Improving Statistical Learning within Functional Genomic Experiments by means of Feature Selection

Osama Mahmoud

A thesis submitted for the degree of

Doctor of Philosophy (Ph.D.)

Department of Mathematical Sciences

University of Essex

August 2015

Dedicated to

My parents, affectionate wife and adorable children

(Seif, Malak and Salma)

Acknowledgements

It is a pleasure to thank those who helped me to make this thesis possible. Foremost, I would like to express my sincere gratitude to my supervisors Prof. Berthold Lausen and Dr. Andrew Harrison (Harry) for their continuous support during my PhD study and research. I am grateful for their patience, motivation, enthusiasm, and immense knowledge. Their guidance helped me throughout the research and thesis writing period. I could not imagine having better supervisors for my PhD study. Thanks Berthold and Harry for showing your trust in me!

There are many other people who supported me throughout my PhD. I owe my deepest gratitude to Dr. Aris Perperoglou for his help and suggestions whenever needed. Thanks Aris for your efforts to explain things clearly and simply. My sincere thanks also goes to Dr. Benjamin Hofner, Dr. Werner Adler and Dr. Andreas Mayr, biostatisticians at Institut für Medizininformatik, Biometrie und Epidemiologie, University of Erlangen, Germany, for everything I have learnt from them. Thanks to my Supervisory Board members Dr. John Ford and Dr. Haslifah Hashim who gave me valuable feedbacks and suggestions. Thanks for all the staff at the Department of Mathematical Sciences, University of Essex for their kindness and support.

I also thank Camilla Thomsen and Claire Watts (previous and current Departmental

Administrator respectively), Anne Owen (Computer Support Officer), Shauna McNally (Graduate Administrator) and Vicki Forster (General Administrator) for their help in administrative issues.

I am grateful to the Helwan University, the Egyptian Ministry of Higher Education and the Egyptian Cultural and Education Bureau (ECEB) in London for sponsoring my PhD research. I equally thank Prof. Essam Abouelkassem, Prof. Afaf El-Dash, Prof. Ibrahim Hassan, Dr. Nadia Khalifa, Dr. Sayed El-Shair and Dr. Ahmed Abdelhadi whose support will be remembered always.

I wish to thank my entire family and my family-in-law for their love and support. I particularly thank my parents, Fathy Mahmoud and Nema Saad, and my younger sister and brother, Shaimaa Mahmoud and Abdelhaleem Mahmoud, who have been a great source of motivation. Thanks also goes to friends for their prayers and emotional support.

Lastly but importantly, I thank my wife, Doaa Saeed Ibrahim, for her love and care. Thanks for being with me in hard times and for your emotional support whilst you were also busy in looking after our children. For my children, Seif, Malak and Salma Mahmoud, I would say thanks for being the coolness of my eyes.

Abstract

A Statistical learning approach concerns with understanding and modelling complex datasets. Based on a given training data, its main aim is to build a model that maps the relationship between a set of input features and a considered response in a predictive way. Classification is the foremost task of such a learning process. It has applications encompassing many important fields in modern biology, including microarray data as well as other functional genomic experiments.

Microarray technology allow measuring tens of thousands of genes (features) simultaneously. However, the expressions of these genes are usually observed in a small number, tens to few hundreds, of tissue samples (observations). This common characteristic of high dimensionality has a great impact on the learning processes, since most of genes are noisy, redundant or non-relevant to the considered learning task.

Both the prediction accuracy and interpretability of a constructed model are believed to be enhanced by performing the learning process based only on selected informative features. Motivated by this notion, a novel statistical method, named Proportional Overlapping Scores (POS), is proposed for selecting features based on overlapping analysis of gene expression data across different classes of a considered classification task. This method results in a measure, called *POS* score, of a feature's relevance to the learning task.

POS is further extended to minimize the redundancy among the selected features.

The proposed approaches are validated on several publicly available gene expression datasets using widely used classifiers to observe effects on their prediction accuracy. Selection stability is also examined to address the captured biological knowledge in the obtained results. The experimental results of classification error rates computed using the Random Forest, *k* Nearest Neighbor and Support Vector Machine classifiers show that the proposals achieve a better performance than widely used gene selection methods.

Declaration

The work submitted in this thesis is the result of my own investigation, except where otherwise stated. It has not already been accepted for any degree, and is also not being concurrently submitted for any other degree.

Abbreviations

Abbreviations	Details
Bagging	Bootstrap Aggregation
CART	Classification and Regression Trees
RF	Random Forest
*k*NN	k Nearest Neighbour
SVM	Support Vector Machine
KKT	Karush-Kuhn-Tucker conditions
C.V.	Cross Validation
LASSO	Least Absolute Shrinkage and Selection Operator
POS	Proportional Overlapping Scores method
Wil-RS	Wilcoxon Rank Sum test method
ISIS	Iteratively Sure Independent Screen method
mRMR	minimum Redundancy Maximum Relevance method
MP	MaskedPainter method
POS	Proportional Overlapping Score measure
RDC	Relative Dominant Class measure
$\|.\|$	Size of a set $\{.\}$
$\langle I \rangle$	length of the interval I

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Introduction

Statistical Learning refers to a set of approaches for constructing a predictive model based on a given dataset. It encompasses many methods including Classification Trees (Breiman 1984), Random Forest (Breiman 2001), Boosting (Freund & Schapire 1997), k Nearest Neighbour (Cover & Hart 1967) and Support Vector Machines (Cortes & Vapnik 1995). The main goal of statistical learning is to train a given set of data, training data, to model an effective prediction rule that can be then used to predict unseen/new data.

The recent revolution in functional genomic technologies leads to generate vast amount of data. Microarray, as well as other high-throughput functional genomic technologies, provide effective tools for studying thousands of genes simultaneously. The challenge of understanding these data has led to the development of new tools in statistical learning.

Classification is the foremost task of statistical learning within the biological domain (Friedman et al. 2001). For a particular classification task, microarray data are inherently noisy

since most genes are irrelevant and uninformative to the given classes (phenotypes).

Both the prediction accuracy and interpretability of a constructed classifier could be enhanced by performing the learning process based only on selected informative features. One of the main aims of gene expression analysis is to identify genes that are expressed differentially between various classes. The problem of identification of these discriminative genes for their use in classification has been investigated in many studies (Chen et al. 2014, Apiletti et al. 2012, Peng et al. 2005).

A major challenge is the problem of dimensionality; tens of thousands of genes' expressions are observed in a small number, tens to few hundreds, of observations. Given an input of gene expression data along-with observations' target classes, the problem of gene selection is to find among the entire dimensional space a subspace of genes that best characterizes the response target variable. Since the total number of subspaces with dimension not higher than r is $\sum_{i=1}^{r}\binom{P}{i}$, where P is the total number of genes, it is hard to search the subspaces exhaustively.

Alternatively, various search schemes are proposed e.g., best individual genes (Su et al. 2003), Max-Relevance and Min-Redundancy based approaches (Peng et al. 2005), Iteratively Sure Independent Screening (Fan et al. 2009) and MaskedPainter approach (Apiletti et al. 2012). Identification of discriminative genes can be based on different criteria including: p-values of statistical tests e.g. t-test or Wilcoxon rank sum test (Lausen et al. 2004, Altman et al. 1994); ranking genes using statistical impurity measures e.g. information gain, gini index and max minority (Su et al. 2003).

Here, the overlap between gene expression measures for different classes is utilized. The thesis provides a strategy that uses the information given by observations' classes as

well as expression data for detection of the differentially expressed genes between target classes. The possibility of improving a classifier performance and prediction accuracy by identifying discriminative genes that are relevant to the considered classification task is investigated.

The thesis proposes a procedure that analyses the overlap between gene expression of different classes, to identify the minimum set of genes which yield the best classification accuracy on a training set whilst avoiding the effects of outliers. Based on this procedure, a novel statistical method, named as Proportional Overlapping Scores (POS), is proposed for selecting discriminative features for a considered classification task. This method results in a measure, called *POS* score, of a feature's relevance to the classification problem.

Several widely used classifier models: Random Forest; k Nearest neighbour; Support Vector Machines, are used to evaluate the efficiency of the proposed approach in improving the learning process. POS method is validated on 12 publicly available gene expression datasets by comparison with five well-known gene selection techniques: Wilcoxon Rank Sum (Wil-RS); Minimum Redundancy Maximum Relevance (mRMR); MaskedPainter (MP); Iteratively Sure Independent Screening (ISIS). The experimental results of classification error rates computed using the considered classifiers show that POS achieves a better performance. The proposed approach with the conducted experiments have been published in Mahmoud et al. (2014a).

The POS method is further extended to minimize the redundancy among the selected features. A recursive strategy is proposed to assign a set of complementary informative genes. The scheme exploits gene masks defined by POS to identify more integrated genes in terms of their classification patterns. The proposed version, named POSr method, is

published in Mahmoud et al. (2015)

The approaches proposed in this thesis are implemented in an R-package, called '*propOverlap*', publicly available on CRAN (Mahmoud et al. 2014b).

1.2 Thesis Organization

Chapter 2 provides a background for statistical learning. It illustrates the difference between supervised and unsupervised learning and also discusses the basics of classification and regression trees (CART) and ensemble learning schemes. Detailed explanation of several classification models such as Random Forest, k Nearest Neighbour and Support Vector Machine are also provided. Finally, some methods and metrics for evaluating a classifier performance are described.

Chapter 3 illustrates different approaches for feature selection. Different categories of feature selection methods are described. The chapter also introduces the general criterion of gene expressions overlap for identifying discriminative genes.

Chapter 4 proposes a procedure for identifying the minimum subset of genes that provide the best classification accuracy on a set of given training data. The procedure provides a definition of gene mask that measure the classification power of each gene in a considered binary classification problem. This chapter also presents a novel score, *POS*, for measuring the overlapping degree between expressions of different classes. An algorithm for detecting the minimum set of genes that correctly classify the maximum number of observations avoiding outliers effect is also given. The research within this chapter has

been published (Mahmoud et al. 2014a).

Chapter 5 proposes a novel method, named 'POS', for gene selection based on the defined *POS* score along-with the minimum subset of genes. The chapter also shows the results of misclassification error rates obtained by POS using Random Forest, k Nearest Neighbour and Support Vector Machine classifiers. The results from POS are compared with the ones yielded by widely used gene selection methods such as Wilcoxon Rank Sum (Wil-RS), Minimum Redundancy Maximum Relevance (mRMR), MaskedPainter (MP), Iteratively Sure Independent Screening (ISIS). Scores of stability selections are provided for the proposed approach and the compared methods. This work has been published in Mahmoud et al. (2014a).

Chapter 6 proposes an extended version of POS method, named 'POSr', for minimizing the selection redundancy using a recursive strategy to assign a set of complementary discriminative genes. This chapter shows the misclassification error rates as well as stability scores for the proposed approach. The obtained results are compared with POS and other gene selection methods. The research within this chapter has been published in Mahmoud et al. (2015).

Chapter 7 summarises the conclusions of the thesis and suggests future directions in which this research might be extended.

1.3 Published Work

Peer-reviewed Papers:

1. Mahmoud, O., Harrison, A., Perperoglou, A., Gul, A., Khan, Z., Metodiev, M. & Lausen, B. (2014a): A feature selection method for classification within functional genomics experiments based on the proportional overlapping score, BMC Bioinformatics 15(1).

2. Mahmoud, O., Harrison, A., Gul, A., Khan, Z., Metodiev, M. & Lausen, B. (2015): Minimizing redundancy among genes selected based on the overlapping analysis, in Proceedings of the European Conference on Data Analysis, Bremen, Germany [In Press].

Published R Packages:

1. Mahmoud, O., Harrison, A., Perperoglou, A., Gul, A., Khan, Z. & Lausen, B. (2014b): propOverlap: Feature (gene) selection based on the Proportional Overlapping Scores. R package version 1.0. URL: http://CRAN.R-project.org/package=propOverlap

Chapter 2

Background for Statistical Learning

Statistical learning techniques are described as either *supervised*, *semi-supervised* or *unsupervised*. The distinction results from how the learning process identifies its training data.

2.1 Supervised vs. Unsupervised Learning

In supervised learning, training data are usually presented as (X, Y) such that $X \in \mathfrak{R}^{N \times P}$ is a feature matrix in which N observations are reported each with P features (dimensions), whilst $Y \in \mathfrak{R}^{N}$ is a vector of output labels (supervisors). Classification techniques (e.g., Classification and Regression Trees, Random Forest, k Nearest Neighbour and Support Vector Machine, presented in Sections 2.2, 2.4, 2.5 and 2.6 respectively) provide important representative examples of supervised learning.

Unsupervised learning defines the training data to contain only the feature matrix X (i.e., N observations are presented each with P features without supervised output labels Y). Clustering techniques (e.g., k means and hierarchical clustering) are the classical

representative examples for unsupervised learning.

Semi-supervised learning falls between unsupervised learning and supervised learning. It refers to a set of tasks and techniques that treat data with supervised output labels for part of it.

2.2 Classification and Regression Trees (CART)

Classification and Regression Trees (CART) have been around since Breiman et al. (1984) proposed a procedure for building trees to predict categorical and continuous response variables for classification and regression problems respectively. Many refinements of CART approach have been developed for enhancing its uses in various fields (e.g., Chipman et al. 1998, Loh 2002, Su et al. 2004). CART are considered a base classifier for most of the ensemble learning methods which are discussed in Section 2.3. They are used in many fields including statistics, applied mathematics and computer science, etc. Moreover, they are usually linked to machine and statistical learning, and data mining.

The CART approach uses the training data to construct a binary decision tree which is then used for predicting the class labels of new data (in case of classification problems) or the real-values of the response (in case of regression). This is accomplished by recursively splitting the feature space into two disjoint regions (e.g., two outcomes in the case of a binary feature (for more details, see Section 2.2.1)).

For classification problems, a set of training data (X, Y) are given. Figure 2.1 illustrates the structure of CART where X_{i_l}, $l = 1, 2$ are two features (predictors) and $B(X_{i_l})$ is a given condition associated with X_{i_l} for splitting leaf nodes. The root node definitely contains the

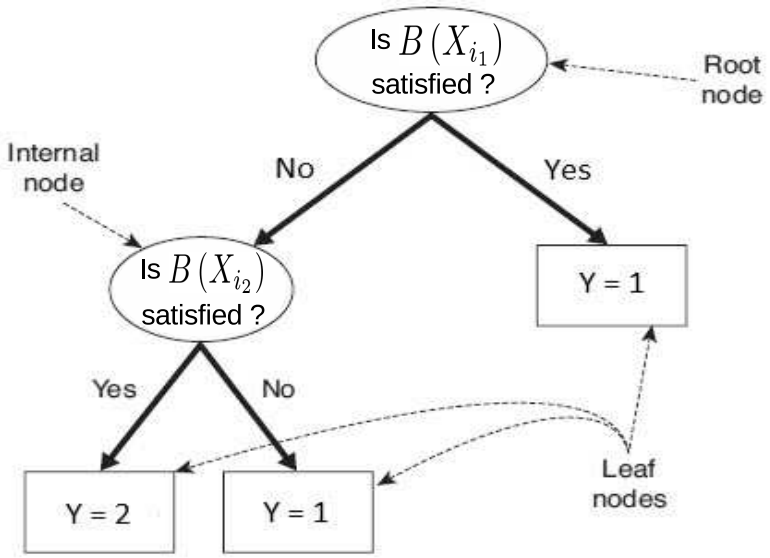

Figure 2.1: *An example for the basic CART structure*

full training data while each of the internal and leaf nodes contains a subset of the data associated with its parent node. The whole structure of a classification tree is accomplished via a recursive binary splitting procedure applied for each node. This procedure aims to separate the training data into reasonably purer subsets in terms of their classes distribution. A stop condition is employed to terminate the splitting process.

Generally, an exponential number of distinct classification trees can be built from a given set of features. While some of them are more accurate than others, finding the optimal tree is computationally infeasible for high dimensional datasets due to the exponential size of the entire search space (Tan et al. 2007). Nevertheless, many algorithms have been developed for building a reasonably accurate CART in a reasonable amount of time. These algorithms usually grow a tree by making a series of locally optimum decisions about which feature to use for partitioning the training data at each node.

2.2.1 Best Split of Nodes

The CART algorithm provides a tool for determining a test condition $B(X_i)$, associated with feature X_i, that should be selected among different feature types in order to get the best split for a given node.

For binary features, the test condition $B(X_i)$ generates two potential outcomes by which a two-way split is formed.

Nominal features having many values produce a test condition which can be expressed either into multi-way split or a two-way split. For the former splitting way, each outcome corresponds to one of the feature distinct values. For the latter way, splitting is accomplished by grouping the feature values into two non-empty disjointed subsets. Some algorithms, such as CART, produce only two-way (binary) splits by considering all the $2^{m-1} - 1$ ways of creating a binary partition of m feature values.

Similarly, ordinal features can produce multi-way or binary split providing that the grouping process, if any, does not violate the order of the feature values.

Finally, for continuous features, the test condition $B(X_i)$ can be expressed as a comparison test with binary outcomes ($X_i \leq \alpha$ Vs. $X_i > \alpha$). Otherwise, it could be presented as a range query with outcomes of the form $\alpha_i < X_i \leq \alpha_{i+1}$, $i = 1, \ldots, m$. One possible approach is discretizing the continuous values into ordinal intervals. Afterwards, each new ordinal value will be assigned to one outcome of a multi-way split. Also, adjacent intervals can be grouped into binary outcomes as long as the order is preserved (Tan et al. 2007).

It is essential to define an objective measure for evaluating the goodness of each test condition, and then identifying the best test condition. This goodness of a test condition is estimated upon the impurity level of nodes, discussed in the following paragraph, before

and after splitting using that condition. On this basis, the best is the one which leads to lowest impurity of observations class distribution before and after splitting.

Measures for detecting the best split

Selection of the best split is based on the degree of impurity. An objective measure can be defined for evaluating the goodness of the split by comparing the degree of impurity of the parent with those of child nodes. The most widely used impurity measures (Friedman et al. 2001), *Gini Index*, *Entropy* and *Classification Error*, are shown respectively in (2.1)-(2.3).

$$Gini\ Index\,(t) = 1 - \sum_{c=1}^{C} \theta_{ct}^2\,, \tag{2.1}$$

$$Entropy\,(t) = -\sum_{c \in S_t} \theta_{ct}\, log_2\, \theta_{ct}\,,\ S_t = \{c \mid \theta_{ct} \neq 0,\ c = 1, \ldots, C\}\,, \tag{2.2}$$

$$Classification\ Error\,(t) = 1 - \max_{c}\, \theta_{ct}, \tag{2.3}$$

where θ_{ct} denotes the proportion of observations belonging to class c among all observations at a given node t, and C is the number of classes. For Entropy measure, the summation is only over the non-empty classes, (i.e., classes for which $\theta_{ct} \neq 0$). Impurity degree for a node t can be computed using these measures. Smaller values represent more skewness for the class distribution of given observations, thus more benefits for splitting purposes.

Figure 2.2 shows the values of these measures for binary classification situations (when $C = 2$). In this case, θ_{ct} represented by the x-axis in Figure 2.2, can refer to any of the two classes since $\theta_{1t} = 1 - \theta_{2t}$.

Now, the goodness of a test condition at a node t can be evaluated by comparing the impurity degree calculated using one of these impurity measures for the node before

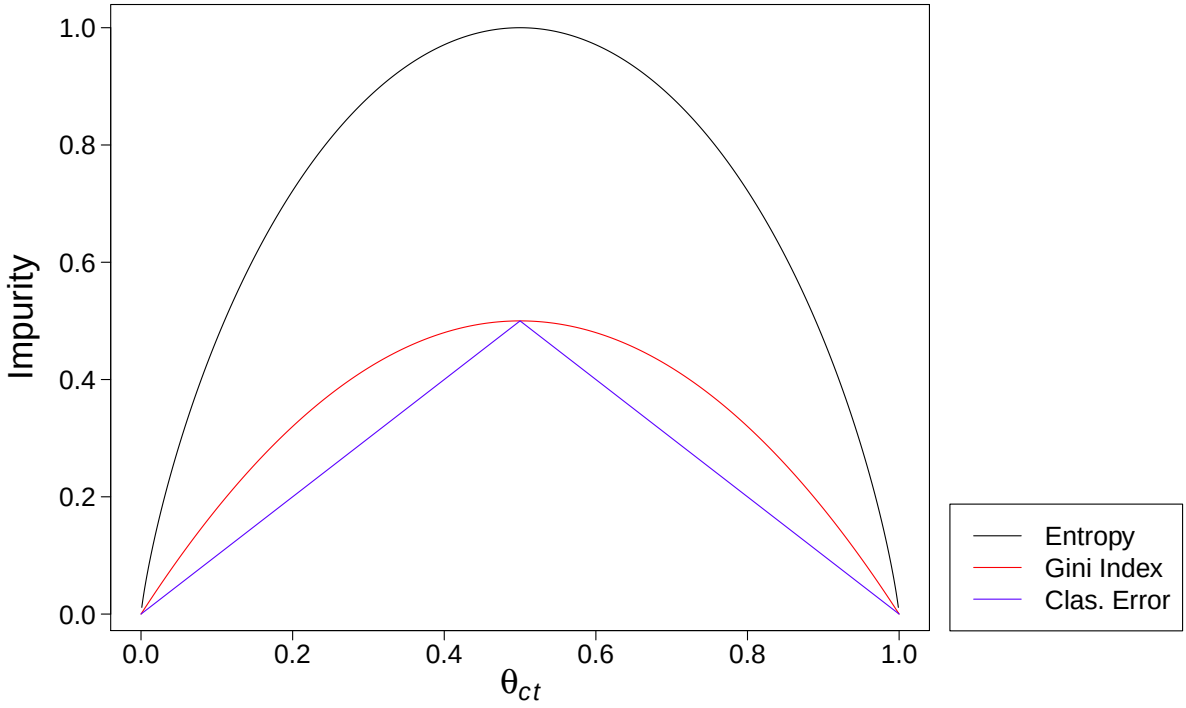

Figure 2.2: *Impurity measures for binary classification problems*

splitting, with impurity degrees of the resulted child nodes, after splitting. The best condition is the one which leads to the largest difference between impurity degrees of before and after splitting. The criterion used for determining goodness of a node split is called "gain" and denoted by $\triangle$ such that (Friedman et al. 2001):

$$\triangle_t = I(t) - \sum_{\beta=1}^{m} \frac{n_{t_\beta}}{n_t} . I\left(t_\beta\right) \tag{2.4}$$

where $I\left(t_\beta\right)$ is the computed impurity of the child node t_β, whereas n_{t_β} refers to the number of observations associated with it, while m represents the number of children nodes (feature outcomes). Algorithms often choose a test condition that maximizes $\triangle$ or equivalently minimizes the weighted average impurity measures of the child nodes as $I(t)$ is the same for all test conditions at a given node t.

2.3 Supervised Ensemble Learning

Ensemble learning was originally presented for supervised learning, specifically classification problems, in 1965 (Nilsson 1965). The basic concept of supervised ensemble learning is to train multiple base classifiers which are basically designed for the same task, then combine their predictions into a single ensemble classifier. This classifier should perform better than any member of base classifiers; otherwise, ensemble process doesn't make sense. In general, diverse weak base classifiers together can produce a strong ensemble classifier if they are given an opportunity to operate within the same procedures. This technique allows better performance, in terms of both generality and accuracy, than could be achieved via any single base classifier. For classification tasks, many researchers have demonstrated the outstanding performance of ensemble learning (Ho et al. 1994, Cho & Kim 1995, Breiman 1996, Oza & Tumer 2001, Dietterichl 2002, Tumer & Oza 2003).

An ensemble procedure should address some fundamental issues such as: how to train each of the single bases (learning algorithm); how to combine their predictions (combining method); how to measure the diversity as a necessary but not sufficient condition of ensemble learning efficiency (diversity). These fundamental arguments are discussed sequentially in Sections 2.3.1 - 2.3.3.

2.3.1 Ensemble Learning Algorithms

Multiple base classifiers can be trained either individually in a parallel system or coordinately in a sequential system. Two of the most common ensemble algorithms, Bagging of classification trees and Boosting, are presented respectively in this section. Those two

powerful algorithms give explicit examples for these different strategies of ensemble methods.

Bagging algorithms

It is one of the simplest and well-known ensemble algorithms. Originally, Bootstrap aggregation (*bagging*) was introduced in Breiman (1996) for supervised ensemble learning. However afterwords, the idea has been extended for unsupervised learning. Fischer & Buhmann (2003) developed uses of the bagging algorithm for some clustering tasks. In addition, Dudoit & Fridlyand (2003) demonstrated the efficiency of using bagging for improving the accuracy of clustering.

The main idea is based on building successive base learners. Each of them is constructed using a bootstrap sample of the considered dataset. Then, a majority vote is taken for prediction. For instance, bagging of the classification trees is conducted by building successive trees using bootstrap samples of the training dataset. Then, the majority vote of predicted classes is taken for the output prediction of bagging. The ensemble learner is then able to reduce the variance of the estimated prediction function (Friedman et al. 2001). A summary of the bagging procedure for classification trees is shown in Algorithm 2.1.

