After imputation was applied, the statistics values and data distribution were changed (section 3.7) whilst, at the same time, the results showed that the classification performance was improved (section 7.3.1).

Objective 3:

(a) To discover the effect of class imbalance on performance of classification and propose the sampling data method for balancing data

(b) To investigate feature selection techniques in clinical datasets

High dimensionality and class imbalance are essentially challenges present in all real-life clinical datasets. High dimensionality was presented in Chapter 1 and to cope with the issue, dimensionality reduction techniques were demonstrated in Chapter 4, including both feature extraction and feature selection techniques. Feature extraction can reduce the dimensions of the dataset but, like feature selection, it generates new features without an associated meaning. PCA (section 4.2.1) is a popular technique for feature extraction, and the NLPCA (section 4.2.2) is an extended form of this; these two examples of feature extraction were presented. Feature selection selects the features from the original data. This technique, including wrapper and filter models, was demonstrated to reduce the dimensionality of data (section 4.3). Obtaining meaningful selected features was a key concern of this research, so feature selection was focused on as a method for reducing the dimensions of dataset because the subset of features from feature selection will be useful for developing decision support systems. The class imbalance issue was also posed in Chapter 1, section 1.1.4; resampling techniques were used to solve this problem (section 5.3). Over-sampling and under-sampling were applied in this thesis to assess the performance of classification. The distance-based random under-sampling that used in this thesis is proposed in section 5.3.2.2. In Chapter 7, the results showed that after balancing the dataset, performance increased; especially recall (sensitivity) values. The performance of these was validated on clinical datasets by comparison of different classifiers.

Objective 4: To develop a new method for selecting the significant variables by integrating two techniques of dimensionality reduction, namely, feature extraction and feature selection.

The common way to locate significant original features is based on loadings (principal component (PC)) in PCA. For any factor, high loadings (PC) in absolute value indicate that the corresponding variables contribute more than other variables (Guo *et al.*, 2002). A method was proposed to select a subset of features by building-up from this idea; the first PC is selected that preserves as much of the information present in the complete data as possible. The optimal subset of features is obtained by applying symmetrical uncertainty (SU) to select the original variables having high association with the PC. In order to avoid classification bias from redundancy, mutual information (MI) is applied to optimise the optimal subset of features by removing the redundant features.

The proposed method (FS-PPC) was evaluated on a real heart failure dataset (LIFELAB) that was discussed in Chapter 1, and also was assessed on clinical datasets from the UCI that were presented in Table 7.17 from Chapter 7. The results showed that the proposed feature selection method successfully identified an optimal subset of features. The subset from FS-PPC led to a subset of features with less redundancy than other studied feature selection methods. Hence, it can be claimed that FS-PPC selects a reliable subset for classification by reducing the bias from redundant features.

8.3 Summary and future research

Having discussed the contributions this research has made to the current state-of-the-art in data mining in clinical datasets, we would like to look at the limitations of the work, and promising avenues for future research. This thesis has examined the requirements and problem of data mining in clinical dataset, and explored how to produce a suitable feature selection process to select significant variables in a clinical dataset.

The work in this thesis builds on the basic idea of uniting concepts from feature extraction and feature selection to reduce the numbers of dimensions of datasets (Foldiak, 1989; Kramer, 1991; Yu and Liu, 2004; Sa *et al.*, 2007). The main focuses of the research were increasing the performance and the reliability of classification in the clinical data. The optimal set of features that results will be useful for decision support systems. In light of the findings presented in this thesis and the conclusions drawn in section 8.2, specific contributions to the area of handling complexities of clinical data by applying data mining techniques are as follows:

(1) The definition of a dataset issue to minimise the problem.

(2) The adaptation of a data mining framework to quantify the data issues.

(3) The implementation of data mining techniques to solve the problem of data, e.g. missing values, imbalanced classes and high dimensionality.

(4) The definition of an accuracy measure to quantify the target class.

(5) Assessment of the performance of data mining techniques by classification.

(6) Selection of an optimal set of features that can be used for decision support systems.

In the wider context of feature selection, the proposed method has to deal with a large number of features (variables) and redundancies among the features. This makes it more difficult to reveal the optimal features in clinical datasets. In this thesis, we develop methods that allow us to better understand the structure of a large set of clinical data. Principal component (Kramer, 1991; Tabachnick and Fidell, 1996; Jolliffe, 2005; Zabiri *et al.*, 2009) has been used as a part of this method because it uses all the original features to generate the new features and discards redundant features. Thus, feature extraction is an approach to visualization that extracts important regions or objects of interest algorithmically from a large dataset

(Reinders *et al.*, 1998) but cannot explicitly eliminate irrelevant features (Menze *et al.*, 2009). The reduced dimensionality of the data set is essentially one that yields new (and fewer) dimensions than before. These new dimensions do not necessarily carry any meaning, nor can they be directly associated with the variables of the dataset. On the other hand, feature selection also reduces the dimensions, but retains the labels associated with the variables, so in a sense the new set of features is a subset of the original set of features. Both the categories of techniques are used frequently; for example in image processing feature extraction is a popular technique, while, where it is important that labels are retained for the features (e.g. clinical systems) feature selection is a dominant technique for reduction of dimensions. For interpretation purposes or future investigations, feature selection can be achieved by choosing informative features (variables). This method, feature selection by projecting onto principal component (FS-PPC) is presented that applies a PC with SU to find the subset of features with high association and then eliminates redundant features by applying MI, an algorithm listed in Chapter 6, section 6.3.4.

Although the research in this thesis contributes ideas and techniques to the field of data analysis, like any research, it provides scope for further work. In Chapter 1, it was noted that the exploratory and investigative nature of clinical data naturally poses issues. The idea of using feature selection in a clinical dataset was proposed in Chapter 6. The performance accuracy of this method, were evaluated by classification, was slightly decreased, as with other feature selection methods. Hence, future work will attempt to improve the performance of classification and to work more effectively on different domains of data and multi-classes. Herein we have concluded that feature selection is one of the most useful tools for developing the prediction model because decision support systems require meaningful and significant features to make a decision in order to create an effective model. The implemented algorithms for feature selection will be used as a predictor in a prognosis model for a decision support system. The most important path is to design and create the appropriate predictive model and it is essential to think of what features should be selected and their precision and accuracy. Therefore, the challenge of uncertainty in clinical data will arise, to be handled by a probabilistic approach.

BIBLIOGRAPHY

Abdala, O. T. & Saeed, M. 2004 Estimation of missing values in clinical laboratory measurements of ICU patients using a weighted K-nearest neighbors algorithm. Computers in Cardiology, 2004, 19-22 Sept., vol. 31**,** pp. 693-696.

Acuna, E. & Rodriguez, C. 2004. The treatment of missing values and its effect in the classifier accuracy. *Classification, Clustering and Data Mining Applications,* vol.**,** pp. 639-648.

Addison, D., Wermter, S. & Arevian, G. 2003 A comparison of feature extraction and selection techniques. Proceedings of the Proceedings of International Conference on Artificial Neural Networks (Supplementary Proceedings). pp. 212-215.

Afzal, Z., Schuemie, M., van Blijderveen, J., Sen, E., Sturkenboom, M. & Kors, J. 2013. Improving sensitivity of machine learning methods for automated case identification from free-text electronic medical records. *BMC Medical Informatics and Decision Making,* vol. 13, 1**,** pp. 30.

Aha, D. & Bankert, R. 1996. A comparative evaluation of sequential feature selection algorithms. *In:* Fisher, D. & Lenz, H.-J. (eds.) *Learning from Data: Artificial Intelligence and Statistics V.* Springer-Verlag, vol. pp. 199-206.

Alcalá-Fdez, J., Sánchez, L., García, S., Del, Ventura, S., Garrell, J., Otero, J., Romero, C., Bacardit, J., Rivas, V., Fernández, J. & Herrera, F. 2009. KEEL:

a software tool to assess evolutionary algorithms for data mining problems. *Soft Computing - A Fusion of Foundations, Methodologies and Applications,* vol. 13, 3, pp. 307-318.

Ali, S. I. & Shahzad, W. 2012. A Feature Subset Selection Method based on Symmetric Uncertainty and Ant Colony Optimization. *Emerging Technologies (ICET), 2012 International Conference on,* vol. 60, 11, pp. 1-6.

Altınçay, H. & Ergün, C. 2004. Clustering Based Under-Sampling for Improving Speaker Verification Decisions Using AdaBoost. *In:* Fred, A., Caelli, T., Duin, R. W., Campilho, A. & de Ridder, D. (eds.) *Structural, Syntactic, and Statistical Pattern Recognition.* Springer Berlin Heidelberg, vol. 3138, pp. 698-706.

Anand, S. S., Bell, D. A. & Hughes, J. G. 1996. EDM: A general framework for Data Mining based on Evidence Theory. *Data & Knowledge Engineering,* vol. 18, 3, pp. 189-223.

Autio, L., Juhola, M. & Laurikkala, J. 2007. On the neural network classification of medical data and an endeavour to balance non-uniform data sets with artificial data extension. *Computers in Biology and Medicine,* vol. 37, 3, pp. 388-397.

Azevedo, A. & Santos, M. F. 2008 KDD, SEMMA and CRISP-DM: a parallel overview. *In:* Abraham, A., ed. Proceedings of the Proceedings of the IADIS European Conference on Data Mining 2008, July 24-26 Amsterdam, The Netherlands. IADIS, pp. 182-185.

Bardhan, I. R. & Thouin, M. F. 2013. Health information technology and its impact on the quality and cost of healthcare delivery. *Decision Support Systems,* vol. 55, 2, pp. 438-449.

Barnard, J. & Meng, X. L. 1999. Applications of multiple imputation in medical studies: from AIDS to NHANES. *Stat Methods Med Res,* vol. 8, 1, pp. 17-36.

Batista, G. E. A. P. A. & Monard, M. C. 2003. An Analysis of Four Missing Data Treatment Methods for Supervised Learning. *Applied Artificial Intelligence,* vol. 17, 5-6, pp. 519-533.

Battiti, R. 1994. Using mutual information for selecting features in supervised neural net learning. *Neural Networks, IEEE Transactions on,* vol. 5, 4, pp. 537-550.

Biau, G., Devroye, L. & Lugosi, G. 2008. Consistency of Random Forests and Other Averaging Classifiers. *Journal of Machine Learning Research,* vol. 9, pp. 2015-2033.

Blake, C. L. & Merz, C. J. 1998. *UCI repository of machine learning databases* [Online]. University of California, Irvine. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html/ [Accessed 14 October 2013].

Bonney, W. 2011. *Impacts and Risks of Adopting Clinical Decision Support Systems*: InTech.

Borovicka, T., Jirina, M., Jr. & Kordik, P. 2012. *Selecting Representative Data Sets*.

Breiman, L. 1996. Bagging Predictors. *Mach Learn,* vol. 24, 2, pp. 123 - 140.

Breiman, L. 2001. Random Forests. *Machine Learning,* vol. 45, 1, pp. 5-32.

Breiman, L. 2004. Consistency for a Simple Model of Random Forests,. Statistics Department, University of California Berkeley.

Brown, G. 2009 A New Perspective for Information Theoretic Feature Selection. Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics. pp.

Burez, J. & Poel, D. V. d. 2009. Handling class imbalance in customer churn prediction. *Expert System Application,* vol. 36, 3, pp. 4626-4636.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C. & Wirth, R. 2000. CRISP-DM 1.0 Step-by-step data mining guide. SPSS Inc.

Chawla, N., Japkowicz, N. & Kotcz, A. 2004. Editorial: special issue on learning from imbalanced data sets. *SIGKDD Explor. Newsl.,* vol. 6, 1, pp. 1-6.

Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. 2002. SMOTE: synthetic minority over-sampling technique. *J. Artif. Int. Res.,* vol. 16, 1, pp. 321-357.