Boosting algorithms

Like bagging, the boosting approach was firstly proposed for a supervised ensemble and was originally designed for classification problems. Freund & Schapire (1997) introduced an algorithm named Adaboost which inspired Boosting algorithms for improving performance of classification (Schapire et al. 1998). Then, some recent studies (e.g. Pavlovic

Algorithm 2.1 *Bootstrap Aggregation (Bagging) for Classification Trees*

Inputs: Set of training data (X, Y).
Output: Ensemble of classification trees $\{T_b\}_1^B$.

1: **for** $b = 1$ *to* B **do**
2: draw a bootstrap sample s_b of size N from the training dataset.
3: grow a classification tree T_b based on the bootstrapped sample s_b.
4: **end for**
5: **return** Ensemble of trees, $\{T_b\}_1^B$.

To make a prediction at a new observation x_{new} ***:***
6: Let $\hat{f}_b(x_{new})$ be the class prediction of the bth classification tree.
7: **return** $\hat{y}_{new} = \hat{f}_{bag}^B(x_{new}) = majority\ vote\ \left\{\hat{f}_b(x_{new})\right\}_1^B$.

2004, Frossyniotis et al. 2004, Saffari & Bischof 2007, Liu et al. 2007) have extended it into unsupervised learning.

The main idea of boosting technique is that a powerful ensemble classifier can be produced by integrating the outputs of various weak classifiers. From this perspective, boosting bears a resemblance to bagging. However, we shall illustrate that the similarity is at best superficial and that boosting is fundamentally different.

The basic idea is that each training individual observation is associated with an adapted weight based on how the observation was classified in the previous iteration, initial weights are usually set in a balanced setting at the first iteration. Observations with higher weight values (more misclassified) are then more likely to be selecting for training data of the next iteration, paying more attention to observations that are difficult to classify. By sequentially constructing a linear combination of base classifiers which are fitted at each iteration, boosting can concentrate more on 'difficult' individual observations and hence provide an effective ensemble classifier for the considered classification problem.

To illustrate boosting, consider a C-classes classification problem, with the response variable Y. Given a vector of predictors X, a classifier $f_b(X)$ predicts the class label, $\hat{y}$, where

$\hat{y} \in \{1, \ldots, C\}$. Then, the misclassification error rate of the classifier $f_b(X)$ can be shown as:

$$err_b = \frac{1}{N}\sum_{j=1}^{N} I\left(y_j \neq \hat{y}_j\right), \tag{2.5}$$

where

$$I\left(y_j \neq \hat{y}_j\right) = \begin{cases} 1 & if \left(y_j \neq \hat{y}_j\right) \\ 0 & Otherwise \end{cases},$$

such that N represents the number of observations in the training dataset, y_j and $\hat{y}_j$ are the observed and the predicted class respectively of the observation j, $i = 1, \ldots, N$. A weak classifier is the one whose error rate is slightly better than the random guessing. The purpose of boosting is to sequentially fit a base classifier to adaptively versions of the training dataset, then producing a sequence of classifiers $f_b(X), \quad b = 1, \ldots, B$ which are used for classification prediction (Hastie et al. 2009). Consequently, ensemble of these base classifiers produces a final classifier $f(X)$ whose prediction is a weighted majority vote of the base classifiers prediction. Thus, prediction of $f(X)$ for the input x_j, the features value of the jth observation, can be expressed as:

$$\hat{f}^{B}_{boost}\left(x_j\right) = \underset{c}{argmax}\left(\sum_{b=1}^{B}\tau_b \cdot I\left(\hat{f_b}\left(x_j\right) = c\right)\right), \qquad c = 1, \ldots, C. \tag{2.6}$$

where,

$$I\left(\hat{f_b}\left(x_j\right) = c\right) = \begin{cases} 1 & if\ jth\ observation\ is\ assigned\ to\ class\ c\ by\ classifier\ f_b(X) \\ 0 & Otherwise \end{cases}. \tag{2.7}$$

Here, $\tau_1, \tau_2, \ldots, \tau_B$ are computed according to the used boosting algorithm. They weight

Algorithm 2.2 *Boosting*

Inputs: Set of training data (X, Y).
Output: Ensemble classifier $f^B_{boost}(X)$.

1: Initialize weights of the observations such that $w_j = \frac{1}{N}, \quad j = 1, \ldots, N.$
2: **for** $b = 1$ *to* B **do**
3: fit a classifier $f_b(X)$ to the training data sampled using the weights w_j.
4: $err_b = \frac{\sum_{j=1}^{N} w_j I(y_j \neq \hat{f}_b(x_j))}{\sum_{j=1}^{N} w_j}$
5: based on err_b, calculate τ_b by which the contribution of the classifier $f_b(X)$ is weighted.
6: Update w_j based on the current status (misclassified or not) of the jth observation, $j = 1 \ldots, N$.
7: **end for**
8: **return** the final classifier, $f^B_{boost}(X)$, by aggregating the base classifiers $f_b(X)$ associated with their weights τ_b, $b = 1, \ldots, B$.

To make a prediction at a new observation x_{new} ***:***

9: **return** $\hat{y}_{new} = \hat{f}^B_{boost}(x_{new}) = \underset{c}{argmax}\left(\sum_{b=1}^{B} \tau_b \cdot I\left(\hat{f}_b(x_{new}) = c\right)\right), \qquad c = 1, \ldots, C.$

the contribution of each base classifier $f_b(X)$ and their effect is to give higher impact to the more accurate classifiers in the sequence. Algorithm 2.2 shows the general procedure of the boosting technique.

Different boosting algorithms modify the general procedure shown in Algorithm 2.2. For instance, AdaBoostM1 algorithm, the most popular boosting algorithm (Freund & Schapire 1997), defines τ_b and w_j (lines 5 and 6 respectively in Algorithm 2.2) as follows:

$$\tau_b = log\left\{\frac{1 - err_b}{err_b}\right\},$$

$$w_j \leftarrow w_j \cdot exp\left[\tau_b \cdot I\left(y_j \neq \hat{f}_b\left(x_j\right)\right)\right].$$

Boosting ensemble algorithm is constructed based on sequential iterations with pertinent feedback from the previous base classifier. This is different from parallel algorithm strategies applied in Bagging and Random Forest (introduced in Sections 2.3.1 and 2.4

respectively).

2.3.2 Ensemble Combining Methods

Whenever multiple base classifiers are constructed, the ensemble learning algorithms should apply a convenient tool to combine their individual outputs into a single form of ensemble classifier. There are a large number of methods for model combination. *Linear combiner*, *the product combiner*, and *majority voting combiner* are the most commonly used in practice and demonstrate good performance for a numerous applications of ensemble learning (Brown 2009).

The **linear combiner** is for models whose response is a real-valued variable. It is used for some supervised learning tasks such as regression and classification which produce estimated class probabilities. For the latter case, the linear combiner can be formulated as an ensemble probability estimate as follows:

$$p\left(\hat{y}|x\right) = \sum_{b=1}^{B} \tau_b \cdot p_b\left(\hat{y}|x\right) \tag{2.8}$$

where $p_b\left(\hat{y}|x\right)$ is the probability estimate of class label y given the input data x using the bth base classifier. While τ_b is the assigned weight of the bth classifier.

The **product combiner** is more suitable than linear under the assumption that the class probability estimates $p_b\left(\hat{y}|x\right),\ b = 1, \ldots, B$ are independent. It is the theoretically optimal combination strategy under that assumption. This combiner can be formulated by multiplying base classifiers' probability estimates as follows:

$$p_b(\hat{y}|x) = \frac{1}{\gamma}\prod_{b=1}^{B} p_b(\hat{y}|x) \tag{2.9}$$

where γ is a constant functioning as a normalization factor to adjust $p(\hat{y}|x)$ into a form of a valid distribution (Brown 2009).

Both linear and product combiners are employed if and only if the base classifiers produce real-valued outputs. When the base classifier instead estimates the class labels, the **majority voting combiner** can be used. Using this method, the class label with the most votes among all trained base classifiers is assigned as the ensemble prediction. Therefore, the ensemble prediction output using majority vote combiner can be formulated as shown in Boosting, (cf., (2.6)). When the weights of base classifiers are uniformly distributed (*i.e.* $\tau_b = 1/B$, $\forall b$), a simple majority voting combiner is employed as in Bagging, (cf., line 7 in Algorithm 2.1).

2.3.3 Ensemble Diversity

An ensemble is performed by complementing a weak single classifier with other base classifiers, which make errors on different observations, to enhance the diversity among the combined classifiers. Diversity of the base classifiers is considered a necessary but not sufficient condition for the success of the ensemble learning. The bases have to be diverse and accurate in order to produce an optimal ensemble learning output.

Measurements of ensemble diversity could be divided into two distinct groups, pairwise measures and non-pairwise measures. In the former group, the difference between a pair of base classifiers is considered one at a time, the ensemble diversity measure is then obtained

by averaging overall differences across all pairs (e.g., *Double-fault measure* (Giacinto & Roli 2001) and *Disagreement measure* (Skalak et al. 1996)). On the other hand, non-pairwise measures consider all the base classifiers together (e.g., *Entropy measure* (Cunningham & Carney 2000), *Generalized Diversity* (Partridge & Krzanowski 1997), *Kohavi-Wolpert Variance* (Kohavi et al. 1996) and *Measure of Difficulty* (Hansen & Salamon 1990)).

2.4 Random Forest

The Random Forest (RF) approach was developed by Breiman (2001) as an extension of the Classification and Regression Trees (CART) technique presented in Section 2.2. The Bagging algorithm, described in Section 2.3.1, is considered the basis of the Random Forest. Since Bagging constructs each tree using a different bootstrap sample of the dataset, RF has a similar procedure to Bagging with an additional layer of randomness. RF consists of bagging of decision tree learners with a randomized selection of predictors at each split. Unlike CART, each node is split using the best among a randomly chosen subset of predictors. RF achieves a powerful performance compared to many other classifiers including discriminant analysis, neural networks and support vector machines, and is robust against over-fitting (Breiman 2001). The algorithm of RF modified from Hastie et al. (2009) for classification problems is introduced in Algorithm 2.3.

The main idea of Bagging, shown in Section 2.3.1, is to average many noisy models in order to reduce the variance of the final ensemble model. Trees are ideal candidates for applying Bagging since they are famed as noisy models, thus they can benefit greatly from the averaging process. Since each tree constructed using Bagging procedure is identically

Algorithm 2.3 *Random Forest for Classification*

Inputs: Set of training data (X, Y).
Output: Random Forest classifier $f_{RF}(X)$.

1: **for** $b = 1$ *to* n_{tree} **do**
2: draw a bootstrap sample, s_b, of size N form the original dataset.
3: construct a random-forest tree T_b to the bootstrapped sample, s_b, by recursively repeating the following steps for each terminal node, until reaching the stop condition, which might be minimum node size n_{min} or the terminal node contains members of only one class:
3a: select m_{try} predictors at random from the P predictors.
3b: choose the best split among those m_{try} predictors.
3c: split the node into two child nodes.
4: **end for**
5: **return** the ensemble of trees, $\{T_b\}_1^{n_{tree}}$.

To make a prediction at a new observation x_{new} ***:***

6: let $\hat{f}_b(x_{new})$ be the class prediction of the bth random-forest tree T_b.
7: **return** $\hat{f}_{RF}(x_{new}) = majority\ vote\ \left\{\hat{f}_b(x_{new})\right\}_1^{n_{tree}}$.

distributed (i.d.), the expectation of their average is the same as the expectation of any single tree of them. In other words, the bias of bagged trees is equivalent to the bias of the individual trees. Hence, the only hope of improvement is via variance reduction. This idea is in contrast to boosting, shown in Section 2.3.1, as the trees are sequentially grown to repeatedly reduce the bias, and hence they are not i.d. trees (Hastie et al. 2009).

The variance of the average of n_{tree} i.d. variables $T_1, \ldots, T_{n_{tree}}$ with variance σ^2 and positive pairwise correlation ρ can be expressed as:

$$\begin{aligned} V\left[\frac{1}{n_{tree}}\left(T_1 + \ldots + T_{n_{tree}}\right)\right] &= \frac{1}{{n_{tree}}^2}\left[n_{tree} \cdot \sigma^2 + n_{tree}\left(n_{tree} - 1\right)\rho\sigma^2\right] \\ &= \frac{1-\rho}{n_{tree}} \cdot \sigma^2 + \rho\sigma^2 \end{aligned} \tag{2.10}$$

Hence, as n_{tree} increases, the first term of (2.10) tends to disappear, but the second remains. Therefore, the magnitude of the correlation between pairs of bagged trees can

affect the benefits of averaging. A higher correlation between results in a higher variance of the ensemble. The idea of RF (Algorithm 2.3) is to improve the performance by reducing the variance of bagging through decreasing the correlation between the trees, without increasing the variance of them too much. This idea can be achieved by selecting m_{try} predictors randomly among all the P predictors at each split through tree-growing process. This leads to the production of more diverse trees (see ensemble diversity in Section 2.3.3). Therefore, Bagging can be thought of as the special case of RF obtained when $m_{try} = P$. Usually, m_{try} values are chosen as $\sqrt{P}$, which is the default setting in the R package '*randomForest*' (Liaw & Wiener 2002), but sometimes they are as low as 1.

When a bootstrap sample is drawn with replacement from the data, some observations are not involved in this bootstrap sample. These are called 'out-of-bag' (OOB) observations and can be used to give an internal estimate of the misclassification error rate. On average, each observation would be OOB 36.8% of times, since each observation has the probability $\left(1 - \frac{1}{N}\right)^N$ for being OOB observation of a particular bootstrap sample. As N tends to be large, this probability tends to $e^{-1} \approx 0.368$.

For computing this OOB error rate, each tree is used to predict the class for its OOB observations. Therefore, for each observation, the error rate is estimated by averaging the misclassification predictions produced by the trees for which this observation was out-of-bag. An overall error rate (OOB error rate) can be estimated by averaging over all the observations.

RF is not sensitive to the choice of any of its parameters. Therefore, the default choices of $n_{tree} = 500$, $m_{try} = \sqrt{P}$ and $n_{min} = 1$ work well for most classification problems (Cutler & Stevens 2006). Consequently, fine-tuning is not essentially required and its effect should

be relatively small (Díaz-Uriarte & De Andres 2006). Moreover, Breiman (2001) shows that adding more trees to an ensemble of the random forest does not lead to an over-fitting problem.

In regression, the depth of the trees should be controlled by determining the minimum number of observations in the leaf nodes. Hence, the parameter of minimum node size, n_{min}, needs to be tuned. The default setting for regression problems is set to be 5 in the '*randomForests*' R package (Liaw & Wiener 2002).

Merits of Random Forest

Many positive properties make RF an effective approach for classification tasks within high-dimensional datasets in terms of the prediction accuracy. Some of these properties are (Breiman 2001):

1. RF is considered one of the most accurate learning algorithms available for classification problems throughout high-dimensional settings.

2. It can present the same level of highly accurate performance on large databases.

3. It is usually not very sensitive to training data outliers.

4. It provides estimates of feature importance in classification problems. This merit has special influence when applying RF for datasets which contains large number of features, such as microarray gene expression or proteomics data sets in which genes or proteins are carrying various biological characteristics with different impact on the predicted classes.

5. High effective performance could be held even when dealing with thousands of features, as the situation of gene expression microarray datasets.

6. If a large proportion of the data are missing, RF involve an effective method for estimating these missing data and maintain the same level of accuracy.

7. RF provides proximities that can be used for clustering purposes.

8. It is very user-friendly in the sense that it has only three tuning parameters: the total number of trees in the forest, the number of predictors within the random subset at each node and the minimum node size which are represented by n_{tree}, m_{try} and n_{min} respectively.

2.5 k Nearest Neighbour

Another simple approach for classification problems is the k nearest neighbour (kNN) classifier. It is a non-parametric supervised learning algorithm which performs a lazy learning strategy, where generalization beyond training data is deferred until a test observation is required to be classified. It uses the training dataset with a nearest neighbour rule to classify an observation to a target class c. In kNN, a set of k training observations that are closest to the test observation in the feature space are identified and then the test observation is classified to the class of majority in these k nearest observations. If $k = 1$, then the test observation is simply assigned to the class of its nearest neighbour. For finding nearest neighbours of a test observation, a distance (similarity) metric is used (e.g., Euclidian distance). The key elements of kNN approach, upon which its performance mainly depends,

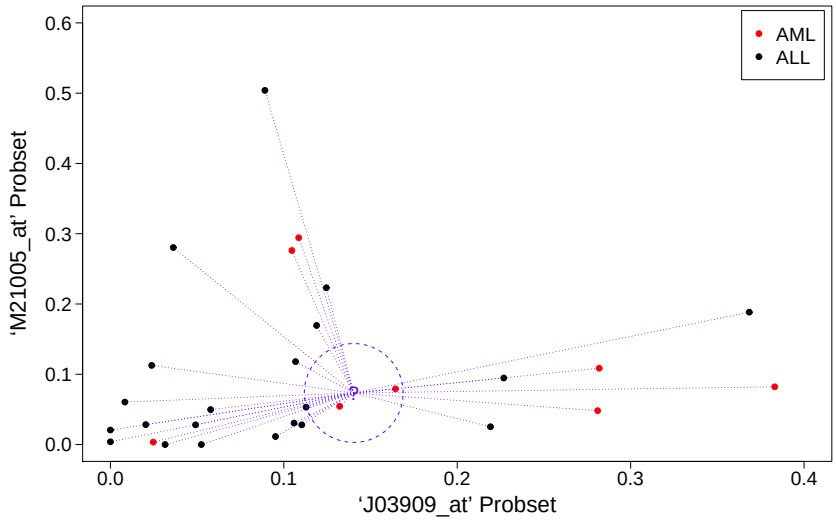

Figure 2.3: *k-nearest neighbour framework in a 2-dimensional feature space for 'Leukaemia' dataset. Blue lines represent the Euclidian distances from the test observation, depicted by '?' symbol, to all the training observations belonging to two classes: Acute Myeloid Leukemia (AML) shown in red dots; Acute Lymphoblast Leukemia (ALL) shown in black dots. For k = 3, the nearest neighbours are identified within the dashed circle and the test observation is assigned to 'AML' class as the majority of the nearest observations*

are the distance metric and the chosen value of k (Ghosh et al. 2005).

The general framework of a kNN classifier in 2-dimensional setting is shown in Figure 2.3. From the 'Leukaemia' dataset (described in Appendix A.2), a subset of observations whose gene expressions fall within a particular domain in respect with the considered feature space is shown in Figure 2.3. Two features (genes), 'J03909_at' and 'M21005_at', are represented on the horizontal and vertical axes respectively. The patients that represent the training observations belong to one of two types of Leukaemia, either Acute Myeloid Leukemia (AML) or Lymphoblast Leukemia (ALL). The test observation, denoted by '?' symbol, is classified to the class of the majority in their neighbourhood. Euclidian distances for the test observation (point) are measured from all given training observations. Then its k nearest neighbours are identified based on the lowest distance ($k = 3$ in Figure 2.3). The

observation is assigned to the popular class in its neighbourhood which is 'AML'.

General Rule of k Nearest Neighbour Classifier

According to the kNN rule, an unclassified (test) observation, x_{new}, is assigned to the class label, $\hat{y}_{new}$, of majority in its k nearest neighbours among the training dataset, where $\hat{y}_{new} = c$ and $c \in \{1, \ldots, C\}$. Although classification is the primary application of kNN, it can be also used for density estimation.

The k data points in the feature space lying within the neighbourhood of an observation x_{new} are used to estimate the density function at x_{new}. The neighbourhood is identified using a form of distance measure. A sphere (circle in two dimensional settings) centered at x_{new} capturing the k training points of this neighbourhood, irrespective of their classes, is drawn. The estimated density at x_{new} can be defined as:

$$\hat{p}(x_{new}) = \frac{k}{vN}, \tag{2.11}$$

where v denotes the volume (area) of the sphere (circle). When the density at x_{new} is high, then k points can be quickly found as they are intuitively close to x_{new}. Hence, the volume of the required sphere is small and then the obtained density, according to (2.11), is high. On the other hand, when the density is low then the volume of the sphere required to encompass k nearest neighbours is large which leads to obtain a low density from (2.11). Therefore, the density is mainly influenced by v which performs a similar role to the bandwidth parameter in kernel density estimation.

The estimated conditional density of x_{new} given a class c can be similarly defined as:

$$\hat{p}\left(x_{new} \,\middle|\, y_{new} = c\right) = \frac{k_c}{vN_c}\,, \tag{2.12}$$

where k_c and N_c denote number of observations from the cth class that are involved within the sphere and the entire training data respectively, such that $k = \sum_{c=1}^{C} k_c$ and $N = \sum_{c=1}^{C} N_c$. The estimator of class prior probability denoted by $\hat{\pi}$ is given by:

$$\hat{\pi} = \hat{p}\left(y_{new} = c\right) = \frac{N_c}{N}. \tag{2.13}$$

Using Bayes rule, the posterior probability for class membership of the test observation x_{new} can be expressed by combining (2.11)-(2.13) as follows:

$$\hat{p}\left(y_{new} = c \,|x_{new}\right) = \frac{\hat{p}\left(x_{new} \,\middle|\, y_{new} = c\right) \cdot \hat{p}\left(y_{new} = c\right)}{\hat{p}\left(x_{new}\right)} = \frac{\frac{k_c}{vN_c} \cdot \frac{N_c}{N}}{\frac{k}{vN}} = \frac{k_c}{k}. \tag{2.14}$$

The test observation is assigned to the class label c that has the largest fraction of the observations belonging to c among the k nearest neighbours of the test observation (Bishop et al. 2006).

2.6 Support Vector Machine

One of the most common classifiers is the Support Vector Machine (SVM). It is a well-known supervised learning model in which training observations are used to recognize a pattern that can predict the classes of new observations. An SVM model is a representation of a

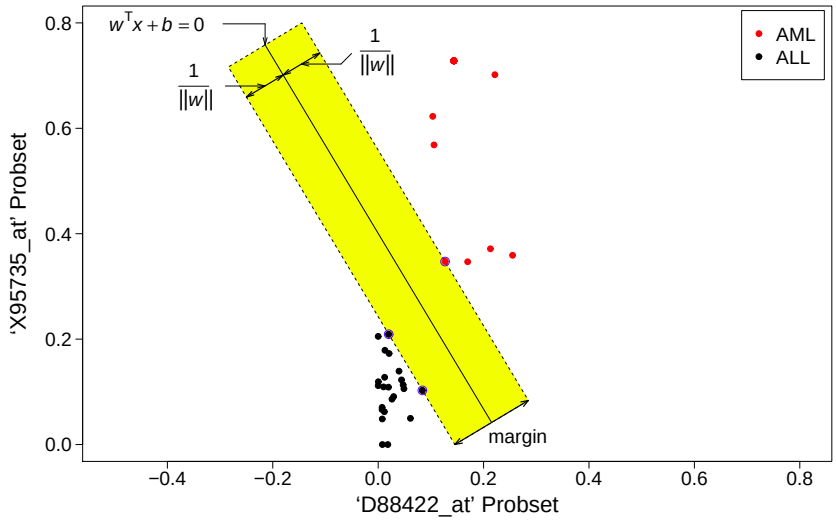

Figure 2.4: *Support vector classifier in a* 2*-dimensional feature space for the 'Leukaemia' dataset with a two linearly separable classes: Acute Myeloid Leukemia (AML) shown in red dots; Acute Lymphoblast Leukemia (ALL) shown in black dots. The hyperplane (line, in* 2*-dimensional setting) of the decision boundary is the solid line, while dashed lines bound the shaded maximal margin of width* $2/\|w\|$*. The points highlighted by blue circles that lie on the margin boundaries are called 'support vectors'.*

hyperplane that separates 'optimally' the feature space into two disjoint regions such that training observations of separate classes are divided by this hyperplane into two groups with a 'margin' that is as maximum as possible.

For situations of linearly separable classes as illustrated by Figure 2.4, the main goal is to design a hyperplane

$$f(x) = w^T x + b = 0 \tag{2.15}$$

that classifies correctly all the training observations. A classification rule that associated with this hyperplane can then be given by

$$f_{SVM}(X) = sign\left[w^T x + b\right]. \tag{2.16}$$

It can be shown that $w^T x_j + b$ gives the signed distance from a point x_j to the hyperplane defined in (2.15). Since the classes are linearly separable, one can find a hyperplane $f(x)$, as shown in (2.15), such that $y_j . f\left(x_j\right) > 0 \ \forall j$ where $y_j \in \{-1, 1\}$. Such a hyperplane is not unique (Vapnik & Vapnik 1998). The best solution is the one that has the maximum 'margin' between the training observations from different classes, see Figure 2.4.