Chen, T., Cao, Y., Zhang, Y., Liu, J., Bao, Y., Wang, C., Jia, W. & Zhao, A. 2013. Random Forest in Clinical Metabolomics for Phenotypic Discrimination and Biomarker Selection. *Evidence-Based Complementary and Alternative Medicine,* vol. 2013, pp. 11.

Cheong Hee, P., Haesun, P. & Pardalos, P. 2004 A comparative study of linear and nonlinear feature extraction methods. Proceedings of the Data Mining, 2004. ICDM '04. Fourth IEEE International Conference on, 1-4 November. pp. 495-498.

Choi, E. & Lee, C. 2003. Feature extraction based on the Bhattacharyya distance. *Pattern Recognition,* vol. 36, 8, pp. 1703-1709.

Chow, T. W. S., Piyang, W. & Ma, E. W. M. 2008. A New Feature Selection Scheme Using a Data Distribution Factor for Unsupervised Nominal Data. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. 38, 2, pp. 499-509.

Chun-Nan, H., Hung-Ju, H. & Dietrich, S. 2002. The ANNIGMA-wrapper approach to fast feature selection for neural nets. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. 32, 2, pp. 207-212.

Cleland, J. G., McMurray, J. J., Kjekshus, J., Cornel, J. H., Dunselman, P., Fonseca, C., Hjalmarson, A., Korewicki, J., Lindberg, M., Ranjith, N., van Veldhuisen, D. J., Waagstein, F., Wedel, H. & Wikstrand, J. 2009. Plasma concentration of amino-terminal pro-brain natriuretic peptide in chronic heart failure: prediction of cardiovascular events and interaction with the effects of rosuvastatin: a report from CORONA (Controlled Rosuvastatin Multinational Trial in Heart Failure). *Journal of the American College of Cardiology,* vol. 54, 20, pp. 1850-9.

Cleland, J. G. F., Gemmell, I., Khand, A. & Boddy, A. 1999. Is the prognosis of heart failure improving? *European Journal of Heart Failure,* vol. 1, 3, pp. 229-241.

Coetzee, F. 2005. Correcting the Kullback-Leibler distance for feature selection. *Pattern Recognition Letters,* vol. 26, 11, pp. 1675-1683.

Collins, L. M., Schafer, J. L. & Kam, C. M. 2001. A comparison of inclusive and restrictive strategies in modern missing data procedures. *Psychol Methods,* vol. 6, 4**,** pp. 330-51.

Committee for Medicinal Products for Human Use 2009. Guideline on Missing Data in Confirmatory Clinical Trials. pp.

Cortes, C. & Vapnik, V. 1995. Support-Vector Networks. *Machine Learning,* vol. 20, 3**,** pp. 273-297.

Crone, S. F. & Finlay, S. 2012. Instance sampling in credit scoring: An empirical study of sample size and balancing. *International Journal of Forecasting,* vol. 28, 1**,** pp. 224-238.

Cruz-Barbosa, R., Bautista-Villavicencio, D. & Vellido, A. 2011 Comparative diagnostic accuracy of linear and nonlinear feature extraction methods in a neuro-oncology problem. Proceedings of the Third Mexican conference on Pattern recognition, Cancun, Mexico, vol.**,** pp. 34-41.

Cunningham, P. 2008. Dimension Reduction. *In:* Cord, M. & Cunningham, P. (eds.) *Machine Learning Techniques for Multimedia.* Berlin Heidelberg: Springer, vol. pp. 91-112.

Dag, H., Sayin, K. E., Yenidogan, I., Albayrak, S. & Acar, C. 2012 Comparison of feature selection algorithms for medical data. Proceedings of the Innovations in Intelligent Systems and Applications (INISTA), 2012 International Symposium on, 2-4 July 2012. pp. 1-5.

David, J. M. & Balakrishnan, K. 2010. Significance of Classification Techniques in prediction of Learning Disabilities. *International Journal of Artificial Intelligence & Applications,* vol. 1, 4**,** pp. 111.

Dempster, A. P., Laird, N. M. & Rubin, D. B. 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological),* vol. 39, 1**,** pp. 1-38.

Diaz, U. & Alvarez, A. 2006. Gene Selection and Classification of Microarray Data Using Random Forest. *BMC Bioinformatics,* vol. 7**,** pp. 3.

Drummond, C. & Holte, R. 2003 C4.5, Class Imbalance, and Cost Sensitivity: Why Under-sampling beats Over-sampling. Proceedings of the Workshop on Learning from Imbalanced Datasets II, ICML, Washington DC, USA. pp. 1-8.

Eaton, C., DeRoos, D., Deutsch, T., Lapis, G. & Zikopoulos, P. 2012. *Understanding big data : analytics for enterprise class {Hadoop} and streaming data*: McGraw-Hill.

Fawcett, T. 2006. An introduction to ROC analysis. *Pattern Recognition Letters,* vol. 27, 8**,** pp. 861-874.

Fayyad, U. & Irani, K. 1993 Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning. Proceedings of the 13th International Joint Conference on Artificial Intelligence, Chambery, France, vol.**,** pp. 1022-1027.

Fernández, A., Luengo, J., Derrac, J., Alcalá-Fdez, J. & Herrera, F. 2009. *Implementation and Integration of Algorithms into the KEEL Data-Mining*

Software Tool Intelligent Data Engineering and Automated Learning - IDEAL 2009: Springer Berlin / Heidelberg.

Fodor, I. 2002. A Survey of Dimension Reduction Techniques.

Foldiak, P. 1989. Adaptive network for optimal linear feature extraction. *International Joint Conference on Neural Networks (IJCNN),* vol.**,** pp. 401-405 vol.1.

Ford, J. B., Roberts, C. L., Algert, C. S., Bowen, J. R., Bajuk, B. & Henderson-Smart, D. J. 2007. Using hospital discharge data for determining neonatal morbidity and mortality: a validation study. *BMC Health Services Research,* vol. 7**,** pp. 188.

Forman, G. 2003. An Extensive Empirical Study of Feature Selection Metrics for Text Classification. *Journal of Machine Learning Research,* vol. 3**,** pp. 1289-1305.