The Optimization Problem

For simplicity, the vector w is normalized so that

$$\left|w^T x_{sv} + b\right| = 1 \tag{2.17}$$

where x_{sv} is a support vector for the assigned hyperplane. Since, w is a perpendicular vector on the hyperplane in (2.15), the Euclidean distance from the hyperplane to its support vector is the projection of the vector $x_{sv} - x$ on w, where x can be any point on the hyperplane $w^T x + b = 0$. The margin is defined as the double of this distance. Therefore, the assigned margin can be defined as

$$\begin{aligned} margin &= 2 . \left|\frac{w}{||w||} . (x_{sv} - x)\right|, \\ &= \frac{2}{||w||} \left|w^T x_{sv} + b - \left(w^T x + b\right)\right| = \frac{2}{||w||}. \end{aligned} \tag{2.18}$$

The margin in (2.18) is obtained by applying the expressions shown in (2.15) and (2.17). Now, the following optimization problem should be considered in order to assign a hyperplane with maximum margin for given training observations.

$$\begin{aligned} &\textit{minimize} \quad \tfrac{1}{2}\, w^T w \\ &\textit{subject to} \quad y_j\left(w^T x_j + b\right) \geq 1, \qquad j = 1, \ldots, N\,, \quad w \in \mathbb{R}^P,\ b \in \mathbb{R}. \end{aligned} \tag{2.19}$$

This problem is quadratic with linear inequality constraints. Therefore, it is a convex optimization problem that can be solved by quadratic programming by means of Lagrange multipliers (Vapnik & Vapnik 1998). The corresponding Lagrange (primal) function using Karush-Kuhn-Tucker (KKT) approach (Boyd & Vandenberghe 2009) can be expressed as

$$L\left(w, b, \alpha\right) = \frac{1}{2} w^T w - \sum_{j=1}^{N} \alpha_j \left(y_j \left(w^T x_j + b\right) - 1\right) \tag{2.20}$$

which is minimized with respect to w and b such that $\alpha_j \geq 0$, where α represents the vector of Lagrange multipliers, $\alpha \in \mathbb{R}^N$. Setting the respective derivatives to zero results in

$$w = \sum_{j=1}^{N} \alpha_j y_j x_j, \tag{2.21}$$

$$\sum_{j=1}^{N} \alpha_j y_j = 0. \tag{2.22}$$

By substituting (2.21) and (2.22) into (2.20), the Lagrange (dual) objective function can be given by

$$L\left(\alpha\right) = \sum_{j=1}^{N} \alpha_j - \frac{1}{2} \sum_{j=1}^{N} \sum_{l=1}^{N} \alpha_j \alpha_l \; y_j y_l \; x_j^T x_l. \tag{2.23}$$

The $L\left(\alpha\right)$ in (2.23) is maximized with respect to α subject to $\alpha_j \geq 0$ and (2.22), $j = 1, \ldots, N$. In addition to (2.21) and (2.22), the KKT conditions include the constraint

$$\alpha_j \left(y_j \left(w^T x_j + b\right) - 1\right) = 0. \tag{2.24}$$

These constraints uniquely characterize the solution to the primal and dual problem. In view of (2.17) and (2.24), it can be shown that $\alpha_j > 0$ for support vectors (i.e., for each x_{sv}), whereas $\alpha_j = 0$ for the other training observations.

SVM provides a procedure that can control its sensitivity to potential outliers when the considered datasets are noisy. When the feature space has no linear separation between observations from different classes, SVM introduces slack variables that allow the margin to be violated. Such a margin is called "soft margin" (Vapnik & Vapnik 1998).

For non-linearly separable situations, SVM can perform classification efficiently by transforming the original feature space, X, into another space Z, usually with higher dimensions, using a function called the 'kernel'. The optimization problem can be expressed in a way that only involves the input features via inner products. Therefore, transformed feature vectors z_j for the input feature vectors x_j are considered and the corresponding Lagrange dual function in (2.23) is expressed in the form

$$L(\alpha) = \sum_{j=1}^{N} \alpha_j - \frac{1}{2} \sum_{j=1}^{N} \sum_{l=1}^{N} \alpha_j \alpha_l \, y_j y_l \, z_j^T z_l. \tag{2.25}$$

where $z_j^T z_l$ is the transformed inner product using the kernel function $K\left(x_j, x_l\right)$. Hence, the transformed space, Z, is not required at all, but we require only the kernel function which produces the inner products in the transformed space, $z_j^T z_l$. A valid kernel should be symmetric positive semi-definite function (Friedman et al. 2001). Two common choices

for kernel function in the SVM literature are

$$\begin{aligned} Qth - Degree\, polynomial: \quad & K\left(x_j,\, x_l\right) = \left(1 + x_j^T x_l\right)^Q, \\ Radial\, basis: \quad & K\left(x_j,\, x_l\right) = exp\left(-\gamma \left\|x_j - x_l\right\|^2\right). \end{aligned} \tag{2.26}$$

2.7 Classifier Performance Evaluation via Cross Validation

A main task in a pattern recognition problem is the assessment of the model performance and its generalization for new data. Ideally the accuracy of a classifier should be assessed on an independent data, called the test dataset, while the classification rule is built on other data, called the training dataset. However, in many real world problems (e.g., experiments of microarray gene expressions), limited observations are available and both modelling and assessment of the model are performed on these limited data. A classifier accuracy calculated from the same training dataset leads to underestimate the true or generalized error rate as the classifier is assessed on the same data that is used to fit it, and thereby giving an optimistic measure for the error rate. Various approaches have been proposed to deal with the problem of classification error estimation. The error of a classifier can be expressed in terms of two factors, i.e. bias and variance (Kim 2009). One of the most commonly and effectively used approaches is the cross validation technique.

Cross-Validation Method

The cross validation (C.V.) technique is the simplest method used for error estimation. It copes with the problem of limiting availability of observations by using a portion of the given dataset to fit the classification model, while the remaining part is used for testing the

model. For instance in F-fold cross validation setting, the data is divided into F folds of approximately the same size. Afterwards, $F - 1$ folds are used for fitting the model and one fold is used for testing the model performance. The process is performed F times with different fold at each time. The F estimates of misclassification error rates are then averaged to obtain a single estimate for the classifier error. Small values of F result in highly biased estimators. On the other hand, large values of F lead to more computationally expensive estimators with a high variance. The special case of $F = N$, also known as leave one out cross validation, implies one observation is used for testing and $N - 1$ observations are used for building the classifier. This setting of C.V. gives an unbiased estimator with high variance (Friedman et al. 2001).

2.8 Summary

A statistical learning approach can be used to model and understand complex datasets. By mapping the relationship between a set of features and a considered response, it can build a predictive model based on a given training data. Based on how the training data is presented, the learning process is described as either *supervised*, when a set of features along-with supervised output labels (response) are trained, or *unsupervised*, when the training data contains only the feature matrix.

Supervised ensemble learning trains multiple base models (classifiers) designed for the same task by combining their predictions into a single ensemble classifier. CART are considered base classifiers for most of the ensemble learning methods (e.g., Random Forest). The CART approach constructs a binary decision tree by recursively splitting the

feature space into two disjoint regions.

Three different classifiers are described in this chapter: Random Forest (RF); *k* Nearest Neighbour (*k*NN); Support Vector Machine (SVM).

RF classifier is an ensemble of trees that are constructed using different bootstrap samples of the dataset. Unlike CART, each node is split using the best among a randomly chosen subset of features.

*k*NN uses the training dataset with a nearest neighbour rule to classify an observation to a target class *c*. A set of *k* training observations that are closest to the test observation in the feature space are sorted out and then the test observation is classified to the class of majority in these *k* nearest observations.

SVM uses the training observations to recognize a pattern that can predicts the classes of new (unseen) data. An SVM model is a representation of a hyperplane that separates 'optimally' the feature space into two disjoint regions such that the training observations of separate classes are divided by this hyperplane into two groups with a 'margin' that is as maximum as possible.

Evaluation of a model's performance can be accomplished by estimating its misclassification error rate on a test dataset. One of the most common technique for error estimation is the cross validation method. It uses a portion of the given dataset to fit the classification model, whilst the remaining part is used for testing the model. It copes with the problem of limiting availability of observations in most microarray gene expressions datasets.

The next chapter will describe various techniques for feature selection. Identification of the relevant and informative features required for classification within functional genomic experiments is also discussed.

Chapter 3

Feature Selection

3.1 Introduction

In statistical learning applications (e.g., classification), one might think that abundance of features bring more discriminating power, thus facilitating the learning process. However, it may cause problems in practice as irrelevant and redundant features result in increased complexity of the model and then degrade its predictive power.

Hence, uninformative, irrelevant and redundant features should be removed from the original feature set prior to utilizing a classifier in order to mitigate these problems. This task is termed feature selection. It is a dimensionality reduction process in which the original set of features, involving P features, is reduced to another set with r features where $r < P$.

One of the most important applications is to identify the relevant and informative features required for classification within functional genomic experiments. The next section presents a detailed discussion for this application.

3.2 Gene Selection

Microarray technology, as well as other high-throughput functional genomics experiments, have become a fundamental tool for gene expression analysis in recent years. For a particular classification task, microarray data are inherently noisy since most genes are irrelevant and uninformative to the given classes (phenotypes, e.g. different stages of a cancer disease) (Apiletti et al. 2007*a*). In addition, most supervised learning algorithms, discussed in Chapters 2, are faced with the problem of selecting a relevant subset of genes upon which to focus in order to achieve an effective performance. Consequently, dimensionality reduction as a preprocessing technique is a fundamental task in microarray data mining. It is the process of reducing the number of genes by identifying and removing as much of redundant and irrelevant information yielded by gene expression profiles as possible.

Dimensionality reduction can be performed with regard to either feature selection or feature extraction. The former yields a subset of the original features (in our context, genes), whereas the later applies a transformation of the given feature space into a lower dimensional space. From the biological point of view, it is more efficient to select real genes than to create artificial features with uncertain biological meaning. Therefore, an effective feature selection approach allows better interpreting for the biological relationship between genes and the considered clinical outcome and then gaining more scientific understanding of the given problem.

The aim of feature selection is to identify the most informative genes for a considered model. The identification of discriminative genes for use in classification has been investigated in many studies (e.g. Chen et al. 2014, Dramiński et al. 2008, Marczyk et al. 2013,

Tusher et al. 2001, Zou et al. 2013, Apiletti et al. 2007*b*, 2012, Peng et al. 2005, Su et al. 2003). An assessment of maximally selected genes or prognostic factors, equivalently selected by the minimum p-values approach, has been discussed in Lausen et al. (2004) and Altman et al. (1994) using gene expression data from clinical cancer research. Their solution was to use an appropriate multiple testing framework, but obtaining study or experiment optimised cut-points for selected genes make comparison with other studies and results difficult.

A major challenge is the problem of dimensionality; tens of thousands of genes' expressions are observed in a small number, tens to few hundreds, of observations. The problem of gene selection is to find a subspace of genes that best characterizes the response target variable (class labels in our context). Various alternative search schemes have been proposed to handle the problem of feature selection, e.g. best individual genes (Su et al. 2003), Max-Relevance and Min-Redundancy based approaches (Peng et al. 2005), Iteratively Sure Independent Screening (Fan et al. 2009) and MaskedPainter approach (Apiletti et al. 2012). Identification of discriminative genes can be based on different criteria including: p-values of statistical tests e.g. t-test or Wilcoxon rank sum test (Lausen et al. 2004, Altman et al. 1994); ranking genes using statistical impurity measures e.g. information gain, gini index and max minority (Su et al. 2003); analysis of overlapping expressions across different classes (Apiletti et al. 2007*b*, 2012).

For high dimensional data, overfitting is a common statistical problem that occurs when a model is excessively complex (i.e., having too many parameters relative to the given number of observations). A model that has been overfit usually provides perfect classification performance, with approximately zero error rate, on the training data, but this

seemingly wonderful performance does not apply to new data. Thus, an overfitted model has generally poor predictive power for new data, as it describes the noise in the given data rather than the underlying relationship. A way to improve prediction accuracy, as well as interpretation of the biological relationship between genes and the considered clinical outcomes, is to use a supervised classification based on expressions of discriminative genes identified by an effective gene selection technique. This procedure of pre-selection of informative genes helps in avoiding overfitting and building a faster model by providing only the features that contribute most to the considered classification task. However, a search for the subset of informative genes presents an additional layer of complexity in the learning process (Saeys et al. 2007).

3.3 Methods of Gene Selection

One of the differences among various feature selection procedures is the way they perform the search in the feature space. Three categories of feature selection methods can be distinguished: wrapper, embedded and filter methods.

3.3.1 Wrapper Methods

Wrapper methods evaluate gene subsets using a predictive model which is run on the dataset partitioned into training and testing sets. Each gene subset is used with a training dataset to train the model, which is then tested on the test set. Calculating a model prediction error from the test set gives a score for that gene subset. The gene subset with the highest evaluation is selected as the final set on which to run a particular model. The wrapper

methods are computationally expensive since they need a new model to be fitted for each gene subset. Genetic algorithm based feature selection techniques are representative examples for wrapper methods (Saeys et al. 2007).

3.3.2 Embedded Methods

Embedded methods perform feature selection search as part of the model construction process. They are less computationally expensive than the wrapper methods. An example of this category is a classification tree based classifier (Olshen & Stone 1984). Another common representative example of this approach is the 'Least Absolute Shrinkage and Selection Operator' (LASSO) method (Tibshirani 1996). It constructs a linear model which penalises the regression coefficients, shrinking many of them to zero. Features with non-zero regression coefficients are selected by the LASSO algorithm.

3.3.3 Filter Methods

Filter methods assess genes by calculating a relevant score for each gene. The low-relevant genes are then removed. The selected genes may then be used to serve classification using many types of classifiers. Gene selection filter-based methods can scale easily to high-dimensional datasets since they are computationally simple and fast compared with the other approaches.

There are two sub-categories of filter methods, univariate and multivariate. In classification problems, univariate approaches are based on statistical measures which separately consider each gene to detect differences between two classes (e.g., Wilcoxon rank-sum method) or among three or more classes (e.g., Kruskal-Wallis test method). Ignoring de-

pendencies among genes is a common disadvantage of these methods, which may lead to worse performance when compared with other approaches. Multivariate approaches, instead, incorporate gene dependencies within the feature selection process. The Minimum Redundancy Maximum Relevance (Ding & Peng 2005) method and Fast Correlation Based Feature Selection (Yu & Liu 2004) are two examples for multivariate procedures.

Various examples for filter-based approaches have been proposed for identification of discriminative genes (e.g., Ding & Peng 2005, Talloen et al. 2007, Dramiński et al. 2008, Ultsch et al. 2010, Lu et al. 2011, Marczyk et al. 2013). Filtering methods can introduce a measure for assessing importance of genes (Ding & Peng 2005, Dramiński et al. 2008, Ultsch et al. 2010, De Jay et al. 2013), present thresholds by which informative genes are selected (Marczyk et al. 2013), or fit a statistical model to expression data in order to identify the discriminative genes (Lu et al. 2011, Talloen et al. 2007).

A measure named 'relative importance', proposed by Dramiński et al. (2008), is used to assess genes and to identify informative ones based on their contribution in classification when large number of decision trees have been constructed. The contribution of a particular gene to the relative importance measure is defined by a weighted scale of the overall number of splits made on that gene in all constructed trees. Dramiński et al. (2008) use decision tree classifiers for measuring the genes' relative importance, not for the aim of fitting classification rules.

Ultsch et al. (2010) propose an algorithm, called 'PUL', in which the differentially expressed genes are identified based on a measure for retrieval information named PUL-score.

Ding & Peng (2005) propose a framework, named 'minimum redundancy maximum relevance (mRMR)' based on a series of intuitive measures of relevance, to the response

target, and redundancy, between genes being selected.

De Jay et al. (2013) developed an R package, named 'mRMRe', by which an ensemble version of mRMR has been implemented. It uses two different strategies to select multiple gene sets, rather than a single set, in order to mitigate the potential effect of the low sample-to-dimensionality ratio on the stability of the results.

Marczyk et al. (2013) propose an adaptive filter method based on the decomposition of the probability density function of gene expression means or variances into a mixture of Gaussian components. They determine thresholds to filter genes via tuning the proportion between the pools sizes of removed and retained genes.

Lu et al. (2011) propose another criterion to identify the informative genes in which principle component analysis has been used to explore the sources of variation in the expression data and to filter out genes corresponding to components with less variation.

Talloen et al. (2007) use factor analysis models rather than principle component analysis to identify informative genes. Several comparisons between algorithms of identifying informative genes in microarray data are presented in Liu et al. (2013), Ultsch et al. (2010).

3.4 Gene Expressions Overlap

Analyzing the overlap between gene expression measures for different classes can be another important criterion for identifying discriminative genes. This strategy utilities the information given by observation classes as well as expression data for detection of the differentially expressed genes between target classes. A classifier can then use these selected genes to enhance its classification performance and prediction accuracy. A

procedure specifically designed to select genes based on their overlapping degree across different classes was recently proposed by Apiletti et al. (2007*b*). This procedure, named Painter's feature selection method, proposes a measure calculating an overlapping score for each gene. For binary class situations, this score estimates the overlapping degree between both classes taking into account only one factor i.e., length of the interval of overlapping expressions. It has been defined to provide higher scores for longer overlapping intervals. Genes are then ranked in ascending order according to their scores. This measure has been extended by Apiletti et al. (2012) using another factor, i.e. the number of overlapped observations, in the analysis. Apiletti et al. (2012) characterize each gene by means of a *gene mask* that represents the capability of a gene to unambiguously assign training observations to their correct classes. Characterization of genes using training observation masks with their overlapping scores allow the detection of the minimum set of genes that provides the best classification coverage on a training dataset. A final gene set is then provided by combining the minimum gene subset with the top ranked genes according to the overlapping score. Since gene masks, proposed by Apiletti et al. (2012), are defined based on the range of the training expression intervals, a caveat of this technique is that the construction of gene masks could be affected by outliers.

Since only a small number of the genes, among tens of thousands of potential candidates, show relevance with the targeted disease, many studies address the problem of defining which is the appropriate number of genes to select (Peng et al. 2005). While an excessively conservative selected number of genes may cause an information loss, an excessively liberal number may increase the noise in the resulting dataset.

Chapter 4 proposes a statistical method that selects the minimum number of genes,

based on expressions overlap, that provide the best classification accuracy for observations in a given training dataset, avoiding the effects of expression outliers. The genes belonging to this minimum subset can be used as genetic markers for further biological investigations.

3.5 Summary

Microarray data as well as other functional genomic experiments produce measurements of tens of thousands of genes (features) that are observed in a smaller number of observations, tens to few hundreds. This characteristic of high dimensionality has a great impact on the learning process since most of genes are noisy, redundant or irrelevant to the considered classification problem. These features may result in an overfitting problem by increasing complexity of the model and then degrading its predictive power.

A statistical learning process could be improved by removing the uninformative, irrelevant and redundant features from the original feature space prior to utilizing a classifier. This task is termed feature selection.

Feature selection algorithms differ from each other in the way they perform the search for the subset of informative genes in the feature space. Mainly, there are three categories of feature selection algorithms which are wrapper, embedded, and filter methods.

The wrapper methods evaluate a subset of genes using a predictive model whose prediction error gives a score for that gene subset. The embedded methods perform feature selection search as part of the model construction process. Classification tree based classifier is an example of the embedded feature selection approach where the feature providing the best split for training observations is selected at each node. Whilst the filter

methods assess genes by calculating a relevant score for each gene. The low-relevant genes are then removed.

The process of gene selection can enhance classifier performance, avoid overfitting, provide faster models and gain a deeper understanding and interpretation of the underlying learning process.

The next chapter explains the identification of informative genes based on analysing the overlap between expressions across two classes (phenotypes). The idea of selecting genes based on this criterion, taking into account the proportions of overlapping observations, is proposed. An algorithm for detecting the minimum subset of genes that provide the maximum number of correctly classified observations in a training set is then introduced.

Chapter 4

Minimum Subset of Genes for Binary Class Problems

4.1 Introduction

Biomedical researchers may be interested in identifying small sets of genes that could be used as genetic markers for diagnostic purposes in clinical research. This typically involves obtaining the smallest possible subset of genes that can still provide a good predictive performance, whilst removing redundant ones (Díaz-Uriarte & De Andres 2006).

A procedure serving this goal is proposed in this chapter. It selects the minimum subset of genes that yield the best classification accuracy on a training dataset avoiding the effects of outliers. The procedure utilizes the interquartile range approach to robustly detect the minimum subset of genes that maximizes the correct assignment of training observations to their corresponding classes.

Microarray data are usually presented in the form of a gene expression matrix, $X = \left[x_{ij}\right]$,

such that $X \in \Re^{P \times N}$ and x_{ij} is the observed expression value of gene i for observation (tissue sample) j where $i = 1, \ldots, P$ and $j = 1, \ldots, N$. Each observation is also characterized by a target class label, y_j, representing the phenotype of the tissue sample being studied. We consider that $Y \in \Re^N$ be the vector of class labels such that its jth element, y_j, has a single value c which is either 1 or 2.

As discussed in Chapter 3, analyzing the overlap between expression intervals of a gene for different classes can provide a classifier with an important aspect of a gene's characteristic. The idea is that a certain gene i can assign observations to class c because their gene i expression interval in that class is not overlapping with gene i interval of the other class. In other words, gene i has the ability to correctly classify observations for which their gene i expressions fall within the expression interval of a single class. For instance, Figure 4.1a presents expression values of gene i_1 with 36 observations belonging to two different classes. It is clear that gene i_1 is relevant for discriminating observations between the target classes, because their values are falling in non-overlapping ranges. Figure 4.1b, on the other hand, shows expression values for another gene i_2, which looks less useful for distinguishing between these target classes, because their expression values have a highly overlapping range.

This chapter proposes a procedure that initially exploits the interquartile range approach to robustly define gene masks that report the discriminative power of genes with a training set of observations avoiding outlier effects. A measure named the proportional overlapping score (*POS*) is defined to measure a gene relevance score that estimates the overlapping degree between the expression intervals of both given classes taking into account three factors: (1) length of overlapping region; (2) number of overlapped observations; (3) the

proportion of a classes' contribution to the overlapped observations. The latter factor is the incentive for the name we gave to our measure. POS measure is assigned for each gene.

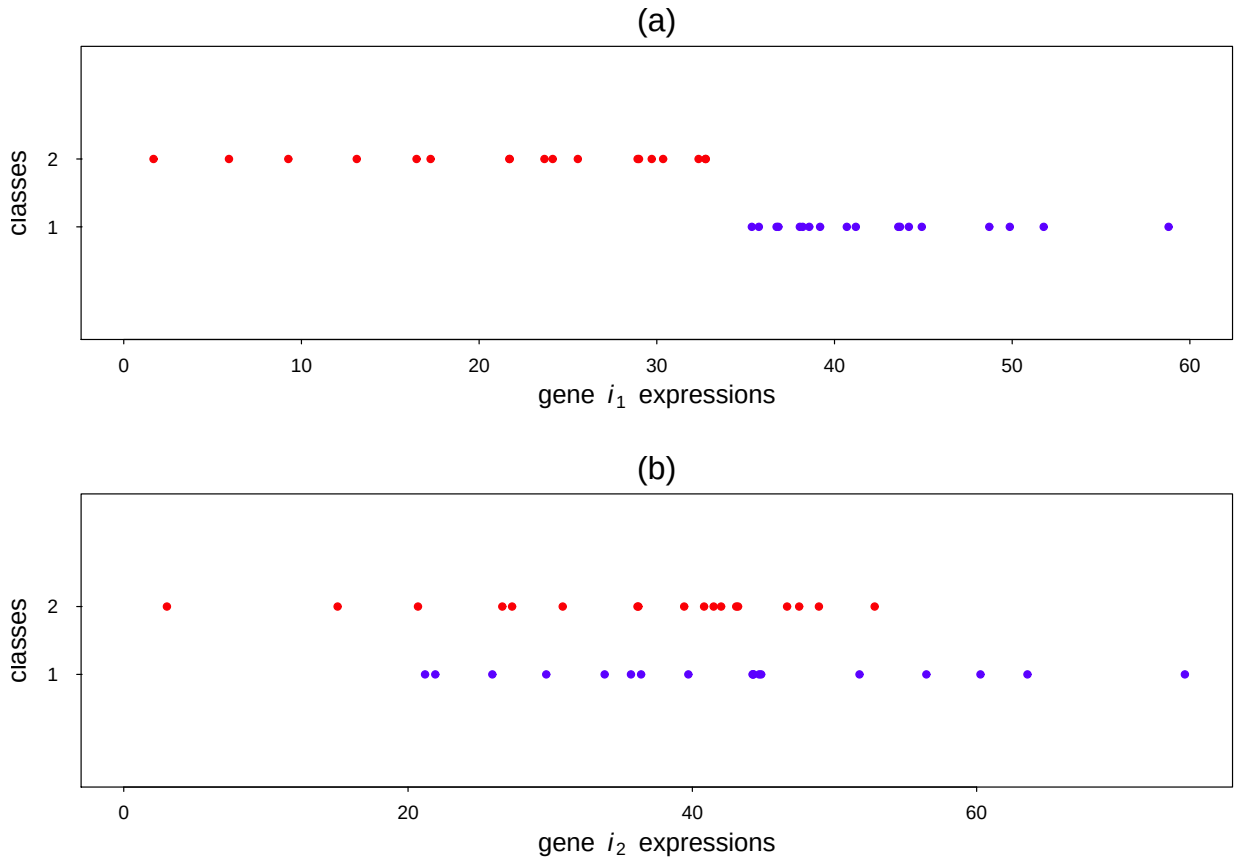

Figure 4.1: *An example for two different genes with different overlapping pattern. Expression values of two different genes (i_1, i_2) each of which with 36 observations belonging to 2 classes, 18 observations for each class:* ***(a)*** *expression values of gene i_1,* ***(b)*** *expression values of gene i_2.*

Baralis et al. (2008) have proposed a method that is somewhat similar to our procedure for detecting a minimum subset of genes from microarray data. The main differences are that Baralis et al. (2008) use the whole expression range to define the intervals which are employed for constructing gene masks, and then apply a set-covering approach to obtain the minimum feature subset. The same technique is performed by Apiletti et al. (2012) to get a minimum gene subset using a greedy approach rather than the set-covering.