Fox, J., Glasspool, D., Patkar, V., Austin, M., Black, L., South, M., Robertson, D. & Vincent, C. 2010. Delivering clinical decision support services: There is nothing as practical as a good theory. *Journal of Biomedical Informatics,* vol. 43, 5**,** pp. 831-843.

Garcia, S. & Herrera, F. 2009. Evolutionary undersampling for classification with imbalanced datasets: proposals and taxonomy. *Evol Comput,* vol. 17, 3**,** pp. 275-306.

Gardner, M. W. & Dorling, S. R. 1998. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric Environment,* vol. 32, 14–15**,** pp. 2627-2636.

Gentle, J. E. & Hardle, W. 2004. *Handbook of Computational Statistics*: Springer.

Gibbons, J. & Chakraborti, S. 2003. *Nonparametric Statistical Inference (Statistics: a Series of Textbooks and Monogrphs),* New York: CRC Press.

Gilchrist, J., Townsend, D., Ennett, C. M., Frize, M. & Bariciak, E. 2008 Discrimination of Inconsistencies in Medical Data. MeMeA 2008 -IEEE International Workshop on Medical Measurements and Applications, 9-10 May Ottawa, Ontario, Canada, vol.**,** pp. 87-92.

Groves, P., Kayyali, B., Knott, D. & Kuiken, S. V. 2013. The 'big data' revolution in healthcare: Accelerating value and innovation. New Jersey: Center for US Health System Reform Business Technology Office.

Guo, B., Damper, R. I., Gunn, S. R. & Nelson, J. D. B. 2008. A fast separability-based feature-selection method for high-dimensional remotely sensed image classification. *Pattern Recognition,* vol. 41, 5**,** pp. 1653-1662.

Guo, Q., Wu, W., Massart, D. L., Boucon, C. & de Jong, S. 2002. Feature selection in principal component analysis of analytical data. *Chemometrics and Intelligent Laboratory Systems,* vol. 61, 1–2**,** pp. 123-132.

Gupta, A., Kumar, N. & Bhatnagar, V. 2005. Analysis of Medical Data using Data Mining and Formal Concept Analysis. *Proceedings of World Academy of Science, Engineering and Technology, Vol 6,* vol.**,** pp. 253-256.

Gupta, A. & Sharda, R. 2013. Improving the science of healthcare delivery and informatics using modeling approaches. *Decision Support Systems,* vol. 55, 2**,** pp. 423-427.

Hall, M. A. & Smith, L. A. 1999. Feature Selection for Machine Learning: Comparing a Correlation based Filter Approach to the Wrapper. *the Twelfth International Florida Artificial Intelligence Research Society Conference,* vol.**,** pp. 219-223.

Han, J., Kamber, M. & Pei, J. 2012. *Data mining: concepts and techniques,* USA: Morgan Kaufmann.

Hardin, J. M. & Chhieng, D. 2007. Data Mining and Clinical Decision Support Systems. *In:* Berner, E. (ed.) *Clinical Decision Support Systems.* Springer New York, vol. pp. 44-63.

Hastie, T. & Tibshirani, R. 1998 Classification by pairwise coupling. Proceedings of the 1997 conference on Advances in neural information processing systems 10, Denver, Colorado, USA, vol.**,** pp. 507-513.

He, H. & Garcia, E. A. 2009. Learning from Imbalanced Data. *IEEE Trans. on Knowl. and Data Eng.,* vol. 21, 9**,** pp. 1263-1284.

Ho, T. & Basu, M. 2002. Complexity measures of supervised classification problems. *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 24, 3**,** pp. 289-300.

Honghai, F., Guoshun, C., Cheng, Y., Bingru, Y. & Yumei, C. 2005 A SVM regression based approach to filling in missing values. Proceedings of the 9th international conference on Knowledge-Based Intelligent Information and Engineering Systems - Volume Part III, Melbourne, Australia, vol.**,** pp. 581-587.

Hunt, R., Johnston, M., Browne, W. & Zhang, M. 2011. Sampling Methods in Genetic Programming for Classification with Unbalanced Data. *In:* Li, J. (ed.) *AI 2010: Advances in Artificial Intelligence.* Springer Berlin Heidelberg, vol. 6464**,** pp. 273-282.

IBM 2011. IBM Big Data Success Stories. USA: IBM Corporation.

Jaeger, J., Sengupta, R. & Ruzzo, W. L. 2003. Improved gene selection for classification of microarrays. *Pacific Symposium on Biocomputing,* vol.**,** pp. 53-64.

Jagannathan, R. & Petrovic, S. 2009 Dealing with missing values in a clinical case-based reasoning system. Proceedings of the ICCSIT 2009 2nd IEEE International Conference on Computer Science and Information Technology, 8-11 August. pp. 120-124.

Janecek, A. G., Gansterer, W. N., Demel, M. A. & Ecker, G. F. 2008 On the Relationship between Feature Selection and Classification Accuracy. Proceedings of the JMLR: Workshop and Conference. pp. 90-105.

Jinxiu, C., Donghong, J., Chew Lim, T. & Zhengyu, N. 2005 Unsupervised Feature Selection for Relation Extraction. Proceedings of the.: International Joint Conference on Natural Language Processing, pp. 262-267.

Jirapech-Umpai, T. & Aitken, S. 2005. Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes. *BMC Bioinformatics,* vol. 6, 1**,** pp. 148.

Joachims, T. 2006 Training linear SVMs in linear time. Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, Philadelphia, PA, USA, vol.**,** pp. 217-226.

Jolliffe, I. 2005. Principal Component Analysis. *Encyclopedia of Statistics in Behavioral Science.* John Wiley & Sons, Ltd, vol. pp.

Jolliffe, I. T. 2002. Principal Component Analysis and Factor Analysis. *Principal Component Analysis.* New York: Springer, vol. pp. 150-166.

Juhola, M. & Laurikkala, J. 2013. Missing values: how many can they be to preserve classification reliability? *Artificial Intelligence Review,* vol. 40**,** pp. 231-245.