4.2 Definition of Core Intervals

For a certain gene i, by considering the expression values x_{ij} with a class label c_j for each observation j, we can define two expression intervals, one for each class, for that gene. The cth class interval for gene i can be defined in the form:

$$I_{i,c} = [a_{i,c}, b_{i,c}], \quad i = 1, \ldots, P, \quad c = 1, 2, \tag{4.1}$$

such that:

$$a_{i,c} = Q_1^{(i,c)} - 1.5\, IQR^{(i,c)}, \; b_{i,c} = Q_3^{(i,c)} + 1.5\, IQR^{(i,c)}, \tag{4.2}$$

where $Q_1^{(i,c)}$, $Q_3^{(i,c)}$ and $IQR^{(i,c)}$ denote the first, third empirical quartiles, and the interquartile range of gene i expression values for class c respectively. Figure 4.2 shows the potential effect of expression outliers on extending the underlying intervals, if the range of training expressions are considered instead. Based on the defined core intervals, we present the following definitions:

Definition 4.1 ***Non-outlier observations set***, $\mathbb{L}_i$, for gene i is defined as the set of observations whose expression values fall inside their own target classes core interval. This set can be expressed as:

$$\mathbb{L}_i = \left\{ j \,:\, x_{ij} \in I_{i,c_j}, \quad j = 1, \cdots, N \right\}, \tag{4.3}$$

where c_j is the observed class label for observation j.

Definition 4.2 ***Total core interval***, I_i, for gene i is given by the region between the global minimum and global maximum boundaries of core intervals for both classes. It is defined

as:

$$I_i = [a_i,\, b_i]\,, \tag{4.4}$$

such that $a_i = min\{a_{i,1},\, a_{i,2}\}$, $b_i = max\{b_{i,1},\, b_{i,2}\}$, where $a_{i,c}$, $b_{i,c}$ respectively represent the minimum and maximum boundaries of core interval, $I_{i,c}$, of gene i with target class $c = 1, 2$, see (4.1) and (4.2).

Definition 4.3 ***The overlap region***, $I_i^{(v)}$, for gene i is defined as the interval yielded by the intersection between core expression intervals of both target classes. It can be addressed as:

$$I_i^{(v)} = I_{i,1} \cap I_{i,2}. \tag{4.5}$$

Definition 4.4 ***Set of non-overlapping observations***, $\mathbb{V}'_i$, for gene i is defined as the set consisting of elements of $\mathbb{L}_i$, defined in Definition 4.1, whose expression values don't fall within the overlap interval $I_i^{(v)}$, defined in Definition 4.3. In this way, we can define this set as:

$$\mathbb{V}'_i = \left\{ j :\ j \in \mathbb{L}_i \ \wedge\ x_{ij} \in I_{i,1} \ominus I_{i,2} \right\}. \tag{4.6}$$

Definition 4.5 ***Set of overlapping observations***, $\mathbb{V}_i$, for gene i is the set containing the observations whose expression values fall within the overlap interval $I_i^{(v)}$, defined in Definition 4.3. The overlapping observations set can be defined as:

$$\mathbb{V}_i = \mathbb{L}_i - \mathbb{V}'_i, \tag{4.7}$$

where $\mathbb{V}'_i$ represents the non-overlapping observations set which is defined as shown in Definition 4.4.

The set of overlapping observations belonging to class c is represented by $\mathbb{V}_{i,c}$ and can be defined as:

$$\mathbb{V}_{i,c} = \left\{ j \,\middle|\, j \in \mathbb{V}_i \ \wedge \ c_j = c \right\}, \tag{4.8}$$

note that $\sum_{c=1}^{2} \left|\mathbb{V}_{i,c}\right| = |\mathbb{V}_i|$. For convenience, $\langle I \rangle$ notation is used with interval I to represent its length while $|.|$ notation is used with set $\{.\}$ to represent its size.

4.3 Gene Masks

For each gene, we define a mask based on its observed expression values and constructed core intervals presented in Section 4.2. Gene i's mask reports the observations that gene i can unambiguously assign to their correct target classes, i.e. the non-overlapping observations set $\mathbb{V}'_i$. Thus, gene masks can represent the capability of genes to classify correctly each observation, i.e. it represents a gene's classification power. For a particular gene i, element j of its mask is set to 1 if the corresponding expression value x_{ij} belongs only to core expression interval I_{i,c_j} of the single class c_j, i.e. if observation j is a member of the set $\mathbb{V}'_i$. Otherwise, it is set to zero.

We define the gene masks matrix $M = \left[m_{ij}\right]$ in which the mask of gene i is presented by $M_{i.}$ (the ith row of M) such that gene mask element m_{ij} is defined as:

$$m_{ij} = \begin{cases} 1 & if \quad j \in \mathbb{V}'_i \\ 0 & otherwise \end{cases}, \quad \begin{matrix} i = 1, \ldots, P \\ j = 1, \ldots, N \end{matrix}. \tag{4.9}$$

Figure 4.2 shows the constructed core expression intervals $I_{i,1}$ and $I_{i,2}$ associated with a particular gene i along-with its gene mask. The gene mask presented in this figure is sorted

corresponding to the observations ordered by increasing expression values.

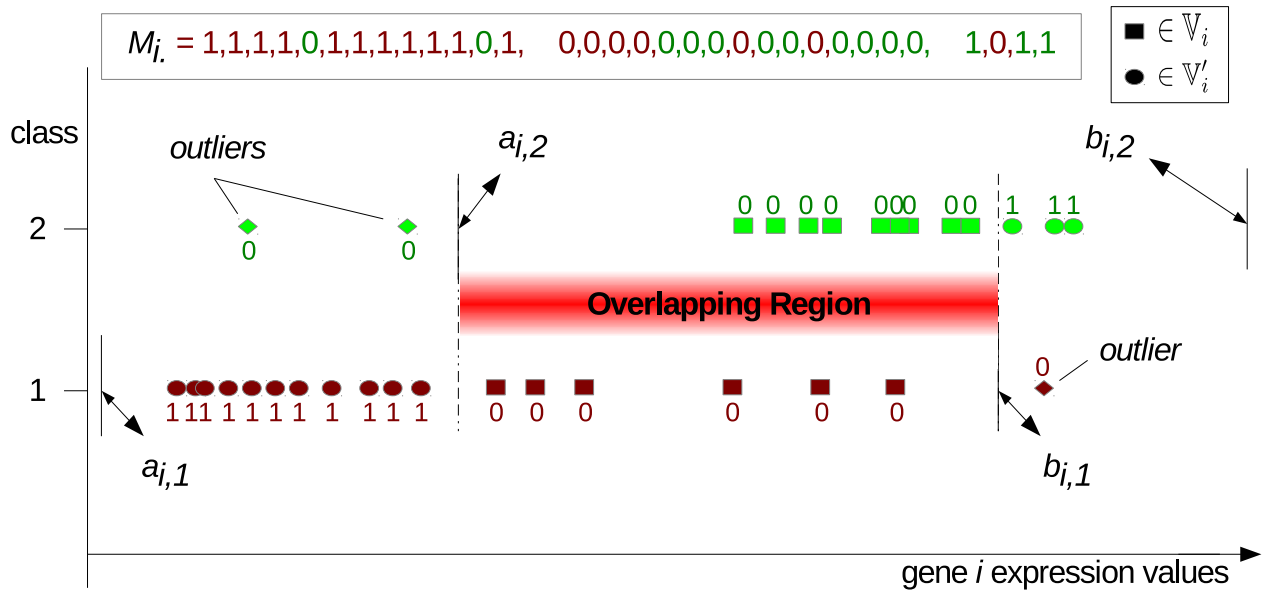

Figure 4.2: *An example for core expression intervals of a gene with* 18 *and* 14 *observations belonging to class 1 and 2 respectively with its associated mask elements.*

4.4 The Proposed *POS* Measure

A novel overlapping score is developed to estimate the overlapping degree between different expression intervals. Figures 4.3a and 4.3b represent examples of 2 different genes, i_1 and i_2, with the same length of overlap interval, $\left\langle I_{i_1}^{(v)} \right\rangle = \left\langle I_{i_2}^{(v)} \right\rangle = \left\langle I_i^{(v)} \right\rangle$, length of total core interval, $\langle I_{i_1} \rangle = \langle I_{i_2} \rangle = \langle I_i \rangle$, and total number of overlapped observations, $\left|\mathbb{V}_{i_1}\right| = \left|\mathbb{V}_{i_2}\right| = 12$. These figures demonstrate that performing the ordinary overlapping scores, proposed in earlier studies (Apiletti et al. 2007*b*, 2012), result in the same value for both genes. But, there is an element which differs in both examples and it may also affect the overlap degree between classes. This element is the distribution of overlapping observations by classes. Gene i_1 has six overlapped observations from each class, whereas gene i_2 has ten and two

overlapping observations from class 1 and 2 respectively. By taking this status into account, gene i_2 should be reported to have less overlap degree compared to gene i_1. I develop a new score, called proportional overlapping score (*POS*), that estimates the overlapping degree of a gene taking into account this element, i.e. proportion of each class's overlapped observations to the total number of overlapping observations.

POS for a gene i is defined as:

$$POS_i = 4 \, \frac{\left\langle I_i^{(v)} \right\rangle}{\langle I_i \rangle} \, \frac{|\mathbb{V}_i|}{|\mathbb{L}_i|} \left(\prod_{c=1}^{2} \theta_c \right), \tag{4.10}$$

where θ_c is the proportion of class c observations among overlapping observations. Hence, θ_c can be defined as:

$$\theta_c = \frac{\left|\mathbb{V}_{i,c}\right|}{|\mathbb{V}_i|}. \tag{4.11}$$

According to (4.10), values of POS measure are $\frac{9}{21}.\frac{\left\langle I_i^{(v)} \right\rangle}{\langle I_i \rangle}$ and $\frac{5}{21}.\frac{\left\langle I_i^{(v)} \right\rangle}{\langle I_i \rangle}$ for genes i_1 and i_2 in figures 4.3a and 4.3b respectively.

Larger overlapping intervals or higher numbers of overlapping observations results in an increasing *POS* value. Furthermore, as proportions θ_1 and θ_2 get closer to each other, the *POS* value increases. The most overlapping degree for a particular gene is achieved when $\theta_1 = \theta_2 = 0.5$ while the other two factors are fixed. We include the multiplier "4" in (4.10) to scale *POS* score to be within the closed interval $[0, 1]$. In this way, a lower score denotes gene with higher discriminative power.

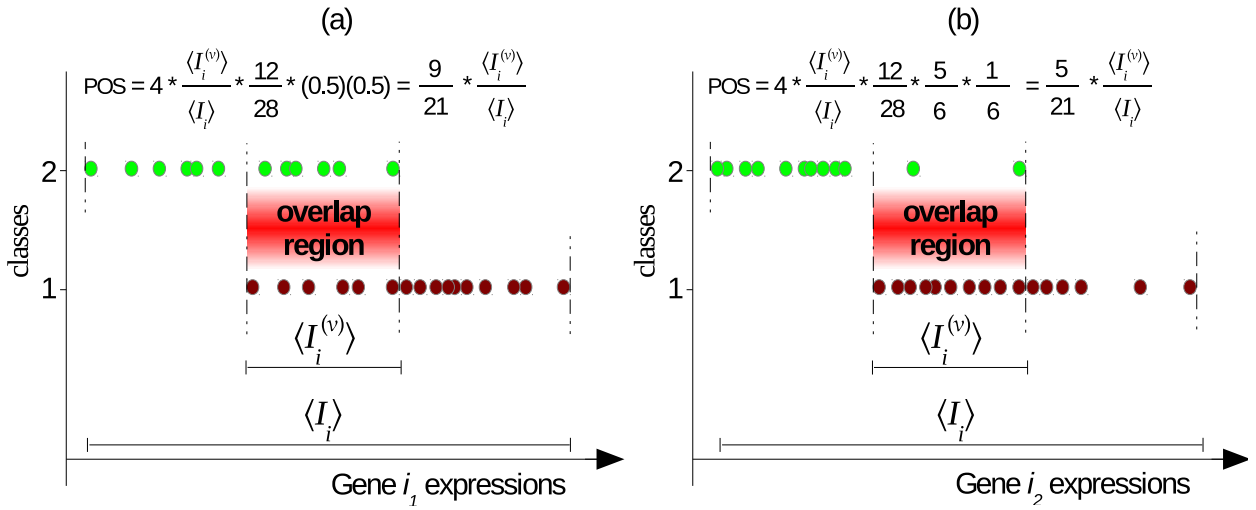

Figure 4.3: *Examples for expression values of 2 genes distinguishing between 2 classes:* ***(a)*** *gene i_1 has overlapping observations distributed as 1:1,* ***(b)*** *gene i_2 has its overlapping observations distributed as 5:1 for class1:class2.*

4.5 Identifying the Minimum Subset of Genes

The information provided by the constructed gene masks, presented in Section 4.3, and the *POS* scores, defined in Section 4.4, are analyzed to identify the minimum subset of genes. This subset is designed to be the minimum one that correctly classify the maximum number of observations in a given training set, avoiding the effects of expression outliers. This procedure also allows disposing of redundant information e.g., genes with similar expression profiles.

The procedure exploits the defined core intervals in (4.1) along-with the *POS* measure, in (4.10), to select the minimum subset of genes that provides the maximum coverage of the training observations. Let $\mathbb{G}$ be a set containing all genes (i.e., $|\mathbb{G}| = P$). Also, let $M_{..}(\mathbb{G})$ be its aggregate mask which is defined as the logical disjunction *(logic OR)* between all masks corresponding to genes that belong to the set. It can be expressed as:

Algorithm 4.1 *Greedy Search - Minimum set of genes*

Inputs: M, $M_{..}(\mathbb{G})$ and POS scores for all genes.
output: $\mathbb{G}^*$.

1: $k = 0$ {Initialization}
2: $\mathbb{G}^* = \emptyset$
3: $M_{..}(\mathbb{G}^*) = \mathbf{0}_N$
4: **while** $M_{..}(\mathbb{G}^*) \neq M_{..}(\mathbb{G})$ **do**
5: $k = k + 1$
6: $\mathbb{S}_k = \underset{i \in \mathbb{G}}{argmax} \left(\sum_{j=1}^{N} I\left(m_{ij} = 1\right) \right)$ {Assign gene set whose masks have the max. bits of 1}
7: $g_k = \underset{i \in \mathbb{S}_k}{argmin} (POS_i)$ {Select the candidate with the best score among the assigned set}
8: $\mathbb{G}^* = \mathbb{G}^* + g_k$ {Update the target set by adding the selected candidate}
9: **for all** $i \in \mathbb{G}$ **do**
10: $M_{i.}^{(k+1)} = M_{i.}^{(k)} \wedge !M_{..}(\mathbb{G}^*)$ {update gene masks such that the uncovered samples are only considered}
11: **end for**
12: **end while**
13: **return** $\mathbb{G}^*$

$$M_{..}(\mathbb{G}) = \bigvee_{i \in \mathbb{G}} M_{i.} = M_{1.} \vee \ldots \vee M_{P.} \tag{4.12}$$

The objective is to search for the minimum subset, denoted by $\mathbb{G}^*$, for which $M_{..}(\mathbb{G}^*)$ equals to the aggregate mask of the set of genes, $M_{..}(\mathbb{G})$. In other words, the minimum set of genes should satisfy the following statement:

$$\underset{\mathbb{G}^* \subseteq \mathbb{G}}{argmin} \left(|\mathbb{G}^*| \,\middle|\, \left(M_{..}(\mathbb{G}^*) = \bigvee_{i \in \mathbb{G}^*} M_{i.} = M_{..}(\mathbb{G}) \right) \right). \tag{4.13}$$

A modified version of the greedy search approach used by Apiletti et al. (2012) is applied. The pseudo code of our procedure is reported in Algorithm 4.1. Its inputs are the matrix of gene masks, M; the aggregate mask of genes, $M_{..}(\mathbb{G})$; and POS scores. It produces the minimum set of genes, $\mathbb{G}^*$, as output.

At the initial step ($k = 0$), we let $\mathbb{G}^* = \emptyset$ and $M_{..}(\mathbb{G}^*) = \mathbf{0}_N$ (lines 2,3); where $M_{..}(\mathbb{G}^*)$ is the aggregate mask of the set $\mathbb{G}^*$, while $\mathbf{0}_N$ is a vector of zeros with the length N. Then, at each iteration, k, the following steps are performed:

1. The gene(s) with the highest number of mask bits set to 1 is (are) chosen to form the set $\mathbb{S}_k$ (line 6). This set could not be empty as long as the loop condition is still satisfied, i.e. $M_{..}(\mathbb{G}^*) \neq M_{..}(\mathbb{G})$. Under this condition, our selected genes don't cover the maximum number of observations that should be covered by the target gene set. Note that the definition for gene masks allows $M_{..}(\mathbb{G})$ to report in advance which observations should be covered by the minimum subset of genes. Therefore, there will be at least one gene mask which has at least one bit set to 1 if that condition is to hold.

2. The gene with the lowest *POS* score among those genes in $\mathbb{S}_k$, if it includes more than one gene, is then selected (line 7). It is denoted by g_k.

3. The set $\mathbb{G}^*$ is updated by adding the selected gene g_k (line 8).

4. All gene masks are updated by performing the logical conjunction (*logic AND*) with negated aggregate mask of set $\mathbb{G}^*$ (line 10). The negated mask $!M_{..}(\mathbb{G}^*)$ of the mask $M_{..}(\mathbb{G}^*)$ is the one obtained by applying logical negation (logical complement) on this mask. Consequently, the bits of ones corresponding to the classification of still uncovered observations are only considered. Note that $M_{i.}^{(k)}$ represents an updated mask of gene i at the kth iteration, $M_{i.}^{(1)}$ is the original gene i's mask whose elements are computed according to (4.9).

5. The procedure is iterated and ends when all updated gene masks have no 1 bits

anymore, i.e. the selected genes cover the maximum number of observations. This situation is accomplished iff $M_{..}(\mathbb{G}^*) = M_{..}(\mathbb{G})$.

This procedure detects the minimum set of genes required to provide the best classification coverage for a given training set. In addition, genes are descendingly ordered by number of 1 bits within the minimum set $\mathbb{G}^*$.

4.6 Summary

The idea of selecting genes based on analysing the overlap between their expressions across two classes (phenotypes), taking into account the proportions of overlapping observations, is considered. To this end, intervals of core gene expressions are defined. A gene mask that allows reporting a gene's predictive power avoiding the effects of outliers is robustly constructed for each gene. A novel score, named the Proportional Overlapping Score (*POS*), is then proposed by which a gene's overlapping degree is estimated. The constructed gene masks along-with the gene scores are utilized to assign the minimum subset of genes that provide the maximum number of correctly classified observations in a training set.

The next chapter proposes a gene selection method, named POS, by combining the minimum subset with the top ranked genes according to the *POS* measure to produce a final gene selection. A novel measure for assigning each gene to its relative dominant class (RDC) is also proposed.

Chapter 5

Proportional Overlapping Score Method for Gene Selection

For a given classification problem, finding the optimal number of genes being selected is a challenge. These is a trade-off between information loss, when selecting excessively conservative number, and noise increase, when selecting an excessively liberal number. The procedure presented in Chapter 4 addresses the identification of minimum subset of genes that provide the maximum classification coverage for training dataset based on analyzing overlap in expressions between different classes. The procedure stops the search when the coverage of training observations is maximized. In this chapter, a filter feature selection method, named 'Proportional Overlapping Score' (POS) method, based on the procedure proposed in Chapter 4 is presented. POS selects the most discriminative features according to their expression overlapping degree. In addition, it allows users to select the number of retrieved features and thus provides researchers with the possibility of studying a large number of relevant genes for the target disease.

The POS method is as follows:

- POS initially detects the minimum subset of genes that maximizes the correct assignment of training observations to their corresponding classes avoiding the effects of outliers. The approach presented in Chapter 4 is used for this purpose.

- A new filter-based technique which ranks genes according to their predictive power in terms of the overlapping degree between classes is proposed. In this context, the novel *POS* measure, defined in (4.10), is used.

- POS categorizes genes into the target class labels based on their relative dominant classes. In this context, POS presents a novel measure, called 'Relative Dominant Class' (RDC) measure, by which each gene is assigned to the class label that has the highest proportion, relative to class sizes, of correctly assigned observations.

- The final gene selection is produced by extending the minimum subset of genes with the topmost ranked genes according to the *POS* and *RDC* measures.

The performance of POS method is validated on various benchmarking microarray datasets. The proposed method is compared with several widely used gene selection methods: Iteratively Sure Independent Screening (ISIS) (Fan et al. 2009); Wilcoxon rank-sum approach; mRMR (Ding & Peng 2005); MaskedPainter (Apiletti et al. 2012). The classification error rates of the RF (Breiman 2001), *k*NN (Cover & Hart 1967), and SVM (Cortes & Vapnik 1995) classifiers, discussed in Sections 2.4 - 2.6 respectively, demonstrate that the proposed approach achieves a better performance than the other compared methods.

5.1 The Method

POS method exploits the *POS* measure, defined in (4.10), to rank features based on their expression overlapping between classes, the higher ranked features are the ones with lower *POS* scores and hence are more informative for distinguishing between the considered target classes. *POS* score alone can rank genes according to their overlapping degree, without taking into account the class that has more correctly assigned observations by each gene (the dominant class for that gene). Consequently, high-ranked genes may all have an ability to only correctly classify observations belonging to the same class. Such a case is more likely to happen in situations with unbalanced class-size distributions. As a result, a biased selection could result. Assigning the dominant class on a relative basis, and taking these assignments into account during the gene ranking process allows us to overcome this problem.

5.1.1 Relative Dominant Class Assignments

Gene masks defined in Section 4.3 are used to assign each gene to its relative dominant class (RDC). For a gene i, RDC_i is defined as follows:

$$RDC_i = \underset{c}{argmax} \left(\frac{\sum_{j \in \mathbb{U}_c} I\left(m_{ij} = 1\right)}{|\mathbb{U}_c|} \right), \tag{5.1}$$

where $\mathbb{U}_c$ is the set of class c observations (i.e., $\mathbb{U}_c = \left\{ j \middle| c_j = c \right\}$). Note that $\sum_c |\mathbb{U}_c| = N$, while m_{ij} is the jth mask element of gene i that is defined in (4.9). $I\left(m_{ij} = 1\right)$ represents an indicator which sets to 1 if $m_{ij} = 1$, otherwise it sets to zero.

In this definition, the observations that belong to the set $\mathbb{V}'_i$, described in Definition 4.4, are only considered for each class. These observations are the ones that could be unambiguously assigned to their target classes by gene i. According to the gene mask defined in (4.9), they are the observations with 1 bits in their corresponding gene mask. The proportion of the class's observations to its total sample size is then evaluated. The class with the highest proportion is the Relative Dominant Class of the considered gene. Ties are randomly distributed on both classes. Genes are assigned to their *RDC* in order to associate each gene with the class it is more able to distinguish. As a result, the number of produced genes could be balanced per class at the final selection process of the POS method when the RDC is taken into account. The RDC can avoid misleading assignments due to unbalanced class sizes distribution effects, because it detects the dominant class of a gene based on a relative role.

5.1.2 Final Gene Selection

The gene ranking process is performed by considering both *POS* scores and *RDC*. Within each Relative Dominant Class c (where $c = 1, 2$), all genes that have not been chosen in the minimum set, $\mathbb{G}^*$ detected by Algorithm 4.1, and whose $RDC = c$ are sorted by an increasing order of *POS* values. Thus, given two disjoint groups (one for each class) of ranked genes, the topmost gene is selected from each group in a round-robin fashion to compose the gene ranking list.

The minimum subset of genes, $\mathbb{G}^*$, is extended by adding the top v ranked genes in the gene ranking list, where v is the required number extending the minimum subset up to the total number of requested genes, r, where r is an input of the POS method set by

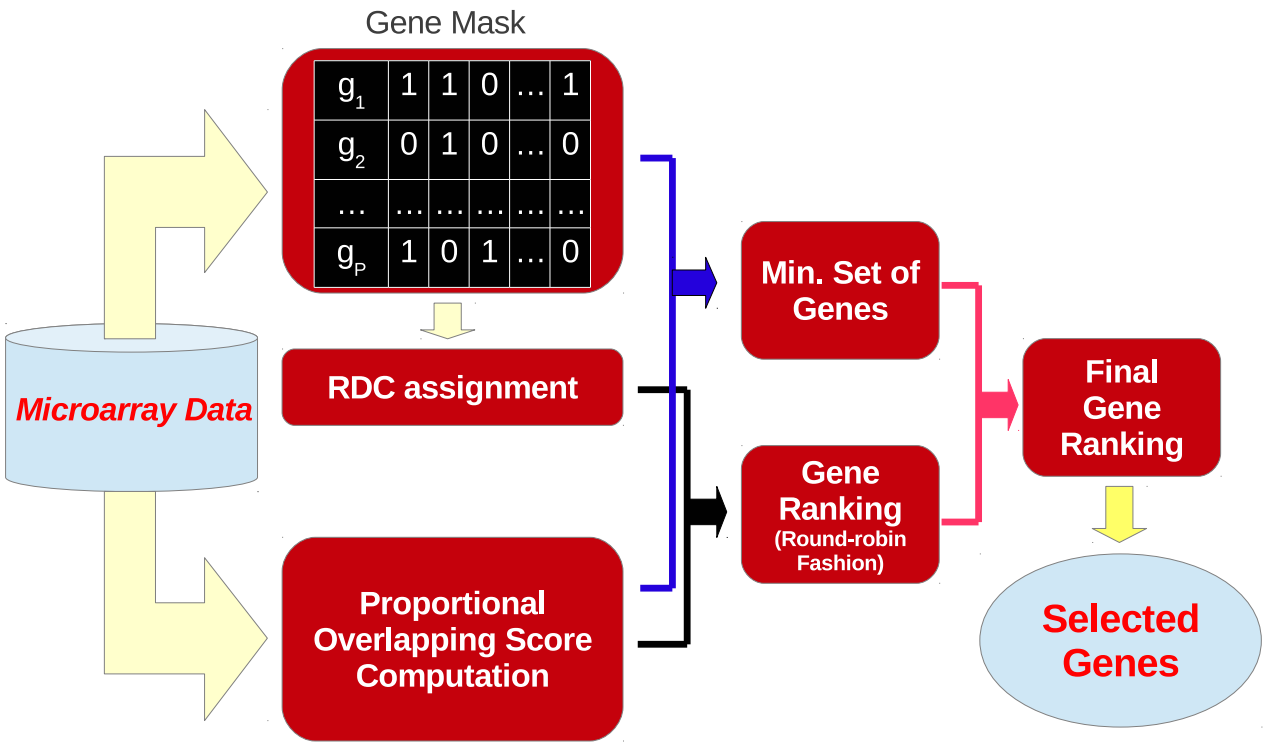

Figure 5.1: *Building blocks of POS method.*

the user. The resulting final set includes the minimum subset of genes regardless of their *POS* values, because these genes allow the considered classifier to correctly classify the maximum number of training observations.

Figure 5.1 shows the building blocks of the POS approach. The pseudo code of POS method is reported in Algorithm 5.1.

5.1.3 Illustrative Example

To illustrate the steps of POS method, presented in Algorithm 5.1, consider the set of genes represented in Figure 5.2(a). Each gene is associated with its constructed gene mask, its proportional overlapping score (*POS*), and its relative dominant class (*RDC*). For instance,

Algorithm 5.1 *POS Method For Gene Selection*

Inputs: X, Y and number of selected genes (r).
Output: Sequence of the selected genes $\mathbb{T}$.

for all $i \in \mathbb{G}$ **do**
  for $c = 1$ **to** 2 **do**
    Calculate $I_{i,c}$ as defined in (4.1).
  end for
  for $j = 1$ **to** N **do**
    Compute m_{ij} as defined in (4.9).
  end for
  Compute POS_i as defined in (4.10) and (4.11).
  Assign RDC_i as defined in (5.1).
end for
Let $M \in \Re^{P \times N}$ be the gene mask matrix, where $M = \left[m_{ij}\right]$.
Obtain $M_{..}(\mathbb{G})$ as defined in equation 4.12. {aggregate mask of genes}
Use the Greedy Search approach, presented in algorithm 4.1, with input set includes M, $M_{..}(\mathbb{G})$, and POS_i, $i = 1, \ldots, P$, to output the minimum subset of genes, $\mathbb{G}^*$.
$\mathbb{G} = \mathbb{G} - \mathbb{G}^*$. {exclude the minimum subset from the set of genes}
for $c = 1$ **to** 2 **do**
  Let $\mathbb{G}_c = \left\langle g_{ck} : g_{ck} \in \mathbb{G},\ RDC_{g_{ck}} = c \right\rangle$ be a sequence of genes such that $POS_{g_{ck}} \leq POS_{g_{c(k+1)}}$, where g_{ck} denotes gene in the kth rank in sequence $\mathbb{G}_c$. {define the sequence of genes sorted by an increasing order of *POS* values within the *RDC* class c}
end for
Getting the Final Gene Ranking
if $r \leq |\mathbb{G}^*|$ **then**
  $\mathbb{T}$ is the set whose members are the first r genes in $\mathbb{G}^*$.
else
  $\mathbb{T} = \mathbb{G}^*$. {initially get the minimum set in our final gene ranking}
  while $\left|\mathbb{T}\right| < r$ **do**
    Extend $\mathbb{T}$ by one gene using round-robin fashion applying on the sequences $\mathbb{G}_1$and $\mathbb{G}_2$.
  end while
end if
return $\mathbb{T}$

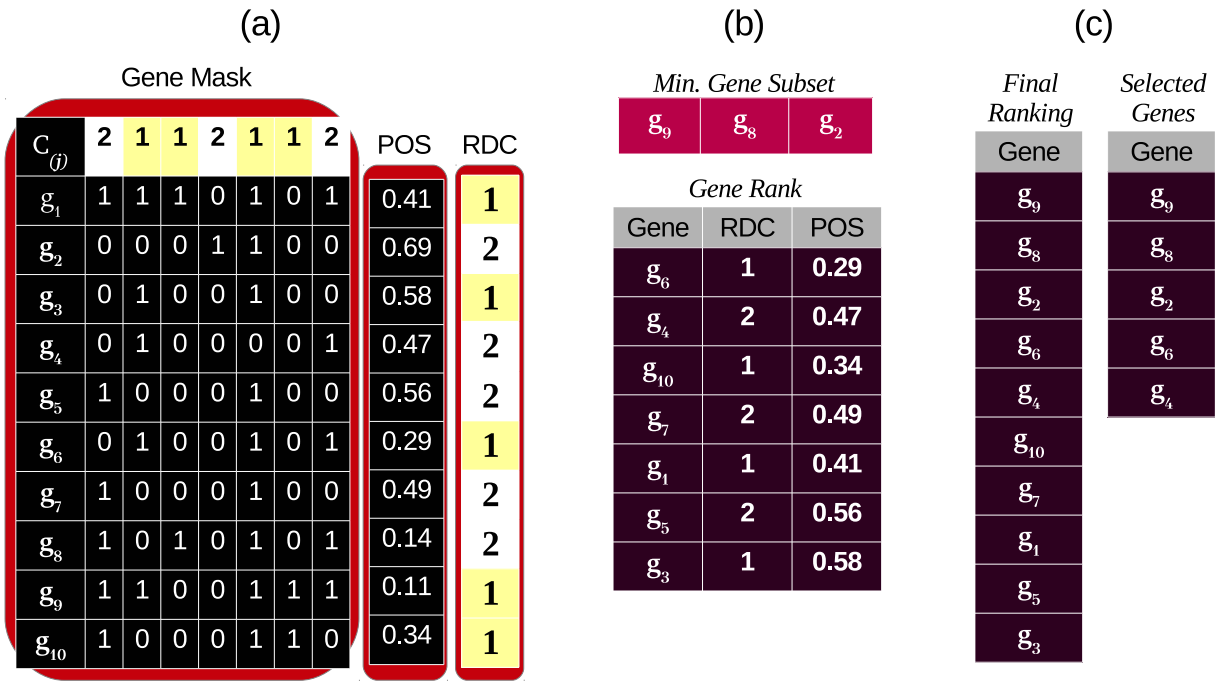

Figure 5.2: *An example of the POS method: (a) genes with their masks, proportional overlapping scores, and relative dominant classes; (b) minimum gene subset obtained by Algorithm 4.1, and gene list ranked by POS and RDC; (c) final ranking, and selected genes at the end of the process.*

gene g_1 has a mask of [1110101], i.e. it classifies unambiguously all training observations except the fourth and the sixth observations, a proportional overlapping score equals 0.41, and its relative dominant class is 1. The first member of the minimum gene subset, selected by the Algorithm 4.1, is g_9, because it is characterized by the highest number of mask elements set to 1 (the same as g_1) and the lowest *POS* (see Algorithm 4.1: lines 6 and 7).

Genes with updated masks are considered to focus only on uncovered observations by the selected gene, g_9, (i.e., the third and forth observations). The best updated masks (i.e., g_1, g_2, and g_8 which all have the same number of 1 bits) are then considered. Again, g_8 is selected as the second member of the minimum gene subset because of its lowest *POS* score. Eventually, the only gene with complementary mask having 1 bit, which is g_2, is selected to finalize the minimum subset. In this example, the minimum number of genes is three. Figure 5.2(b) reports the chosen minimum subset.

The remaining genes are categorized by relative dominant class (*RDC*) and sorted according to *POS* in an ascending order within each category of *RDC*. The procedure of gene ranking is accomplished by selecting the topmost gene from each category of *RDC* in a round-robin fashion (e.g., g_6 from the class 1 category, followed by g_4 from class 2, then g_{10} from class 1, etc.) as shown in Figure 5.2(b).

If I suppose that $r = 5$, then the two top ranked genes (i.e., g_6 and g_4) are added to the selected minimum subset of genes (three genes). The final ranking and the final selection are shown in Figure 5.2(c).

5.2 Results

For evaluating different feature selection methods, one can assess the accuracy of a classifier applied after the feature selection process. Thus, the classification is based only on selected gene expressions. Such an assessment can verify the efficiency of identification of discriminative genes. Jirapech-Umpai & Aitken (2005) have analyzed several gene selection methods available in Su et al. (2003) and have shown that the gene selection method can have a significant impact on a classifier's accuracy. Such a strategy has been applied in many studies including Apiletti et al. (2012) and Peng et al. (2005).

In this section, an experiment is conducted using eleven gene expression datasets in which the POS method is validated by comparison with five well-known gene selection techniques. The performance is evaluated by obtaining the classification error rates from three different classifiers: Random Forest; k Nearest Neighbor; Support Vector Machine.

Table 5.1 summarizes the characteristics of the datasets. The estimated classification

error rate is based on the Random Forest classifier with the full set of features, without pre-selection, using 50 repetitions of 10-fold cross validation.

Eight of the datasets are bi-class, while three, i.e. Srbct, GSE14333 and GSE27854, are multi-classes. The two classes with topmost number of observations are only considered for the Srbct data, while the remaining classes are ignored, since this thesis is interested only in binary classification analysis. For the GSE14333 data, patients with colorectal cancer of Duck stages A and B are combined in a single class representing non-invasive tumors, against patients with stage C, which represents invasive tumors. Whereas for the GSE27854 data, a class composed of colorectal cancer patients with tumor 'Union Internationale Contre le Cancer' (UICC) stages I and II is defined against another class involving patients with III and IV stages. The sources of Microarray data are corrected, normalized and summarized using 'Frozen robust multiarray analysis' (fRMA) method (McCall et al. 2010). All datasets are publicly available. The availability of the datasets are reported in Appendix A.

Table 5.1: Description of used gene expression datasets

Dataset	*Genes*	*Samples*	*Class-sizes*	*Est. Error*	*Source*
Leukaemia	7129	72	47/25	0.049	Golub et al. (1999*a*)
Breast	4948	78	34/44	0.369	Michiels et al. (2005)
Srbct	2308	54	29/25	0.0008	Statnikov et al. (2005)
Prostate	10509	102	52/50	0.088	Statnikov et al. (2005)
All	12625	128	95/33	0.000	Chiaretti et al. (2004)
Lung	12533	181	150/31	0.003	Gordon et al. (2002)
Carcinoma	7457	36	18/18	0.027	Notterman et al. (2001)
GSE24514	22215	49	34/15	0.0406	Alhopuro et al. (2012)
GSE4045	22215	37	29/8	0.2045	Laiho et al. (2007)
GSE14333	54675	229	138/91	0.4141	Jorissen et al. (2009)
GSE27854	54675	115	57/58	0.4884	Kikuchi et al. (2013)

Fifty repetitions of 10-fold cross validation analysis were performed for each combina-

tion of dataset, feature selection algorithm, and a given number of selected genes, up to 50, with the considered classifiers. Random Forest is implemented using the R package 'randomForest' with its default parameters, i.e. ntree, mtry and nodesize are 500, $\sqrt{r}$ and 1 respectively (Liaw & Wiener 2002). The R packages 'class' (Venables & Ripley 2002) and 'e1071' (Meyer et al. 2014) are used to perform the k Nearest Neighbor and Support Vector Machine classifiers respectively. The parameter k for kNN classifier is chosen to be $\sqrt{N}$ rounded to the nearest odd number, where N is the total number of observations (tissue samples). For each experimental repetition, the split seed was changed while the same folds and training datasets were kept for all feature selection methods. To avoid bias, gene selection algorithms have been performed only on the training sets. For each fold, the best subset of genes has been selected according to the Wilcoxon Rank Sum technique (Wil-RS), Minimum Redundancy Maximum Relevance (mRMR) method, MaskedPainter (MP), Iteratively Sure Independent Screening (ISIS), along-with the proposed method, POS. The expressions of the selected genes as well as the class labels of the training observations have then been used to construct the considered classifiers. The classification error rate on the test set is separately reported for each classifier and the average error rate over all the fifty repetitions is then computed. Due to limitations of the R package 'mRMRe' (De Jay et al. 2013), mRMR selections could not be conducted for datasets having more than '46340' features. Therefore, mRMR method is excluded from the analysis of the 'GSE14333' and 'GSE27854' datasets.

The compared feature selection methods are used commonly within the microarray data analysis domain. Apiletti et al. (2012) demonstrate that the MaskedPainter method has outperformed many widely used gene selection methods available in Su et al. (2003).

The mRMR technique, proposed by Ding & Peng (2005), is intensively used in microarray data analysis (e.g., De Jay et al. 2013, Ma et al. 2013). The ISIS feature selection method exploits the principle of correlation ranking with its 'sure independence screening' property showed in Fan & Lv (2008) to select a set of features based on an iterative process. In our experiment, the ISIS technique has been applied using the 'SIS' R package (Fan et al. 2014).

For a large enough input feature set, effective classifier algorithms may have more ability to mitigate the potential effects of noisy and uninformative features by focusing more on the informative ones. For instance, the Random Forest algorithm employs an embedded feature selection procedure that results in less reliance on uninformative input features. In other words, selecting a large number of features may allow a classifier to compensate for potential feature selection shortcomings. For the purpose of comparing the effectiveness of the considered feature selection techniques in improving the classification accuracy, the experiment is designed to focus on small sets of selected features, up to 50 genes.

Tables 5.2 and 5.3 show the average classification error rates obtained by Wil-RS, mRMR, MP and POS with RF, *k*NN and SVM classifiers on Leukaemia and GSE24514 datasets respectively. Each row provides the average classification error rate at a specific number of selected genes, reported in the first column. The aggregate average error value and the minimum error rate for each method with each classifier are provided in the last two rows. Average error rates yielded on the Breast and Srbct datasets using RF, *k*NN, and SVM classifiers are shown in Figure 5.3.

Table 5.2: Average classification error rates yielded by Random Forest, k Nearest Neighbors and Support Vector Machine classifiers on 'Leukaemia' dataset over all the 50 repetitions of 10-fold cross validation

N.genes	RF				*k*NN				SVM			
	Wil-RS	**mRMR**	**MP**	**POS**	**Wil-RS**	**mRMR**	**MP**	**POS**	**Wil-RS**	**mRMR**	**MP**	**POS**
1	0.126	0.211	0.015	**0.003**	0.141	0.220	0.019	**0.005**	0.133	0.238	0.022	**0.005**
2	0.083	0.197	0.017	**0.001**	0.110	0.195	0.059	**0.047**	0.099	0.197	0.053	**0.026**
3	0.068	0.185	0.020	**0.003**	0.086	0.198	**0.070**	0.073	0.078	0.198	0.064	**0.044**
4	0.044	0.180	0.016	**0.001**	0.082	0.194	0.076	**0.069**	0.068	0.178	0.070	**0.050**
5	0.043	0.168	0.015	**0.002**	0.077	0.191	0.084	**0.075**	**0.060**	0.172	0.079	**0.060**
6	0.037	0.170	0.018	**0.005**	0.074	0.188	0.087	**0.065**	**0.052**	0.171	0.082	0.065
7	0.036	0.161	0.018	**0.004**	0.077	0.182	0.090	**0.065**	**0.049**	0.162	0.086	0.069
8	0.035	0.158	0.020	**0.004**	0.081	0.186	0.092	**0.063**	**0.047**	0.166	0.090	0.074
9	0.032	0.161	0.015	**0.003**	0.082	0.176	0.090	**0.067**	**0.049**	0.162	0.092	0.083
10	0.031	0.157	0.018	**0.003**	0.078	0.181	0.094	**0.067**	**0.050**	0.159	0.092	0.079
20	0.030	0.141	0.028	**0.001**	0.085	0.162	0.102	**0.064**	**0.062**	0.145	0.088	0.068
30	0.030	0.131	0.029	**0.001**	0.085	0.155	0.108	**0.070**	**0.058**	0.139	0.093	0.066
40	0.031	0.118	0.031	**0.000**	0.084	0.142	0.105	**0.078**	**0.053**	0.127	0.094	0.069
50	0.031	0.119	0.029	**0.001**	0.083	0.135	0.107	**0.078**	**0.049**	0.126	0.101	0.062
Avg.	0.041	0.157	0.021	**0.002**	0.087	0.179	0.085	**0.063**	0.065	0.167	0.079	**0.059**
Min.	0.030	0.118	0.015	**0.000**	0.074	0.135	0.019	**0.005**	0.047	0.126	0.022	**0.005**

Boldface numbers indicate the minimum average of classification error rates (the highest accuracy) achieved with the corresponding classifier at each size of selected gene sets, reported in the first column.

Table 5.3: Average classification error rates yielded by Random Forest, k Nearest Neighbors and Support Vector Machine classifiers on 'GSE24514' dataset over all the 50 repetitions of 10-fold cross validation

N.genes	RF				*k*NN				SVM			
	Wil-RS	**mRMR**	**MP**	**POS**	**Wil-RS**	**mRMR**	**MP**	**POS**	**Wil-RS**	**mRMR**	**MP**	**POS**
1	0.163	0.352	0.182	**0.090**	0.125	0.304	0.147	**0.096**	0.116	0.274	0.141	**0.085**
2	0.108	0.267	0.143	**0.082**	0.086	0.249	0.117	**0.074**	0.085	0.250	0.108	**0.080**
3	0.098	0.219	0.116	**0.068**	0.077	0.223	0.093	**0.068**	0.075	0.215	0.087	**0.067**
4	0.079	0.186	0.121	**0.067**	0.078	0.186	0.082	**0.065**	0.068	0.185	0.077	**0.063**
5	0.074	0.166	0.103	**0.059**	0.072	0.166	0.070	**0.063**	**0.062**	0.166	0.071	**0.062**
6	0.067	0.147	0.090	**0.058**	0.066	0.155	0.068	**0.059**	**0.060**	0.149	0.064	**0.060**
7	0.065	0.137	0.074	**0.058**	**0.059**	0.142	0.064	0.060	**0.059**	0.135	0.061	0.061
8	0.064	0.128	0.068	**0.052**	**0.057**	0.133	0.060	0.058	0.056	0.126	0.057	**0.054**
9	0.063	0.115	0.075	**0.055**	**0.052**	0.127	0.061	0.057	0.053	0.113	0.052	**0.050**
10	0.063	0.104	0.066	**0.051**	**0.048**	0.116	0.058	0.058	0.050	0.105	**0.047**	0.048
20	0.058	0.076	0.047	**0.037**	**0.032**	0.088	0.048	0.050	0.044	0.078	0.041	**0.039**
30	0.057	0.067	0.039	**0.034**	**0.035**	0.071	0.041	0.043	0.042	0.070	0.038	**0.034**
40	0.057	0.073	0.040	**0.034**	**0.037**	0.063	**0.037**	0.042	0.041	0.069	**0.037**	**0.037**
50	0.055	0.063	0.038	**0.032**	**0.036**	0.041	**0.036**	0.039	0.041	0.059	0.038	**0.036**
Avg.	0.077	0.150	0.086	**0.055**	0.061	0.147	0.070	**0.059**	0.061	0.142	0.066	**0.055**
Min.	0.055	0.063	0.038	**0.032**	**0.032**	0.041	0.036	0.039	0.041	0.059	0.037	**0.034**

Boldface numbers indicate the minimum average of classification error rates (the highest accuracy) achieved with the corresponding classifier at each size of selected gene sets, reported in the first column.

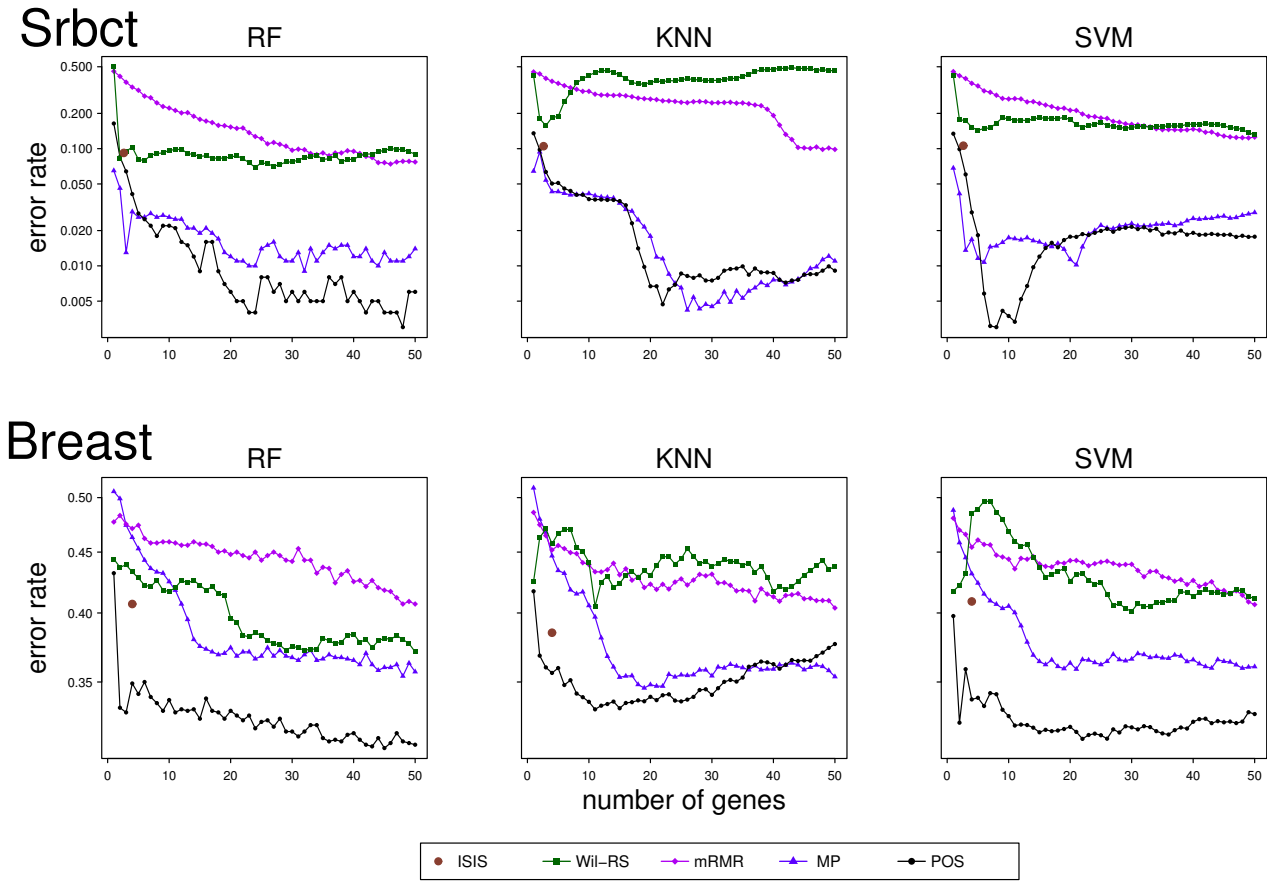

Figure 5.3: *Average classification error rates for 'Srbct' and 'Breast' data based on 50 repetitions 10-fold CV using ISIS, Wil-RS, mRMR, MP and POS methods.*

The Proportional Overlapping Scores (POS) approach yields a good performance with different classifiers on all datasets. For the Random Forest classifier, in particular on Leukaemia, Breast, GSE24514 and GSE4045 datasets, the classification average error rates on the test sets are less than all other feature selection techniques at all selected genes set sizes. On the Srbct, All and Lung datasets, the POS method provides lower error rates than all other methods on most set sizes. While, on the Prostate dataset, POS shows a comparable performance with the best technique, MP. On the Carcinoma dataset, Wil-RS technique has outperformed all methods for feature set sizes more than 20 genes, whereas for smaller sets, the MP method was the best. More details of the RF classifier's results are

reported in Appendix B.1.