Kaiser, H. 1960. The Application of Electronic Computers to Factor Analysis. *Educational and Psychological Measurement,* vol. 20, 1**,** pp. 141-151.

Kalton, G. & Kasprzyk, D. 1982. Imputing for missing survey responses. *Proceedings of the section on Survey Research Methods,* vol.**,** pp. 22-31.

Kamath, C. 2009. The Scientific Data Ming Process. *Scientific Data Mining: a practical perspective.* PA, USA: Society for Industrial and Applied Mathematics.

Ketchum, E. S., Moorman, A. J., Fishbein, D. P., Mokadam, N. A., Verrier, E. D., Aldea, G. S., Andrus, S., Kenyon, K. W. & Levy, W. C. 2010. Predictive value of the Seattle Heart Failure Model in patients undergoing left ventricular assist device placement. *J Heart Lung Transplant,* vol. 29, 9**,** pp. 1021-5.

Kibriya, A. & Frank, E. 2007. *An Empirical Comparison of Exact Nearest Neighbour Algorithms,* Berlin Heidelberg: Springer

Kohavi, R. & John, G. H. 1997. Wrappers for feature subset selection. *Artificial Intelligence,* vol. 97, 1–2**,** pp. 273-324.

Kotani, M., Nakai, M. & Akazawa, K. 1999 Feature extraction using evolutionary computation. Proceedings of the Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on, 1999. pp. 1236 Vol. 2.

Kramer, M. 1991. Nonlinear Principal Component Analysis Using Autoassociative Neural Networks. *AIChE Journal,* vol. 37, 2**,** pp. 233-243.

Levner, I. 2005. Feature selection and nearest centroid classification for protein mass spectrometry. *BMC Bioinformatics,* vol. 6, 68**,** pp. 1-14.

Levy, W. C., Mozaffarian, D., Linker, D. T., Sutradhar, S. C., Anker, S. D., Cropp, A. B., Anand, I., Maggioni, A., Burton, P., Sullivan, M. D., Pitt, B., Poole-Wilson, P. A., Mann, D. L. & Packer, M. 2006. The Seattle Heart Failure Model: prediction of survival in heart failure. *Circulation,* vol. 113, 11**,** pp. 1424-33.

Li, D., Deogun, J. S., Spaulding, W. & Shuart, B. 2004 Towards Missing Data Imputation: A Study of Fuzzy K-means Clustering Method. Proceedings of the Rough Sets and Current Trends in Computing, 4th International Conference, RSCTC 2004, Sweden. Springer, pp. 573-579.

Li, S., Liao, C. & Kwok, J. T. 2006 Gene feature extraction using T-test statistics and kernel partial least squares. Proceedings of the 13th international conference on Neural information processing - Volume Part III, Hong Kong, China, vol., pp. 11-20.

Liang, G. & Zhang, C. 2012 An efficient and simple under-sampling technique for imbalanced time series classification. Proceedings of the 21st ACM international conference on Information and knowledge management, Maui, Hawaii, USA, vol., pp. 2339-2342.

Liao, C., Li, S. & Luo, Z. 2006. Gene Selection for Cancer Classification using Wilcoxon Rank Sum Test and Support Vector Machine. *Computational Intelligence and Security.* Springer Berlin Heidelberg, vol. 4456, pp. 57-66.

Lin, J. H. & Haug, P. J. 2006. Data preparation framework for preprocessing clinical data in data mining. *AMIA Annual Symposium Proceedings,* vol., pp. 489-93.

Little, R. J. A. & Rubin, D. B. 1978. *Statistical Analysis with Missing Data,* New York: John Wiley & Sons.

Liu, H., Li, J. & Wong, L. 2002. A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns. *Genome Informatics,* vol. 13, pp. 51-60.

Lu, Y., Cohen, I., Zhou, X. S. & Tian, Q. 2007 Feature selection using principal feature analysis. Proceedings of the 15th international conference on Multimedia, Augsburg, Germany, vol., pp. 301-304.

Luengo, J., García, S. & Herrera, F. 2011. On the choice of the best imputation methods for missing values considering three groups of classification methods. *Knowledge and Information Systems,* vol. 32, 1**,** pp. 77-108.

Mallinson, H. & Gammerman, A. 2005. Evaluation of Support Vector Machines for Imputation. Department of Computer Science, Royal Holloway, University of London.

Maloof, M. A. 2003 Learning when data sets are Imbalanced and when costs are unequal and unknown. Workshop on Learning from Imbalanced Data Sets II, ICML, Washington DC, 2003., vol.**,** pp.

Martin, J. K. & Hirschberg, D. S. 1996 On the Complexity of Learning Decision Trees. the 4th International Symposium on Artificial Intelligence and Mathematics (AI/MATH96), January 3-5 Florida, USA, vol.**,** pp. 112-115.

Mease, D., Wyner, A. J. & Buja, A. 2007. Boosted Classification Trees and Class Probability/Quantile Estimation. *Journal of Machine Learning Research,* vol. 8**,** pp. 409-439.

Menze, B., Kelm, B. M., Masuch, R., Himmelreich, U., Bachert, P., Petrich, W. & Hamprecht, F. 2009. A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data. *BMC Bioinformatics,* vol. 10, 1**,** pp. 213.

Ming-Syan, C., Jiawei, H. & Yu, P. S. 1996. Data mining: an overview from a database perspective. *Knowledge and Data Engineering, IEEE Transactions on,* vol. 8, 6**,** pp. 866-883.

Mitra, P., Murthy, C. A. & Pal, S. K. 2002. Unsupervised feature selection using feature similarity. *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 24, 3**,** pp. 301-312.

N, N. M., Abdullah, M. M. A., Tan, C.-y., Ramli, N. A., Yahaya, A. S. & Fitri, N. F. M. Y. 2011. Modelling of PM10 concentration for industrialized area in Malaysia: A case study in Shah Alam. *Physics Procedia,* vol. 22, 0**,** pp. 318-324.