For the *k*NN classifier, POS provides a good classification performance. Its classification average error rates are less than all other compared methods on Leukaemia and Breast datasets for most selected set sizes, see Table 5.2 and Figure 5.3. A similar case has been observed in the Lung dataset. On the GSE24514 dataset, Wil-RS technique has outperformed all methods for set sizes that are more than eight, whereas for smaller sets, the POS was the best. While, on Srbct and GSE4045 datasets, POS shows a comparable and a worse performance respectively compared with the best techniques, MP and Wil-RS respectively. More details of the *k*NN classifier's results are reported in Appendix B.2.

For the SVM classifier, POS provides a good classification performance on all used datasets. In particular on Breast and Lung datasets, the classification average error rates on the test sets are less than all other feature selection techniques at all selected genes set sizes, see Figure 5.3 and Table B.18 in Appendix B.3. The performance of POS outperformed all other compared methods on the GSE24514 and Srbct datasets for almost all feature set sizes, see Table 5.3 and Figure 5.3. On Leukaemia and GSE4045 datasets, POS is outperformed by other methods for set sizes more than five and 20 respectively. More details of the SVM classifier's results are reported in the Appendix B.3.

5.2.1 POS Method Quality Performance

The improvement/deterioration in the classification accuracy is analyzed in order to investigate the quality performance of the proposal against the other techniques when the size of the selected gene set varies. The log ratio between the misclassification error rates of the candidate set selected by the best method of the compared techniques and the POS

method is separately computed for each classifier on different set sizes up to 50 genes. At each set size, the best method of the compared techniques is identified and the log ratio, computed using natural logarithm, between its error rate and corresponding error rate of the POS method is reported. Figure 5.4 shows the results for each classifier. Positive values indicate improvements of a classification performance achieved by the POS method over the second best technique. The panel on right bottom of Figure 5.4 shows the averages of log ratios across all considered datasets for each classifier.

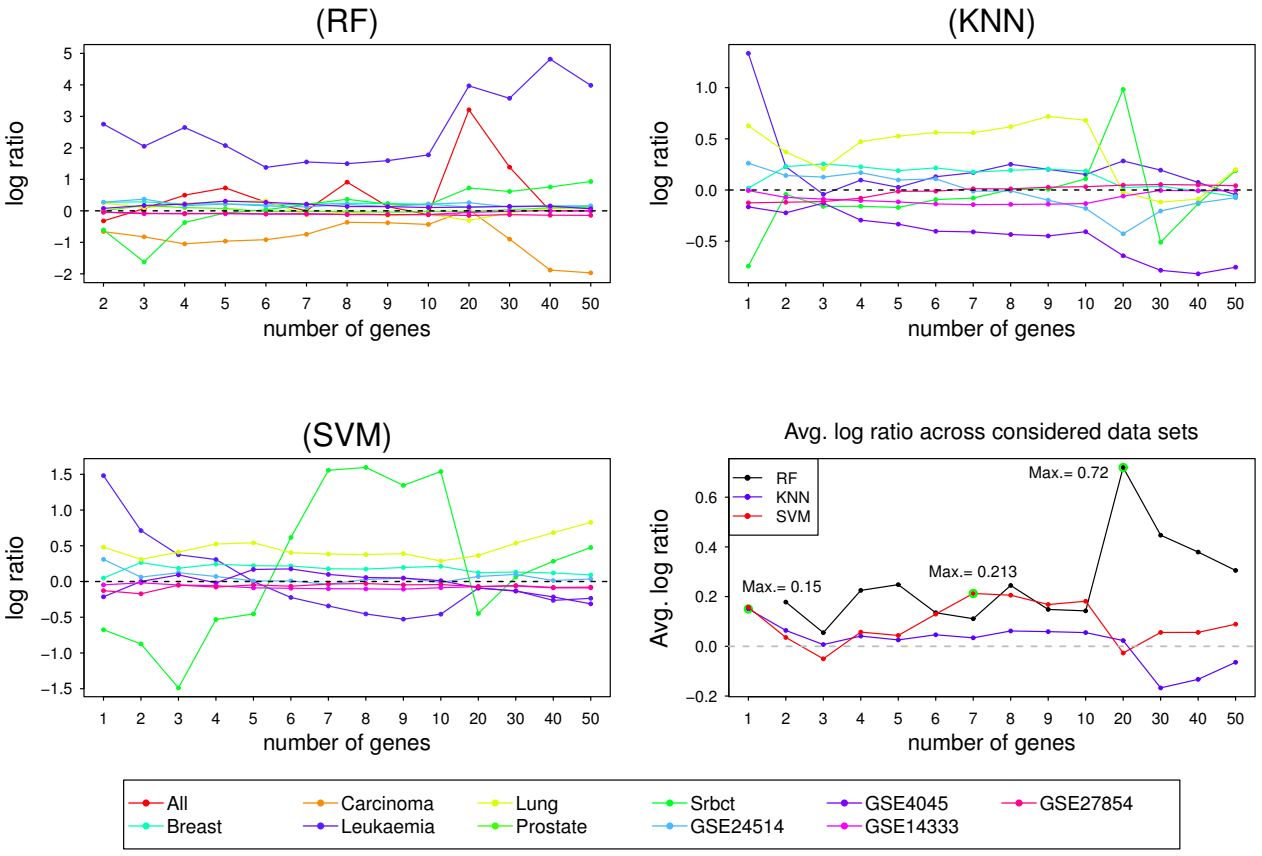

Figure 5.4: *Log ratios measure the improvement/deterioration achieved by the POS method over the best compared method for three different classifiers; RF, kNN and SVM. The last panel shows the averages of log ratios across all datasets for each classifier.*

The POS approach provides improvements over the best method of the compared techniques for most datasets with all classifiers (see RF, *k*NN and SVM panels of Figure 5.4).

On average across all datasets, POS achieves an improvement over the best compared techniques at all set sizes for RF classifier by between 0.055 and 0.720, measured by the log ratio of the error rates.

The highest improvement in RF classification performance measured by log ratio, 0.720, is obtained at gene sets of size 20. For smaller sizes, the performance ratio decreases, but the POS approach still provides the best accuracy (see Figure 5.4). For *k*NN and SVM classifiers, the averages of improvements across 'Leukaemia', 'Breast', 'Srbct', 'Lung', 'GSE24514', 'GSE4045', 'GSE14333' and 'GSE27854' have been depicted at different set sizes up to 50 genes.

The proposed approach achieves improvements for *k*NN classifier at set sizes not more than 20 features. The highest improvement measured by log ratio, 0.150, is obtained at the selected sets composed of a single gene.

For SVM classifier, improvements over the best method of the compared techniques are achieved by the POS method at most set sizes. The highest improvement measured by the log ratio of the error rates, 0.213, is observed at gene sets of size seven, see the right bottom panel of Figure 5.4.

The best performing technique among the compared methods is not always the same for neither all selected gene set sizes, all datasets nor all classifiers. Hence, the POS algorithm keeps its better performance for large as well as small sets of selected genes with Random Forest and Support Vector Machine classifiers on individual datasets. While it keeps its best performance with *k* Nearest Neighbor classifier for only feature sets with small sizes (specifically, not more than 20). Consequently, the POS feature selection approach is more able to adapt to different pattern of data and to different classifiers than the other

techniques, whose performance is more affected by varying the data characteristics and the used classifier.

5.2.2 Minimum Misclassification Error

A method which is more able to minimize the dependency within its selected candidates can reach a particular level of accuracy using a smaller set of genes. To highlight the entire performances of the compared methods against the POS, a comparison between the minimum error rates achieved by each method is performed. Each method obtains its particular minimum at different size of selected gene set. Tables 5.4-5.6 summarize these results for RF, *k*NN and SVM classifiers respectively. Each row shows the minimum error rate (along-with its corresponding size, shown in brackets) obtained by all methods for a specific dataset, reported in the first column. Since the ISIS method may result in selecting sets with different sizes for each fold of the cross validation, the estimated error rate has been reported along-with the average size of the selected feature sets, shown in brackets. In addition, the error rates of the corresponding classifier with the full set of features, without feature selection, are reported in the last column of Tables 5.4 - 5.6.

5.2.3 Stability Evaluation

An effective feature selection technique is expected to produce stable outcomes across several sub-samples of the considered dataset. This property is particularly desirable for biomarker selections within a diagnostic setting. A stable feature selection method should yield a set of biological informative markers that are selected quite often, while it should rarely or never select randomly chosen features.

Table 5.4: The minimum error rates yielded by Random Forest classifier with feature selection methods along-with the classification error without selection

Dataset	ISIS	Wil-RS	mRMR	MP	POS	Full set
Leukaemia	0.003 (1)	0.030 (20)	0.118 (40)	0.015 (9)	**0.0002** (40)	0.049
Breast	0.407 (4)	0.371 (50)	0.407 (48)	0.354 (48)	**0.308** (45)	0.369
Srbct	0.092 (2.63)	0.069 (24)	0.074 (46)	0.009 (32)	**0.003** (48)	0.0008
Prostate	0.097 (4.18)	0.200 (50)	0.140 (50)	0.069 (50)	**0.062** (50)	0.088
All	0.0004 (1.018)	0.143 (40)	0.011 (50)	**0** (40)	**0** (20)	0
Lung	0.022 (3.26)	0.040 (30)	0.016 (48)	0.008 (46)	**0.007** (48)	0.003
Carcinoma	0.171 (1.29)	**0.003** (41)	0.017 (44)	0.019 (5)	0.026 (20)	0.027
GSE24514	0.107 (1.96)	0.054 (47)	0.063 (50)	0.036 (48)	**0.032** (24)	0.041
GSE4045	0.27 (1.47)	0.134 (24)	0.187 (37)	0.137 (21)	**0.114** (27)	0.205
GSE14333	0.423 (9)	**0.421** (10)	-	0.438 (31)	0.437 (34)	0.414
GSE27854	0.448 (5)	**0.401** (15)	-	0.444 (49)	0.451 (6)	0.488

The numbers in brackets represent the size, average size for ISIS method, of the gene sets that corresponding to the minimum error rate.

Table 5.5: The minimum error rates yielded by k Nearest Neighbor classifier with feature selection methods along-with the classification error without selection

Dataset	ISIS	Wil-RS	mRMR	MP	POS	Full set
Leukaemia	0.064 (1)	0.074 (6)	0.135 (50)	0.019 (1)	**0.005** (1)	0.109
Breast	0.385 (4)	0.405 (11)	0.404 (50)	0.346 (19)	**0.332** (11)	0.405
Srbct	0.105 (2.63)	0.157 (3)	0.098 (48)	**0.005** (26)	**0.005** (22)	0.034
Lung	0.030 (3.26)	0.203 (12)	0.027 (49)	0.017 (17)	**0.011** (12)	0.0005
GSE24514	0.074 (1.96)	**0.032** (20)	0.041 (50)	0.036 (50)	0.039 (50)	0.041
GSE4045	0.239 (1.47)	**0.066** (43)	0.207 (38)	0.137 (50)	0.142 (3)	0.103
GSE14333	0.425 (9)	**0.420** (8)	-	0.455 (23)	0.450 (34)	0.438
GSE27854	0.432 (5)	**0.420** (3)	-	0.454 (13)	**0.420** (6)	0.464

The numbers in brackets represent the size, average size for ISIS method, of the gene sets that corresponding to the minimum error rate.

The stability index proposed by Lausser et al. (2013) is used to measure the stability of the compared method at different set sizes of features. Values of this stability score range from $1/\lambda$, where λ is the total number of used sub-samples for the worst unstable selections to 1 for a fully stable selection. In our context, $\lambda = 500$ since fifty repetitions are used with 10-fold cross validation for the analysis, see section 5.2.

Table 5.7 and Figures 5.5 and 5.6 show the stability scores of different feature selection

Table 5.6: The minimum error rates yielded by Support Vector Machine classifier with feature selection methods along-with the classification error without selection

Dataset	ISIS	Wil-RS	mRMR	MP	POS	Full set
Leukaemia	0.018 (1)	0.047 (8)	0.126 (50)	0.022 (1)	**0.005** (1)	0.131
Breast	0.409 (4)	0.401 (39)	0.407 (50)	0.359 (21)	**0.313** (22)	0.438
Srbct	0.106 (2.63)	0.131 (50)	0.124 (49)	0.010 (21)	**0.003** (8)	0.079
Lung	0.013 (3.26)	0.066 (50)	0.026 (50)	0.021 (19)	**0.010** (47)	0.024
GSE24514	0.090 (1.96)	0.041 (40)	0.059 (50)	0.037 (40)	**0.034** (30)	0.070
GSE4045	0.236 (1.47)	0.134 (24)	0.187 (37)	**0.095** (47)	0.114 (29)	0.214
GSE14333	0.416 (9)	0.427 (9)	-	**0.412** (1)	0.431 (1)	0.407
GSE27854	0.434 (5)	**0.431** (25)	-	0.465 (13)	0.456 (8)	0.50

The numbers in brackets represent the size, average size for ISIS method, of the gene sets that corresponding to the minimum error rate.

methods for the 'Srbct', 'GSE27854' and 'GSE24514' datasets respectively. Figure 5.5 shows that the POS approach provides more stable feature selections than Wil-RS and MP methods at most set sizes selected from 'GSE27854' dataset. For GSE24514 dataset, Figure 5.6 depicts the stability scores of compared feature selection techniques at different set sizes. Unlike the mRMR and the MP approaches, both the Wil-RS and the POS methods keep their stability degree for different sizes of feature sets. The POS method provides a stability degree close to the well established Wil-RS method. For the 'Srbct' data, the best stability scores among the compared methods are yielded by POS at most set sizes, see Table 5.7.

Table 5.7: Stability scores of the feature selection techniques over 50 repetitions of 10-fold cross validation for 'Srbct' dataset

N. selected genes	Wil-RS	mRMR	MP	POS
5	0.789	0.097	**0.815**	0.760
10	0.804	0.198	0.788	**0.844**
15	0.804	0.302	0.853	**0.911**
20	0.857	0.405	0.898	**0.908**
25	**0.883**	0.506	0.871	0.872
30	**0.896**	0.579	0.871	0.870
35	**0.868**	0.640	0.852	0.859
40	**0.858**	0.705	0.833	0.847
45	**0.862**	0.754	0.812	0.835
50	**0.873**	0.803	0.800	0.820

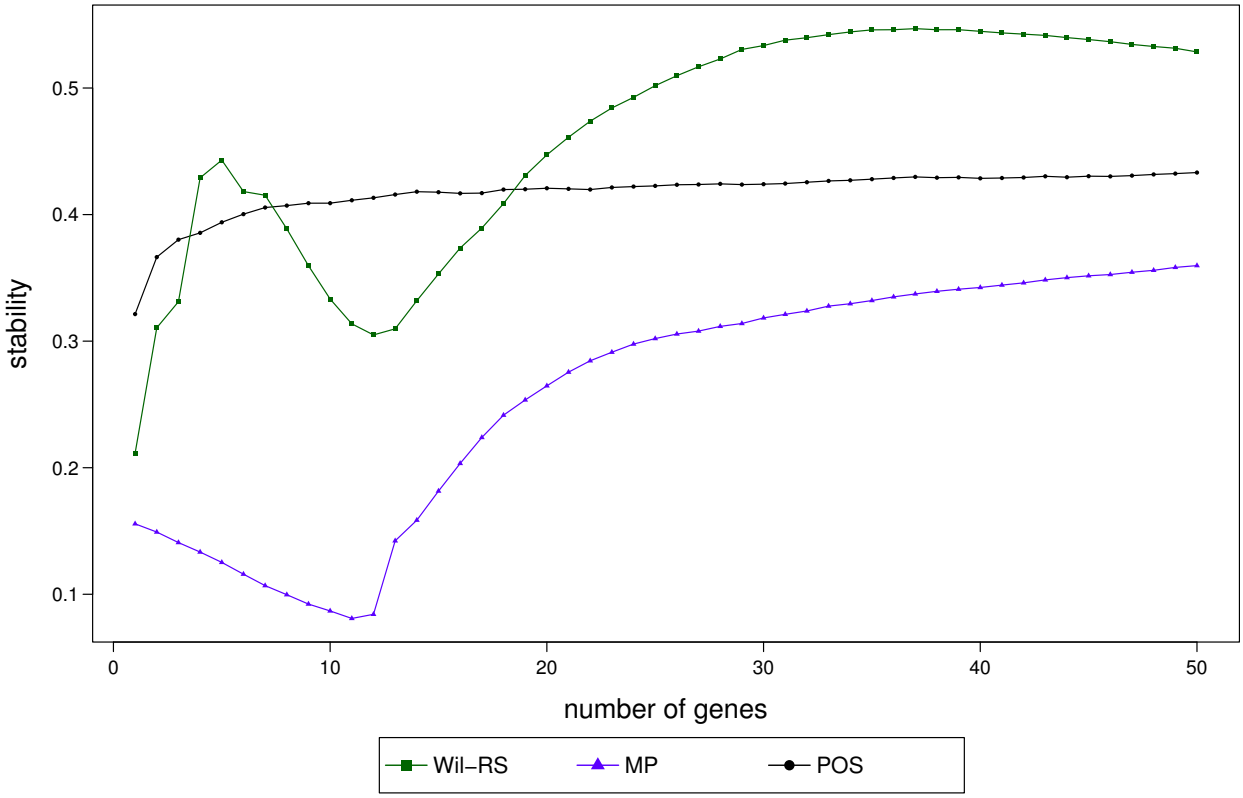

Figure 5.5: *Stability scores at different sizes of features sets that selected by Wil-RS, MP and POS methods on 'GSE27854' dataset.*

A stable selection does not guarantee the relevancy of the selected features to the considered response of the target class labels. The prediction accuracy yielded by a classifier based on the selected features should also be highlighted in conjunction with stability. The

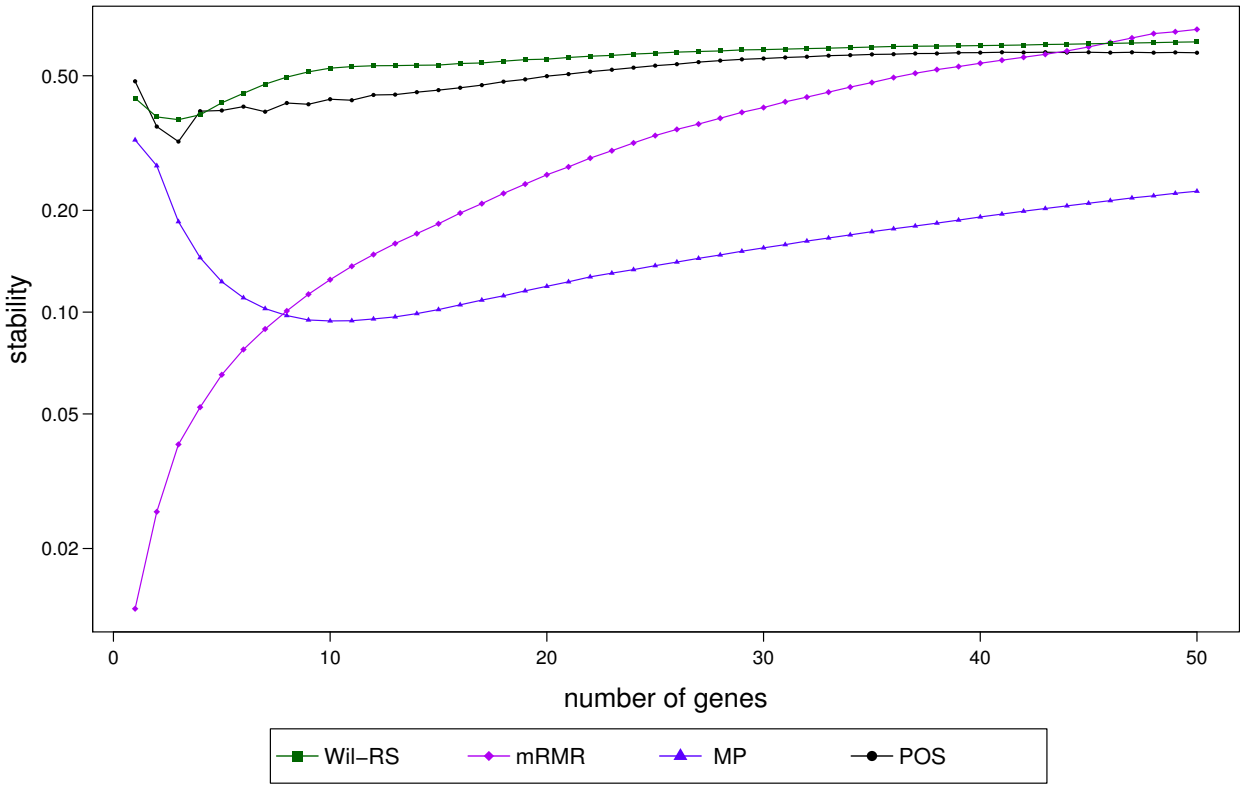

Figure 5.6: *Stability scores at different sizes of features sets that selected by Wil-RS, mRMR, MP and POS methods on 'GSE24514' dataset.*

relation between the accuracy and stability has been outlined by Figures 5.7 and 5.8 for the 'Lung' and 'GSE27854' datasets respectively. The stability scores were combined with corresponding error rates yielded by the three different classifiers: RF; *k*NN; SVM. Different dots for the same feature selection method correspond to different set sizes of features. With the greatest stability and lowest error rate, the best method is the one whose dots are depicted in the upper-left corner of the plot. For all classifiers, POS method achieves a good trade-off between accuracy and stability for 'Lung' data, see Figure 5.7. For the 'GSE27854' data with the *k*NN classifier, POS provides a better trade-off between accuracy and stability than other compared methods. Whereas with the RF and SVM classifiers, POS is outperformed by Wil-RS.

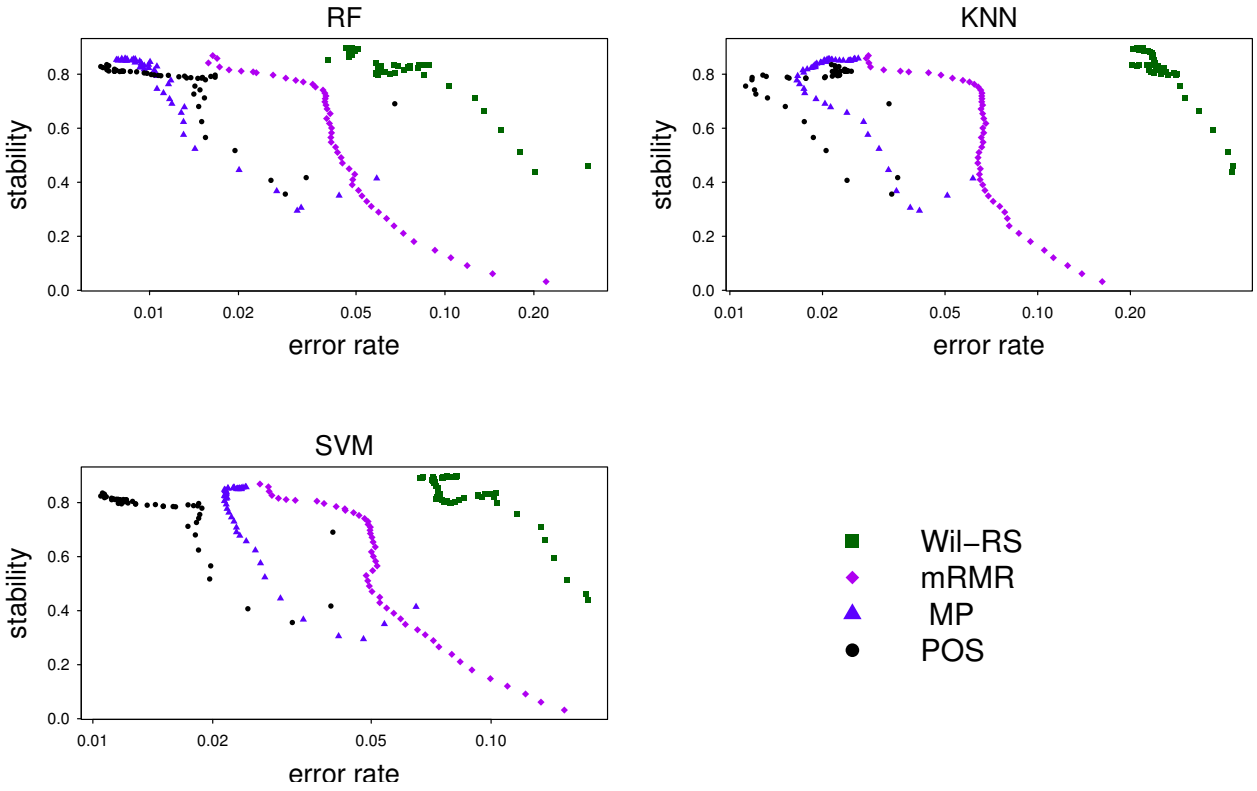

Figure 5.7: *The stability of the feature selection methods against the corresponding estimated error rates on 'Lung' dataset. The error rates have been measured by 50 repetations of 10-fold cross validation for three different classifiers: Random Forest (RF); k Nearest Neighbor (kNN); Support Vector Machine (SVM).*

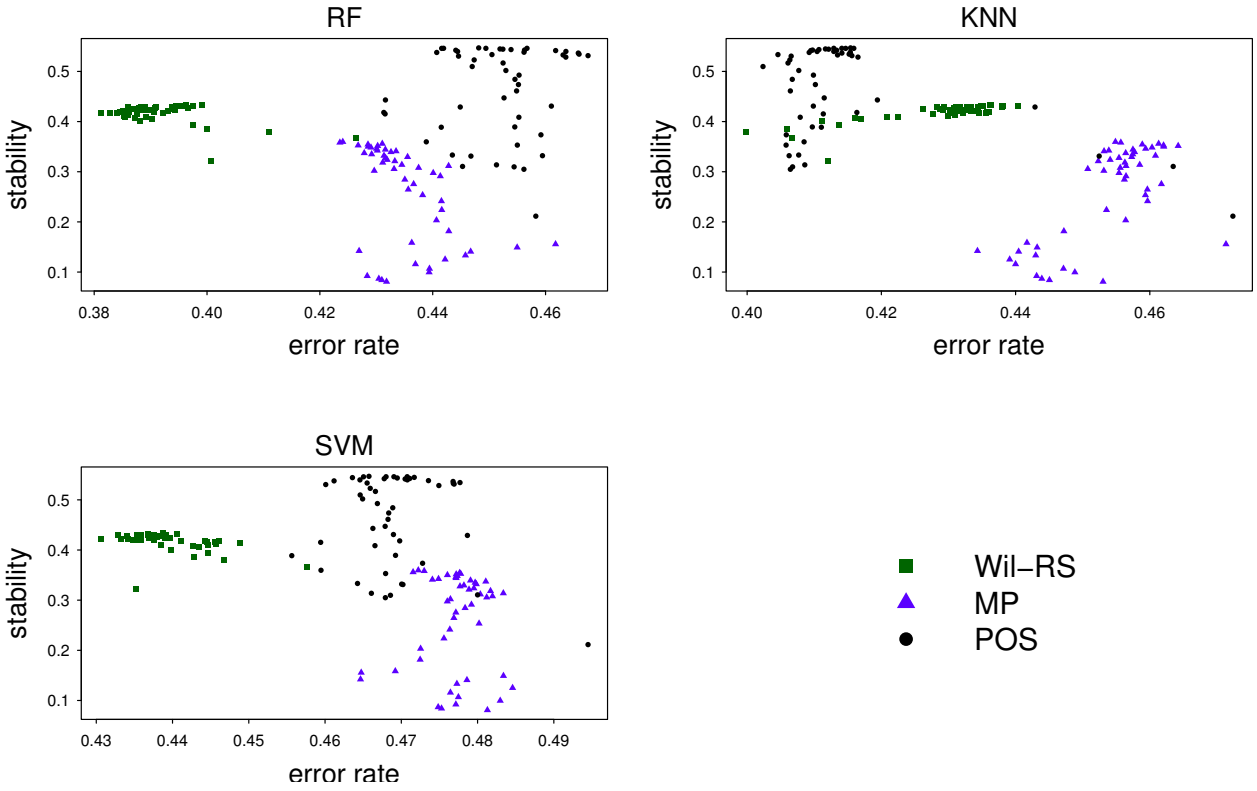

Figure 5.8: *The stability of the feature selection methods against the corresponding estimated error rates on 'GSE27854' dataset. The error rates have been measured by 50 repetations of 10-fold cross validation for three different classifiers: Random Forest (RF); k Nearest Neighbor (kNN); Support Vector Machine (SVM).*

5.3 Summary

A gene selection method, named POS, is proposed. A gene ranking process is performed based on two measures, *POS* and *RDC*, for estimating the overlapping degree across different classes and assigning each gene to its relative dominant class respectively. The detected minimum subset of genes is then extended by adding the top ranked genes to produce a final gene selection.

The POS is applied on eleven publicly available gene expression datasets with different characteristics. Feature sets of different sizes, up to 50 genes, are selected using widely used gene selection methods: Wilcoxon Rank Sum (Wil-RS); Minimum redundancy maximum relevance (mRMR); MaskedPainter (MP); Iteratively sure independence screening (ISIS) along-with my proposal, POS. Then, the prediction models of three different classifiers: Random Forest; k Nearest Neighbor; Support Vector Machine are constructed with the selected features. The estimated classification error rates obtained by the considered classifiers show that POS provides better performance.

The stability of the selections yielded by the compared feature selection methods using the cross validation technique has been highlighted. Stability scores computed at different set sizes of the selected features show that the proposed method has a stable performance for different sizes of selected features. The analysed relationship between classification accuracies yielded by three different classifiers and stability confirms that the POS method can provide a good trade-off between stability and classification accuracy.

All procedures proposed in this chapter have been programmed into an R package named 'propOverlap' (Mahmoud et al. 2014b). It is publicly available for download from

the Comprehensive R Archive Network (CRAN) repository. Its reference manual is reported in Appendix C.

The next chapter extends the POS method to minimize selection redundancy. The extended version, named POSr, detects the minimum gene subset in a recursive way to mitigate redundancy problem in the final gene selection.

Chapter 6

Minimizing Redundancy among Selected Genes

The POS method, presented in Chapter 5, ranks genes based on their discriminative power towards a considered response (e.g., particular target disease). Its final selection is provided by combining the minimum gene subset, obtained by the proposed procedure in Algorithm 4.1, with the top ranked genes according to the estimated *POS* scores together with the assignments of *RDC* measure (see (4.10) and (5.1)).

Feature selections produced by POS are robust against outliers, since gene masks are defined based on the interquartile range of gene's expressions. Results of the conducted experiment, shown in Section 5.2, demonstrate that POS technique provides a good classification performance as well as selections stability.

The identification of the top ranked genes, based on the *POS* relevance score, treats each gene separately from other genes. Such a procedure has linear time complexity with respect to P, the total number of genes, but it may provide a classifier with redundant

information, since two highly ranked genes may duplicate each other, with respect to their classification role, though both are selected within the set of top ranked genes.

Selecting redundant features leads to increasing the complexity of a model without providing further information for the considered classification problem. Minimizing redundancy can enhance the accuracy as well as the interpretability of classifier's results. Moreover, selection stability might be improved by avoiding redundant features within established various sub-samples.

In this chapter, an extended version of POS method, called POSr, is proposed by detecting the minimum gene subset, as defined in Algorithm 4.1, in a recursive way to mitigate redundancy problem in the final gene selection.

6.1 Recursive Minimum Sets for Minimizing Selection Redundancy (POSr)

The POSr can be described as follows.

6.1.1 The Method

The gene mask, defined in Section 4.3, is a measure for classification power of a gene. It reflects the capability of the gene to correctly classify each observation to its target class. Genes with higher number of 1 bits in their masks are more informative for the considered classification problem, see (4.9). When two genes classify in the same way the same observations, then their masks should be identical. Genes with complementary masks, on the other hand, can provide diverse information to the classifier model.

In this chapter, an extended version of POS, called POSr, is proposed. It exploits gene masks along-with POS measure, defined in (4.10), to identify minimum subsets of genes in a recursive way in order to mitigate the potential redundancy in the final gene selection. A minimum subset is identified according to Algorithm 4.1, i.e. it is designed to be the minimum one that correctly classify the maximum number of observations in a given training set, avoiding the effects of expression outliers.

Let $\mathbb{G}_z$ be a set containing the remaining genes at the zth iteration, after removing genes selected at the $(z-1)$th iteration, such that $\mathbb{G}_1$ is the full set of all genes (i.e., $|\mathbb{G}_1| = P$). Also, let $M_{..}(\mathbb{G}_z)$ be its aggregate mask which is defined as shown in (4.12). It can be expressed as follows:

$$M_{..}(\mathbb{G}_z) = \bigvee_{i \in \mathbb{G}_z} M_{i.} \tag{6.1}$$

At iteration z, the objective is to search among the set, $\mathbb{G}_z$, for the minimum subset, denoted by $\mathbb{G}_z^*$, using the procedures reported in Algorithm 4.1. In a similar way to (4.13), the subset $\mathbb{G}_z^*$ is defined as the minimum set whose aggregate mask, $M_{..}(\mathbb{G}_z^*)$, equals to the aggregate mask of the corresponding set of genes, $M_{..}(\mathbb{G}_z)$. In other words, the minimum subset of genes should satisfy the following statement:

$$\underset{\mathbb{G}_z^* \subseteq \mathbb{G}_z}{argmin} \left(\left| \mathbb{G}_z^* \right| \middle| \left(M_{..}(\mathbb{G}_z^*) = \bigvee_{i \in \mathbb{G}_z^*} M_{i.} = M_{..}(\mathbb{G}_z) \right) \right). \tag{6.2}$$

This procedure is performed in a recursive way and ends when the required number of genes, set by the user, are selected.

The pseudo code of our procedure, POSr, is reported in Algorithm 6.1. Its inputs are: the matrix of gene masks, M; POS scores for all genes; number of genes to be selected, r. It

produces the sequence of selected genes, $\mathbb{T}^*$, as output.

Algorithm 6.1 *POSr Method: Recursive Minimum Subsets*

Inputs: M, POS scores and number of required genes (r).
Output: Sequence of the selected genes $\mathbb{T}^*$.

1: $z = 0$ {Initialization}
2: $\mathbb{T} = \emptyset$
3: **while** $\left|\mathbb{T}\right| < r$ **do**
4: $z = z + 1$
5: $k = 0$ {Initialization of individual selection}
6: $\mathbb{G}_z^* = \emptyset$
7: $M_{..}(\mathbb{G}_z^*) = \mathbf{0}_N$
8: **while** $M_{..}(\mathbb{G}_z^*) \neq M_{..}(\mathbb{G}_z)$ **do**
9: $k = k + 1$
10: $\mathbb{S}_{zk} = \underset{i \in \mathbb{G}_z}{argmax}\left(\sum_{j=1}^{N} I\left(m_{ij}^{(k)} = 1\right)\right)$ {Assign gene set whose masks have max. bits of 1}
11: $g_{zk} = \underset{i \in \mathbb{S}_{zk}}{argmin}\,(POS_i)$ {Select the candidate with the best score among the assigned set}
12: $\mathbb{G}_z^* = \mathbb{G}_z^* + g_{zk}$ {Update the target set by adding the selected candidate}
13: **for all** $i \in \mathbb{G}_z$ **do**
14: $M_{i.}^{(k+1)} = M_{i.}^{(k)} \wedge M'_{..}(\mathbb{G}_z^*)$ {update gene masks such that the uncovered observations are only considered}
15: **end for**
16: **end while**
17: $\mathbb{T} = \mathbb{T} + \mathbb{G}_z^*$
18: $\mathbb{G}_{z+1} = \mathbb{G}_z - \mathbb{G}_z^*$
19: **end while**
20: $\mathbb{T}^*$ is the sequence whose members are the first r genes in $\mathbb{T}$
21: **return** $\mathbb{T}^*$

At the initial step ($z = 0$), we let $\mathbb{T} = \emptyset$ (line 2); where $\mathbb{T}$ is a set created to contain the successively selected minimum subsets of genes. Then at each iteration, z, the following steps are performed:

1. we let $k = 0$, $\mathbb{G}_z^* = \emptyset$ and $M_{..}(\mathbb{G}_z^*) = \mathbf{0}_N$ (lines 5-7) to initialize individual selection within the minimum subset $\mathbb{G}_z^*$, where $M_{..}(\mathbb{G}_z^*)$ is the aggregate mask of the set $\mathbb{G}_z^*$,

see (6.1). Then at each sub-iteration, k, the procedure presented in Algorithm 4.1 is performed via the following sub-steps:

(a) Among genes of the set $\mathbb{G}_z$, the one(s) with the highest number of mask bits assigned to 1 is (are) chosen to form the set $\mathbb{S}_{zk}$ (line 10). This set will not be empty as long as the loop condition is still satisfied, i.e. $M_{..}(\mathbb{G}_z^*) \neq M_{..}(\mathbb{G}_z)$. Under this condition, our selected genes don't cover yet the maximum number of observations that should be covered by our target gene subset of the zth iteration, $\mathbb{G}_z^*$. Note that our definition for gene masks allows $M_{..}(\mathbb{G}_z)$ to report in advance which observations should be covered by the minimum subset of genes. Therefore, there will be at least one gene mask which has at least one bit assigned to 1 if that condition is to hold.

(b) The gene with the lowest *POS* score among genes in $\mathbb{S}_{zk}$, if there are more than one, is selected (line 11). It is denoted by g_{zk}.

(c) The set $\mathbb{G}_z^*$ is updated by adding the selected gene, g_{zk} (line 12).

(d) All masks of genes in $\mathbb{G}_z$ are also updated by performing the logical conjunction (*logic AND*) with the negated aggregate mask of set $\mathbb{G}_z^*$ (line 14). The negated mask $M'_{..}(\mathbb{G}_z^*)$ of the mask $M_{..}(\mathbb{G}_z^*)$ is the one obtained by applying logical negation (*logical complement*) on this mask. Consequently, the bits of ones corresponding to the classification of still uncovered observations are only considered. Note that $M_{i.}^{(k)}$ represents updated mask of gene i at the kth iteration such that $M_{i.}^{(1)}$ is its original gene mask whose elements are computed according to (4.9).

2. The sub-steps 1(a)-1(d) for detecting the target of minimum gene subset, $\mathbb{G}_z^*$, are

successively iterated and end when all masks of genes in $\mathbb{G}_z$ have no 1 bits anymore, i.e. the selected genes cover the maximum number of observations. This situation is accomplished iff $M_{..}(\mathbb{G}_z^*) = M_{..}(\mathbb{G}_z)$, (lines 8-16).

3. The set $\mathbb{T}$ is updated by adding the detected minimum subset of genes, $\mathbb{G}_z^*$ (line 17).

4. Genes within the selected minimum subset, $\mathbb{G}_z^*$, are then removed from the set of genes, $\mathbb{G}_z$ (line 18).

5. The procedure is successively iterated and ends when the size of the set $\mathbb{T}$ is greater than or equal the number of required genes, r. Then, the target sequence of selected genes, $\mathbb{T}^*$, is produced by selecting the first r genes in $\mathbb{T}$ (lines 20, 21).

POSr approach combines recursively the detected minimum subsets of genes that provide the best classification coverage for a given training set. Selection of the minimum subsets based on the updated gene masks allows POSr to minimize redundancy among the final selection list. The method of POSr can be described briefly as follows:

- POSr utilizes the defined gene masks to robustly detect the minimum subset of genes that maximizes the correct assignment of training observations to their corresponding classes i.e., the minimum subset that can yield the best classification accuracy on a training set avoiding the effects of outliers.

- Genes involved in the detected minimum subset are removed from the full set of genes. The reduced gene set is then used to detect a new minimum subset of genes. This procedure is performed recursively.

- The final rank is produced by combining the selected minimum subsets.

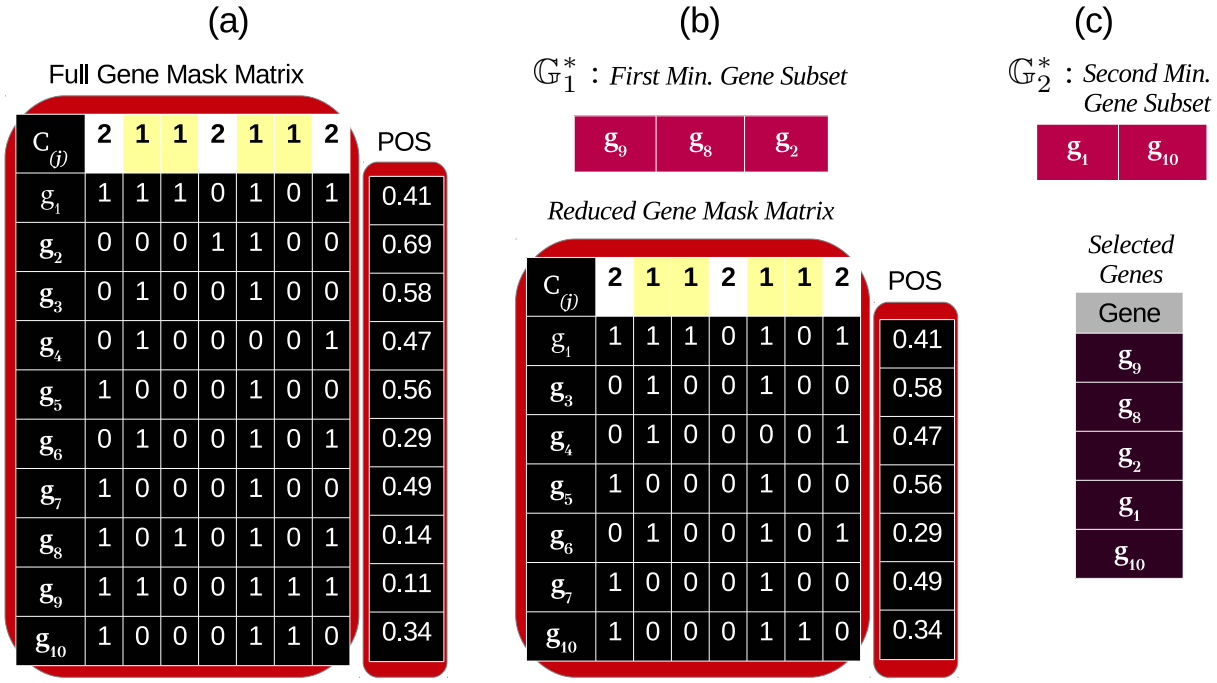

Figure 6.1: *An example of the POSr approach: (a) genes with their masks and proportional overlapping scores; (b) the minimum gene subset, $\mathbb{G}_1^*$, obtained from iteration $z = 1$ in Algorithm 6.1, and the reduced gene mask matrix; (c) the minimum gene subset, $\mathbb{G}_2^*$, and resulted genes at the end of the process.*

6.1.2 Illustrative Example

To illustrate the POSr approach, reported in Algorithm 6.1, consider the set of genes represented in Figure 6.1(a). Each gene is associated with its constructed gene mask and its proportional overlapping score (*POS*). For instance, gene g_1 has a mask of [1110101], i.e. it classifies unambiguously all training observations except the forth and the sixth observations, and a proportional overlapping score equals 0.41.

At the first iteration, $z = 1$, the mask matrix of the full gene set, $\mathbb{G}_1$, is used together with the corresponding *POS* scores to identify the first minimum gene subset, $\mathbb{G}_1^*$, using the procedure described in Algorithm 6.1: lines 5-16. The resulting subset includes g_9, g_8 and g_2.

If five genes are required by the user for biological investigation regarding the two

clinical classes 1 and 2 (hence, $r = 5$), then further iterations are required, since the size of the selected subset, $\left|\mathbb{G}_1^*\right| = 3$, is less than r. The selected genes are then removed and the reduced gene set, $\mathbb{G}_2$, is considered for the iteration $z = 2$, see Figure 6.1(b).

Again, the minimum gene subset, $\mathbb{G}_2^*$, is identified. According to Algorithm 6.1, its members are g_1 and g_{10}. Then the sequence of genes resulting from combining the selected minimum subsets, $\mathbb{G}_1^*$ and $\mathbb{G}_2^*$, is produced as the final selection. It is shown in Figure 6.1(c).

6.2 Results

POSr method is validated by comparison with the well-known gene selection techniques: the Wilcoxon Rank Sum (Wil-RS) technique; Minimum Redundancy Maximum Relevance (mRMR) method; MaskedPainter (MP) approach; Iteratively Sure Independent Screening (ISIS) along-with POS method. The performance is evaluated by obtaining the classification error rates from three different classifiers: Random Forest (RF); k Nearest Neighbor (kNN); Support Vector Machine (SVM).

Fifty repetitions of 10-fold cross validation analysis were performed for each combination of considered dataset described in Table 5.1, feature selection algorithm, and a given number of selected genes, up to 50, with the considered classifiers.

Tables 6.1 and 6.2 show the average classification error rates obtained by Wil-RS, mRMR, MP, POS and POSr with RF, *k*NN and SVM classifiers on Leukaemia and GSE4045 datasets respectively. Each row provides the average classification error rate at a specific number of selected genes (reported in the first column). The aggregate average error value and the minimum error rate for each method with each classifier are provided in the last two

rows. Average error rates yielded on the Breast and Srbct datasets using RF, *k*NN, and SVM classifiers are shown in Figure 6.2.