NHS 2010. National Heart Failure Audit 2010. *Third report for the audit period between April 2009 and March 2010.* Leeds: NHS Information Centre for Health and Social Care.

Novakovic, J. 2009 Using Information Gain Attribute Evaluation to Classify Sonar Targets 17th Telecommunications forum TELFOR 2009, Serbia, Belgrade, vol.**,** pp. 1351 - 1354.

Olson, D. L. & Delen, D. 2008. *Advanced Data Mining Techniques,* Heidelberg: Springer Publishing Company, Incorporated.

Omid, M. 2011. Design of an expert system for sorting pistachio nuts through decision tree and fuzzy logic classifier. *Expert Syst. Appl.,* vol. 38, 4**,** pp. 4339-4347.

Palaniappan, S. & Awang, R. 2008. Intelligent heart disease prediction system using data mining techniques. *2008 Ieee/Acs International Conference on Computer Systems and Applications, Vols 1-3,* vol.**,** pp. 108-115.

Pang, H., Lin, A., Holford, M., Enerson, B. E., Lu, B., Lawton, M. P., Floyd, E. & Zhao, H. 2006. Pathway analysis using random forests classification and regression. *Bioinformatics,* vol. 22, 16**,** pp. 2028-2036.

Paredes, S., Rocha, T., de Carvalho, P., Henriques, J., Harris, M. & Morais, J. 2009. Long term cardiovascular risk models' combination - a new approach. *31st Annual International Conference of the IEEE EMBS,* vol. 4, 10**,** pp. 4711-4714.

Platt, J. C. 1999. Fast training of support vector machines using sequential minimal optimization. *Advances in kernel methods.* MA, USA: MIT Press, vol. pp. 185-208.

Polat, K. & Güneş, S. 2007. A hybrid approach to medical decision support systems: Combining feature selection, fuzzy weighted pre-processing and AIRS. *Computer Methods and Programs in Biomedicine,* vol. 88, 2**,** pp. 164-174.

Poolsawad, N. & Kambhampati, C. 2014. Issues in the mining of heart failure datasets. *International Journal of Automation and Computing,* vol. 11, 2**,** pp. 162-179.

Poolsawad, N., Kambhampati, C. & Cleland, J. G. F. 2011 Feature Selection Approaches with Missing Values Handling for Data Mining - A Case Study of Heart Failure. Proceedings of the International Conference on Data Mining (ICDM 2011), Phuket, Thailand. World Academy of Science, Engineering and Technology, pp. 828-836.

Poolsawad, N., Moore, L., Kambhampati, C. & Cleland, J. G. F. 2012a Handling Missing Values in Data Mining - A Case Study of Heart Failure Dataset.

Proceedings of the The 2012 8th International Conference on Natural Computation (ICNC'12) and the 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'12), Chongqing, China. the Institute of Electrical and Electronics Engineers, pp. 2946-2950.

Poolsawad, N., Moore, L., Kambhampati, C. & Cleland, J. G. F. 2012b Performance Metrics for Classification in Clinical Dataset. the 19th International Conference on Neural Information Processing (ICONIP2012), Doha, Qatar, vol., pp.

Powers, D. 2007. Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation. Adelaide, Australia: School of Informatics and Engineering, Flinders University of South Australia

Prati, R., Batista, G. A. P. A. & Monard, M. 2004. Learning with Class Skews and Small Disjuncts. *In:* Bazzan, A. C. & Labidi, S. (eds.) *Advances in Artificial Intelligence - SBIA 2004.* Berlin Heidelberg: Springer vol. 3171, pp. 296-306.

Qi, C. & Li, H.-X. 2009. Nonlinear dimension reduction based neural modeling for distributed parameter processes. *Chemical Engineering Science,* vol. 64, 19, pp. 4164-4170.

Quinlan, J. R. 1986. Induction of decision trees. *Machine Learning,* vol. 1, 1, pp. 81-106.

Quinlan, J. R. 1992 Learning with Continuous {C}lasses. Proceedings of the 5th Australian Joint Conference on Artificial Intelligence. pp. 343-348.

Quinlan, J. R. 1996. Improved use of continuous attributes in C4.5. *J. Artif. Int. Res.,* vol. 4, 1, pp. 77-90.

Rahman, M. M. & Davis, D. N. 2013 Cluster Based Under-Sampling for Unbalanced Cardiovascular Data. Proceedings of the World Congress on Engineering 2013, July 3-5, 2013 London, UK. pp.

Ramaswami, M. & Bhaskaran, R. 2009. A Study on Feature Selection Techniques in Educational Data Mining *Journal of Computing,* vol. 1, 1, pp. 7-11.

Ramesh, A. N., Kambhampati, C., Monson, J. R. & Drew, P. J. 2004. Artificial intelligence in medicine. *Annals of the Royal College of Surgeons of England,* vol. 86, 5, pp. 334-338.

Rees, A. M. 1997. *Consumer Health USA: Heart Disease and Blood Vessel Disorders,* Arizona: The Oryx Press.

Reinders, F., Spoelder, H. W. & Post, F. 1998. Experiments on the Accuracy of Feature Extraction. *In:* Bartz, D. (ed.) *Visualization in Scientific Computing '98.* Springer Vienna, vol. pp. 49-58.

Reyes-Aldasoro, C. C. & Bhalerao, A. 2006. The Bhattacharyya space for feature selection and its application to texture segmentation. *Pattern Recognition,* vol. 39, 5, pp. 812-826.

Rubin, D. 1987. *Multiple Imputation for Nonresponse in Surveys (Wiley Series in Probability and Statistics)*: Wiley.

Rubin, D. B. 1976. Inference and missing data. *Biometrika,* vol. 63, 3, pp. 581-592.

Rubin, D. B. 1988. On Overview of Multiple Imputation. *Proceedings of the Survey Research Methods Section.* American Statistical Association, vol. pp. 79-84.

Rutanen, K., Germán, G.-H., Eriksson, S.-L. & Egiazarian, K. 2013. A general definition of the big-oh notation for algorithm analysis. *ArXiv e-prints* [Online]. Available: http://adsabs.harvard.edu/abs/2013arXiv1309.3210R [Accessed 25/11/2013].

Sa, W., Cheng-Lin, L. & Lian, Z. 2007 Feature Selection by Combining Fisher Criterion and Principal Feature Analysis. Proceedings of the Machine Learning and Cybernetics, 2007 International Conference on, 19-22 Aug. 2007. pp. 1149-1154.

SAS Institute Inc. Data Mining and the Case for Sampling - A SAS Institute Best Practices Paper Solving Business Problems Using SAS&reg; Enterprise Miner&trade; Software. SAS Institute Inc.

Schafer, J. & Olsen, M. 1998. Multiple Imputation for Multivariate Missing-Data Problems: A Data Analyst's Perspective. *Multivariate Behavioral Research,* vol. 33, 4**,** pp. 545-571.

Schafer, J. L. 1997. *Analysis of incomplete multivariate data,* Florida: Chapman & Hall.

Scholz, M. 2012. Validation of Nonlinear PCA. *Neural Processing Letters,* vol. 36, 1**,** pp. 21-30.

Schwartzman, A., Wolf, T., Gepstein, L., Hayam, G., Lessick, J., Reisfeld, D., Schwartz, Y., Uretzky, G. & Ben-Haim, S. A. 2001. Characterisation of acute myocardial ischaemia in a canine model based on principal component analysis of unipolar endocardial electrograms. *Med Biol Eng Comput,* vol. 39, 5, pp. 571-8.

Senliol, B., Gulgezen, G., Lei, Y. & Cataltepe, Z. 2008 Fast Correlation Based Filter (FCBF) with a different search strategy. Proceedings of the Computer and Information Sciences, 2008. ISCIS '08. 23rd International Symposium on, 27-29 Oct. 2008. pp. 1-4.

Seo, M. & Oh, S. 2012. CBFS: High Performance Feature Selection Algorithm Based on Feature Clearness. *Plos One,* vol. 7, 7, pp. e40419.

Sittig, D. F., Wright, A., Osheroff, J. A., Middleton, B., Teich, J. M., Ash, J. S., Campbell, E. & Bates, D. W. 2008. Grand challenges in clinical decision support. *Journal of Biomedical Informatics,* vol. 41, 2, pp. 387-392.

Sivasankar, E. & Rajesh, R. S. 2012 Design and development of a clinical decision support system for diagnosing appendicitis. Proceedings of the Computing, Communications and Applications Conference (ComComAp), 2012, 11-13 Jan. 2012. pp. 316-321.

Søndberg-Madsen, N., Thomsen, C. & Peña, J. 2003 Unsupervised feature subset selection. Proceedings of the In Proceedings of the Workshop on Probabilistic Graphical Models for Classification. pp. 71-82.

Steinbach, M., Karypis, G. & Kumar, V. 2000 A comparison of document clustering techniques. KDD Workshop on Text Mining, August 20-23 Boston, MA, vol., pp. 1-2.

Strobl, C., Boulesteix, A., Kneib, T., Augustin, T. & Zeileis, A. 2008. Conditional variable importance for random forests. *BMC Bioinformatics,* vol. 9, pp. 307.

Su, Y., Murali, T. M., Pavlovic, V., Schaffer, M. & Kasif, S. 2003. RankGene: identification of diagnostic genes based on expression data. *Bioinformatics,* vol. 19, 12, pp. 1578-1579.

Subramonian, R. 1998a Defining diff as a data mining primitive. Proceedings of the KDD-98 Proceedings. AAAI, pp.

Subramonian, R. 1998b defining diff as a data mining primitives. Proceedings of the KDD-98, New York, USA. American Association for Artificial Intelligence (AAAI), pp.

Suzuki, K. 2011. *Artificial Neural Networks - Methodological Advances and Biomedical Applications,* Rijeka, Croatia: InTech.

Tabachnick, B. & Fidell, L. 1996. *Using Multivariate Statistics,* New York: HarperCollins College Publishers.

Tanwani, A. K. & Farooq, M. 2009. The Role of Biomedical Dataset in Classification. *Artificial Intelligence in Medicine, Proceedings,* vol. 5651, pp. 370-374.

The MathWorks Inc. *MATLAB technical documentation* [Online]. Mathworks.com. Available: http://www.mathworks.co.uk/help/matlab/math/matrix-indexing.html?s_cid=wiki_matlab_8&s_tid=doc_12b [Accessed 27/11/2013.

Theodoridis, S. & Koutroumbas, K. 2006. *Pattern Recognition,* CA, USA: Academic Press.

Touw, W. G., Bayjanov, J. R., Overmars, L., Backus, L., Boekhorst, J., Wels, M. & van Hijum, S. A. 2013. Data mining in the Life Sciences with Random Forest:

a walk in the park or lost in the jungle? *Brief Bioinform,* vol. 14, 3, pp. 315-26.

Turney, P. 2000 Types of cost in inductive concept learning. Workshop on Cost-Sensitive Learning at ICML, June 29-July 2 CA, USA, vol., pp. 15 - 21.

van der Maaten, L. J. P., Postma, E. O. & van den Herik, H. J. 2009. Dimensionality Reduction: A Comparative Review. Tilburg, The Netherlands: Tilburg centre for Creative Computing, Tilburg University.

Ververidis, D. & Kotropoulos, C. 2008. Fast and accurate sequential floating forward feature selection with the Bayes classifier applied to speech emotion recognition. *Signal Processing,* vol. 88, 12, pp. 2956-2970.

Ververidis, D. & Kotropoulos, C. 2009. Information Loss of the Mahalanobis Distance in High Dimensions: Application to Feature Selection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 31, 12, pp. 2275-2281.

Vorobieva, O., Rumyantsev, A. & Schmidt, R. 2007 A CBR Solution for Missing Medical Data. 5th Workshop on CBR in the Health Sciences, August 15 Belfast, Northern Ireland, vol., pp.

Wang, H. & Wang, S. 2010. Mining incomplete survey data through classification. *Knowledge and Information Systems,* vol. 24, 2, pp. 221-233.

Wang, K. J., Makond, B. & Wang, K. M. 2013. An improved survivability prognosis of breast cancer by using sampling and feature selection technique to solve imbalanced patient classification data. *BMC Medical Informatics and Decision Making,* vol. 13, 1, pp. 124.

Wang, L., Chu, F. & Xie, W. 2007a. Accurate Cancer Classification Using Expressions of Very Few Genes. *IEEE/ACM Transactions on Computational Biology and Bioinformatics,* vol. 4, 1, pp. 40-53.

Wang, R. & Tang, K. 2009 Feature Selection for Maximizing the Area Under the ROC Curve. Proceedings of the 2009 IEEE International Conference on Data Mining Workshops, vol., pp. 400-405.

Wang, S., Liu, C.-L. & Zheng, L. 2007b Feature Selection by Combining Fisher Criterion and Principal Feature Analysis. Proceedings of the Machine Learning and Cybernetics, 2007 International Conference on. pp. 1149-1154.

Weiss, G. M. 2004. Mining with rarity: a unifying framework. *SIGKDD Exploration Newsletter,* vol. 6, 1, pp. 7-19.

Wirth, R. & Hipp, J. 2000 CRISP-DM: Towards a standard process model for data mining. Proceedings of the the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining, Manchester, UK. pp. 29-39.

Witten, I. H. & Frank, E. 2005. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*: Morgan Kaufmann Publishers Inc.

Wright, A. & Sittig, D. F. 2008. A framework and model for evaluating clinical decision support architectures. *Journal of Biomedical Informatics,* vol. 41, 6, pp. 982-990.

Wu, B., Abbott, T., Fishman, D., McMurray, W., Mor, G., Stone, K., Ward, D., Williams, K. & Zhao, H. 2003. Comparison of statistical methods for classification of ovarian cancer using mass spectrometry data. *Bioinformatics,* vol. 19, 13**,** pp. 1636-43.

Xuan, G., Chai, P. & Wu, M. 1996 Bhattacharyya distance feature selection. Proceedings of the the 13th International Conference on Pattern Recognition, 25-29 August. pp. 195-199.

Yan-ping, Z., Li-Na, Z. & Yong-Cheng, W. 2010 Cluster-based majority under-sampling approaches for class imbalance learning. Proceedings of the Information and Financial Engineering (ICIFE), 2010 2nd IEEE International Conference on, 17-19 Sept. 2010. pp. 400-404.

Yang, F., Wang, H.-z., Mi, H., Lin, C.-d. & Cai, W.-w. 2009. Using random forest for reliable classification and cost-sensitive learning for medical diagnosis. *BMC Bioinformatics,* vol. 10, Suppl 1**,** pp. S22.

Yang, H. H. & Amari, S. 1998. Complexity issues in natural gradient descent method for training multilayer perceptrons. *Neural Comput,* vol. 10, 8**,** pp. 2137-57.

Yen, S.-J. & Lee, Y.-S. 2006. Cluster-Based Sampling Approaches to Imbalanced Data Distributions. *In:* Tjoa, A. & Trujillo, J. (eds.) *Data Warehousing and Knowledge Discovery.* Springer Berlin Heidelberg, vol. 4081**,** pp. 427-436.

Yom-Tov, E. & Inbar, G. F. 2002. Feature selection for the classification of movements from single movement-related potentials. *IEEE Trans Neural Syst Rehabil Eng,* vol. 10, 3**,** pp. 170-7.

Yu, L. & Liu, H. 2003 Feature selection for high-dimensional data: A fast correlation-based filter solution. Proceedings of the in ICML. pp. 856-863.

Yu, L. & Liu, H. 2004. Efficient Feature Selection via Analysis of Relevance and Redundancy. *Journal of Machine Learning Research,* vol. 5**,** pp. 1205-1224.

Yuansheng, Y., Haiyan, L., Xiaohui, L. & Di, M. 2010 Recursive Feature Selection Based on Minimum Redundancy Maximum Relevancy. Proceedings of the Parallel Architectures, Algorithms and Programming (PAAP), 2010 Third International Symposium on, 18-20 Dec. 2010. pp. 281-285.

Zabiri, H., Ramasamy, M. & Tehv, I. S. Y. 2009 Quantification Analysis for NLPCA-Based Stiction Diagnostic Tool. Proceedings of the Advanced Computer Control, 2009. ICACC '09. International Conference on, 22-24 January. pp. 468-472.

Zhang, J., Goode, K. M., Rigby, A., Balk, A. H. & Cleland, J. G. 2013. Identifying patients at risk of death or hospitalisation due to worsening heart failure using decision tree analysis: evidence from the Trans-European Network-Home-Care Management System (TEN-HMS) study. *Int J Cardiol,* vol. 163, 2**,** pp. 149-56.

Zhang, J. & Mani, I. 2003 KNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction. Proceedings of the ICML'2003 Workshop on Learning from Imbalanced Datasets. pp.

Zhang, Y., Kambhampati, C., Davis, D. N., Goode, K. & Cleland, J. G. F. 2012 A Comparative Study of Missing Value Imputation with Multiclass

Classification for Clinical Heart Failure Data. Proceedings of the The 2012 8th International Conference on Natural Computation (ICNC'12) and the 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'12), Chongqing, China. the Institute of Electrical and Electronics Engineers, pp. 2946-2950.

Zhao, Z. Advancing Feature Selection Research.

Zhao, Z. & Wang, L. 2010 Efficient Spectral Feature Selection with Minimum Redundancy. *In:* Fox, M. & Poole, D., eds. Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA. AAAI Press, pp.

Zheng, Y. & Kwoh, C. K. 2011. A Feature Subset Selection Method Based On High-Dimensional Mutual Information. *Entropy,* vol. 13, 4**,** pp. 860-901.

Zhou, N. & Wang, L. 2007. A modified T-test feature selection method and its application on the HapMap genotype data. *Genomics Proteomics Bioinformatics,* vol. 5, 3-4**,** pp. 242-9.