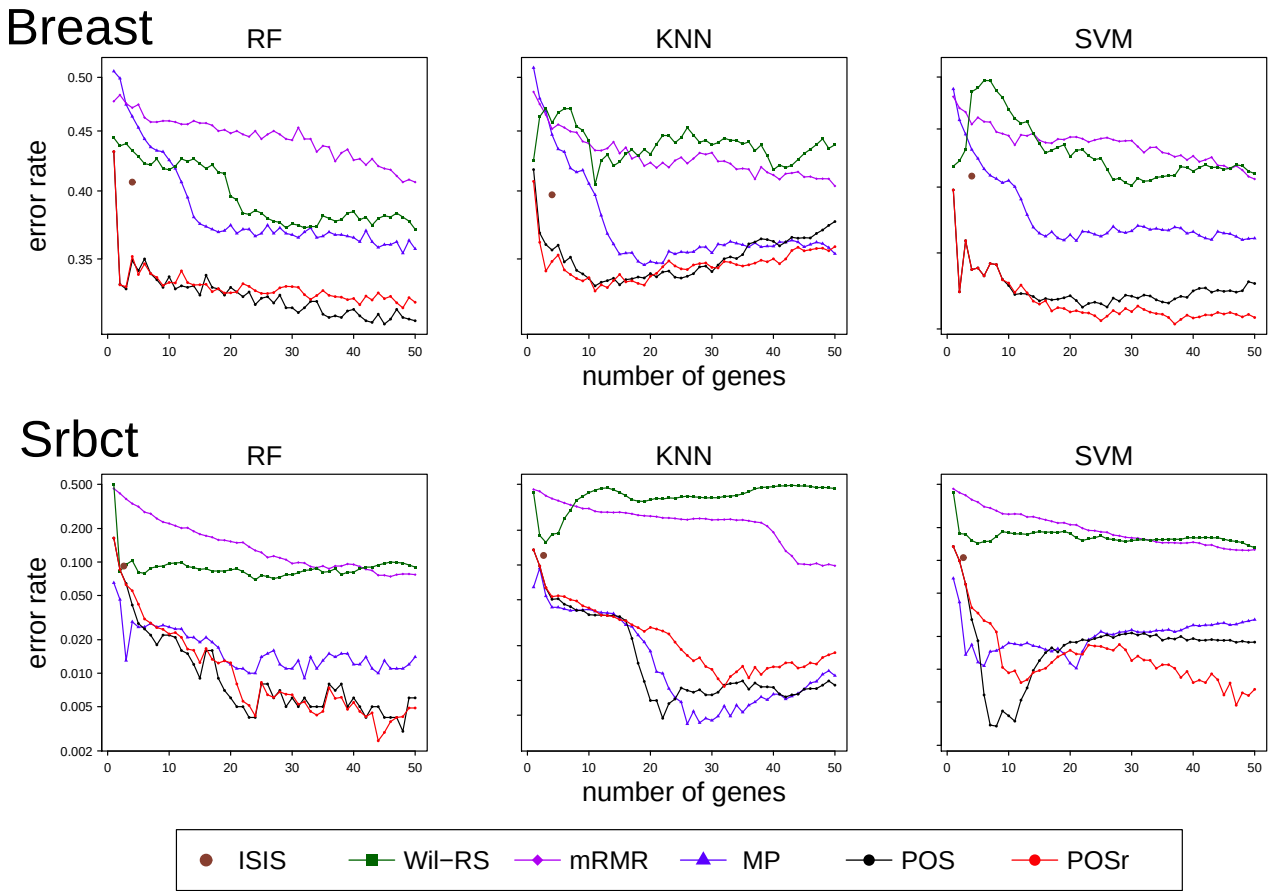

Figure 6.2: *Average classification error rates for 'Srbct' and 'Breast' data based on 50 repetitions 10-fold CV using ISIS, Wil-RS, mRMR, MP, POS and POSr methods.*

The POSr approach yields a good performance with different classifiers on the considered datasets. For the Random Forest classifier, in particular on the Leukaemia, GSE4045, Srbct, Lung and GSE24514 datasets, the classification average error rates on the test sets are less than all other feature selection techniques, including the POS method, at most genes set sizes. While, on the Breast dataset, POSr shows a comparable performance with the best technique, POS. On the GSE14333 dataset, Wil-RS technique has outperformed all methods. A detailed comparison of the minimum classification error rates obtained by

Table 6.1: Average classification error rates yielded by Random Forest, k Nearest Neighbors and Support Vector Machine classifiers on 'Leukaemia' dataset using 50 repetitions of 10-fold cross validation approach

N.genes	RF					*k*NN					SVM				
	Wil-RS	mRMR	MP	POS	POSr	Wil-RS	mRMR	MP	POS	POSr	Wil-RS	mRMR	MP	POS	POSr
1	0.126	0.211	0.015	**0.003**	**0.003**	0.141	0.220	0.019	0.005	**0.003**	0.133	0.238	0.022	**0.005**	**0.005**
2	0.083	0.197	0.017	**0.001**	0.006	0.110	0.195	0.059	0.047	**0.017**	0.099	0.197	0.053	**0.026**	0.039
3	0.068	0.185	0.020	**0.003**	0.007	0.086	0.198	0.070	0.073	**0.031**	0.078	0.198	0.064	0.044	**0.032**
4	0.044	0.180	0.016	**0.001**	**0.001**	0.082	0.194	0.076	0.069	**0.039**	0.068	0.178	0.070	0.050	**0.030**
5	0.043	0.168	0.015	**0.002**	0.003	0.077	0.191	0.084	0.075	**0.035**	0.060	0.172	0.079	0.060	**0.026**
6	0.037	0.170	0.018	0.005	**0.001**	0.074	0.188	0.087	0.065	**0.031**	0.052	0.171	0.082	0.065	**0.028**
7	0.036	0.161	0.018	0.004	**0.001**	0.077	0.182	0.090	0.065	**0.026**	0.049	0.162	0.086	0.069	**0.023**
8	0.035	0.158	0.020	0.004	**0.000**	0.081	0.186	0.092	0.063	**0.021**	0.047	0.166	0.090	0.074	**0.014**
9	0.032	0.161	0.015	0.003	**0.000**	0.082	0.176	0.090	0.067	**0.023**	0.049	0.162	0.092	0.083	**0.016**
10	0.031	0.157	0.018	0.003	**0.000**	0.078	0.181	0.094	0.067	**0.026**	0.050	0.159	0.092	0.079	**0.018**
20	0.030	0.141	0.028	0.001	**0.000**	0.085	0.162	0.102	0.064	**0.031**	0.062	0.145	0.088	0.068	**0.026**
30	0.030	0.131	0.029	0.001	**0.000**	0.085	0.155	0.108	0.070	**0.038**	0.058	0.139	0.093	0.066	**0.024**
40	0.031	0.118	0.031	**0.000**	**0.000**	0.084	0.142	0.105	0.078	**0.040**	0.053	0.127	0.094	0.069	**0.025**
50	0.031	0.119	0.029	0.001	**0.000**	0.083	0.135	0.107	0.078	**0.041**	0.049	0.126	0.101	0.062	**0.022**
Avg.	0.041	0.157	0.021	0.002	**0.001**	0.087	0.179	0.085	0.063	**0.028**	0.065	0.167	0.079	0.059	**0.023**
Min.	0.030	0.118	0.015	**0.000**	**0.000**	0.074	0.135	0.019	0.005	**0.003**	0.047	0.126	0.022	**0.005**	**0.005**

Boldface numbers indicate the minimum average of classification error rates (the highest accuracy) achieved with the corresponding classifier at each size of selected gene sets, reported in the first column.

Table 6.2: Average classification error rates yielded by Random Forest, k Nearest Neighbors and Support Vector Machine classifiers on 'GSE4045' dataset using 50 repetitions of 10-fold cross validation approach

N.genes	RF					*k*NN					SVM				
	Wil-RS	mRMR	MP	POS	POSr	Wil-RS	mRMR	MP	POS	POSr	Wil-RS	mRMR	MP	POS	POSr
1	**0.201**	0.330	0.245	0.248	0.248	0.280	0.235	0.227	**0.213**	**0.213**	**0.201**	0.330	0.221	0.249	0.249
2	0.201	0.266	0.208	**0.186**	**0.186**	0.232	0.228	0.172	0.165	**0.164**	0.201	0.266	0.186	0.186	**0.178**
3	0.195	0.245	0.180	**0.152**	**0.152**	0.225	0.231	0.153	**0.142**	**0.142**	0.195	0.245	0.166	**0.152**	**0.152**
4	0.193	0.236	0.194	0.156	**0.152**	0.224	0.227	0.144	0.166	**0.132**	0.193	0.236	0.153	0.156	**0.137**
5	0.178	0.223	0.172	**0.126**	0.142	0.215	0.225	0.148	0.160	**0.122**	0.178	0.223	0.149	0.126	**0.125**
6	0.182	0.228	0.169	**0.129**	0.132	0.215	0.231	0.153	0.172	**0.112**	0.182	0.228	0.154	0.129	**0.117**
7	0.177	0.218	0.160	0.130	**0.127**	0.214	0.229	0.147	0.171	**0.112**	0.177	0.218	0.143	0.130	**0.117**
8	0.176	0.213	0.155	0.136	**0.121**	0.211	0.231	0.149	0.171	**0.110**	0.176	0.213	0.143	0.136	**0.116**
9	0.172	0.217	0.151	0.132	**0.127**	0.206	0.229	0.148	0.166	**0.120**	0.172	0.217	0.139	0.132	**0.108**
10	0.172	0.211	0.147	0.132	**0.122**	0.210	0.221	0.150	0.165	**0.127**	0.172	0.211	0.134	0.132	**0.102**
20	0.143	0.193	0.141	0.125	**0.113**	0.183	0.211	0.147	0.157	**0.135**	0.143	0.193	0.114	0.125	**0.094**
30	0.140	0.197	0.149	0.122	**0.117**	0.170	0.221	0.147	0.154	**0.129**	0.140	0.197	0.106	0.122	**0.098**
40	0.141	0.192	0.152	0.121	**0.115**	0.170	0.209	0.144	0.158	**0.132**	0.141	0.192	0.098	0.121	**0.093**
50	0.143	0.192	0.154	0.134	**0.114**	0.178	0.214	0.137	0.166	**0.133**	0.143	0.192	0.098	0.134	**0.087**
Avg.	0.172	0.226	0.170	0.145	**0.141**	0.210	0.224	0.155	0.166	**0.135**	0.172	0.226	0.143	0.145	**0.127**
Min.	0.140	0.192	0.141	0.121	**0.113**	0.170	0.209	0.137	0.142	**0.110**	0.140	0.192	0.098	0.121	**0.087**

Boldface numbers indicate the minimum average of classification error rates (the highest accuracy) achieved with the corresponding classifier at each size of selected gene sets, reported in the first column.

each gene selection method is reported in Table 6.3.

For the *k*NN classifier, POSr provides a good classification performance. On the Leukaemia and GSE4045 datasets, its classification average error rates are less than all other feature selection techniques at all selected genes set sizes, see Tables 6.1 and 6.2. On the Breast and Lung datasets, POSr provides lower error rates than all other methods on most set sizes. On the Srbct dataset, POS technique has outperformed all methods for set sizes that are less than 25, whereas for larger sets, the MP method is the best. Wil-RS method provides lower error rates that other approaches on the GSE14333 and GSE24514 datasets.

For the SVM classifier, POSr provides a good classification performance on all used datasets. In particular on the Leukaemia and GSE4045 datasets, the classification average error rates on the test sets are less than all other feature selection techniques at almost all selected genes set sizes, see Tables 6.1 and 6.2. While, on the Breast dataset, POSr shows a comparable performance with the best technique, POS, for feature set sizes less than 14 genes, whereas for larger sets, POSr outperformed all other compared methods. For the Srbct data set, POSr outperformed all other compared methods for feature set sizes more than 14, whereas it is outperformed by POS method for smaller feature sets. On the Lung dataset, POSr provides the best performance for almost all feature set sizes.

A comparison between the minimum error rates achieved by each method highlights the entire performances of the compared methods against the proposed approach, POSr. Each method obtains its particular minimum at different size of selected gene set. Table 6.3 summarizes these results for RF, *k*NN and SVM classifiers. Each row shows the minimum error rate (associated with its corresponding feature set size, shown in brackets) obtained

by all methods for a specific dataset, reported in the first column, using a specific classifier, reported in the second column. In addition, the error rates of the corresponding classifier with the full set of features, without feature selection, are reported in the last column.

Table 6.3 demonstrates that POSr approach provides the minimum error rates for most of the used datasets. Due to limitations of the R package 'mRMRe' (De Jay et al. 2013), mRMR selections could not be conducted for datasets having more than '46340' features. Therefore, mRMR method is excluded from the analysis of the 'GSE14333' dataset.

Table 6.3: Comparison between the minimum error rates yielded by the feature selection methods using RF, *k*NN and SVM classifiers.

Dataset	*Classifier*	*Wil-RS*	*mRMR*	*MP*	*POS*	*POSr*	*Full Set*
Leukaemia	RF	0.030 (20)	0.118 (40)	0.015 (9)	0.0002 (40)	**0.000 (9)**	0.049
	*k*NN	0.074 (6)	0.135 (50)	0.019 (1)	**0.005 (1)**	**0.005 (1)**	0.109
	SVM	0.047 (8)	0.126 (50)	0.022 (1)	**0.005 (1)**	**0.005 (1)**	0.131
Lung	RF	0.040 (30)	0.016 (48)	0.008 (46)	0.007 (48)	**0.006 (48)**	0.003
	*k*NN	0.203 (12)	0.027 (49)	0.017 (17)	0.011 (12)	**0.002 (40)**	0.0005
	SVM	0.066 (50)	0.026 (50)	0.021 (19)	0.010 (47)	**0.008 (38)**	0.024
Breast	RF	0.371 (50)	0.407 (48)	0.354 (48)	**0.308 (45)**	0.317 (48)	0.369
	*k*NN	0.405 (11)	0.404 (50)	0.346 (19)	0.332 (11)	**0.328 (11)**	0.405
	SVM	0.401 (39)	0.407 (50)	0.359 (21)	0.313 (22)	**0.303 (37)**	0.438
Srbct	RF	0.069 (24)	0.074 (46)	0.009 (32)	0.003 (48)	**0.002 (44)**	0.0008
	*k*NN	0.157 (3)	0.098 (48)	0.005 (26)	**0.005 (22)**	0.008 (32)	0.034
	SVM	0.131 (50)	0.124 (49)	0.010 (21)	**0.003 (8)**	0.004 (47)	0.079
GSE4045	RF	0.134 (24)	0.187 (37)	0.137 (21)	0.114 (27)	**0.105 (33)**	0.205
	*k*NN	0.166 (43)	0.207 (38)	0.137 (50)	0.142 (3)	**0.112 (6)**	0.103
	SVM	0.134 (24)	0.187 (37)	0.095 (47)	0.114 (29)	**0.085 (47)**	0.214
GSE14333	RF	**0.421 (10)**	-	0.438 (31)	0.437 (34)	0.442 (44)	0.414
	*k*NN	**0.420 (8)**	-	0.455 (23)	0.450 (34)	0.448 (47)	0.438
	SVM	0.427 (9)	-	**0.412 (1)**	0.431 (1)	0.431 (1)	0.407
GSE24514	RF	0.054 (47)	0.063 (50)	0.036 (48)	**0.032 (24)**	0.034 (26)	0.041
	*k*NN	**0.032 (20)**	0.041 (50)	0.036 (50)	0.039 (50)	0.038 (49)	0.041
	SVM	0.041 (40)	0.059 (50)	0.037 (40)	**0.034 (30)**	0.036 (43)	0.070

Boldface numbers indicate the lowest classification error rates (highest accuracy among compared methods) achieved using the corresponding classifier. The numbers in brackets represent the size of the gene sets that corresponding to the minimum error rate.

The stability index proposed by Lausser et al. (2013) is used to measure the stability of the compared method at different set sizes of features. A similar evaluation for stability

selections of the considered feature selection approaches is conducted in Section 5.2.3.

Figure 6.3 shows stability scores of the feature selection methods for the GSE24514 dataset at different set sizes of selected features. Unlike the mRMR and the MP approaches, all the Wil-RS, POS and POSr methods keep their stability degree for different sizes of feature sets. The POSr method provides a stability degree comparable with POS and the well established Wil-RS method.

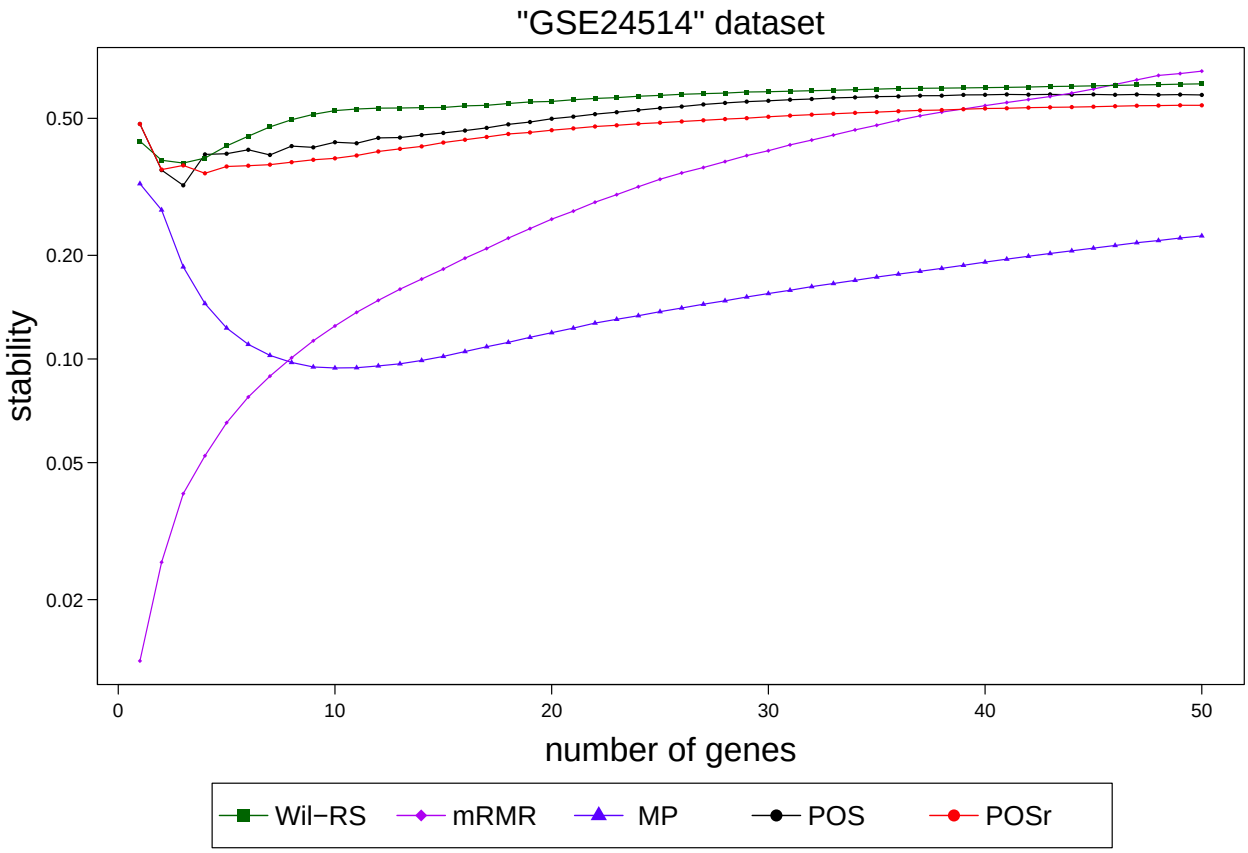

Figure 6.3: *Stability scores at different sizes of features sets that selected by Wil-RS, MP and POS methods on 'GSE24514' dataset.*

The prediction accuracy yielded by RF, *k*NN and SVM classifiers are also highlighted in conjunction with stability. Figure 6.4 outlines the relation between the classification accuracy and selection stability for the 'Lung' dataset. The stability scores were combined

with their corresponding error rates yielded by the three different classifiers: RF; *k*NN; SVM. Different dots for the same feature selection method represent different set sizes of selected features. For all classifiers, POSr method achieves a good trade-off between accuracy and stability. With the *k*NN classifier, POSr provides a better trade-off between accuracy and stability than other compared methods. Whereas with the RF and SVM classifiers, POSr is outperformed by POS method.

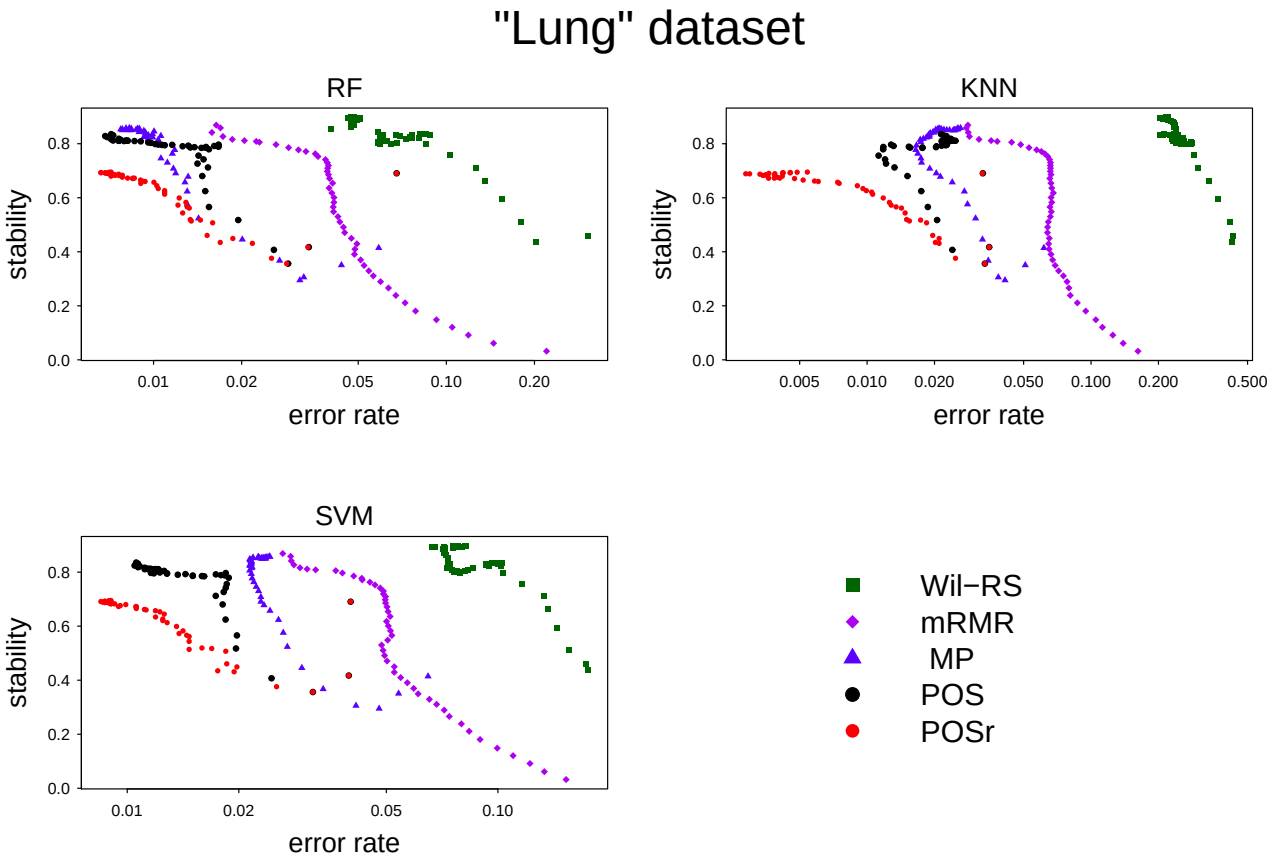

Figure 6.4: *The stability of the feature selection methods against the corresponding estimated error rates on 'Lung' dataset. The error rates have been measured by 50 repetitions of 10-fold cross validation technique for three different classifiers: Random Forest (RF); k Nearest Neighbor (kNN); Support Vector Machine (SVM).*

6.3 Summary

Genes selected according to a uni-variate relevance score, which treats each gene separately, could have a rich redundancy degree among the resulted selections. To handle this situation, the POS technique is further extended to minimize the potential redundancy among the selected genes.

A scheme for minimizing selection redundancy is proposed. It uses a recursive approach to assign a set of complementary discriminative genes. The proposed scheme, named POSr, exploits the gene masks defined by POS to identify more integrated genes in terms of their classification patterns. POSr detects minimum subsets of genes in a successive way. The final selection is then produced by combining these subsets in order to reduce the redundancy among selected genes.

The experimental results of the classification error rates show that the POSr provides an effective approach in enhancing the prediction classification performance using less number of features, by minimizing selection redundancy. The POSr technique produces stable selections for different sizes of selected gene sets.

The next chapter concludes the overall thesis and discusses future plans.

Chapter 7

Conclusions and Future Plans

7.1 Conclusions

A statistical learning approach can be used to model and understand complex datasets. By mapping the relationship between a set of features and a considered response, it can build a predictive model based on a given training data. Both the prediction accuracy and interpretability of the constructed model can be improved by performing the learning process based only on selected relevant features to the considered response.

The foremost task of statistical learning is the classification which has applications encompassing many important fields in modern biology, including analysis of microarray data as well as other functional genomic experiments. Measurements of tens of thousands of genes (features) are observed simultaneously for each observation (tissue sample). This characteristic of high dimensionality has a great effect on the learning process from gene expression microarray data since most of genes are noisy, redundant or non-relevant to the considered learning task. In addition, the limited number of observations, tens to few

hundreds, compared to the number of features provides another challenge for the learning task.

The idea of selecting genes based on analysing the overlap between their expressions across two classes (phenotypes), taking into account the proportions of overlapping observations, is considered in this thesis. To this end, intervals of core gene expressions are defined. A gene mask that allows reporting a gene's predictive power avoiding the effects of outliers is robustly constructed for each gene. A novel score, named the Proportional Overlapping Score (*POS*), is then proposed by which a gene's overlapping degree is estimated. The constructed gene masks along-with the gene scores are utilized to assign the minimum subset of genes that provide the maximum number of correctly classified observations in a training set. This minimum subset of genes is then combined with the top ranked genes according to the *POS* to produce a final gene selection.

Genes selected according to a uni-variate relevance score, which treats each gene separately, could have a rich redundancy degree among the resulted selections, because the set of the top ranked genes may include redundant features. To handle this situation, the idea is further extended to minimize the potential redundancy among the selected genes. A scheme for minimizing selection redundancy is proposed. It extends the Proportional Overlapping Score (POS) technique by using a recursive approach to assign a set of complementary discriminative genes. The proposed scheme, named POSr, exploits the gene masks defined by POS to identify more integrated genes in terms of their classification patterns. POSr detects minimum subsets of genes in a successive way. The final selection is then produced by combining these subsets in order to reduce the redundancy among selected genes.

The proposed procedures, POS and its extended version POSr, are applied on eleven publicly available gene expression datasets with different characteristics. Feature sets of different sizes, up to 50 genes, are selected using widely used gene selection methods: Wilcoxon Rank Sum (Wil-RS); Minimum redundancy maximum relevance (mRMR); MaskedPainter (MP); Iteratively sure independence screening (ISIS) along-with our proposal. The prediction models of three different classifiers: Random Forest; k Nearest Neighbor; Support Vector Machine are constructed with the selected features. The estimated classification error rates obtained by the considered classifiers using 50 repetitions of 10-fold cross validation technique are used for evaluating the performance of POS and POSr.

For the Random Forest classifier, POS and POSr performed better than the compared feature selection methods on the 'Leukaemia', 'Breast', 'GSE24514' and 'GSE4045' datasets at all gene set sizes that have been investigated. For the 'Lung', 'All' and 'Srbct' datasets, POS outperformed all other methods at: small (i.e., less than 7); moderate and large (i.e., > 2); large (i.e., > 5) sets of genes respectively. On average, POS improves the compared techniques by between 5% and 51% of the misclassification error rates achieved by their candidates. POSr provides the minimum error rates among all compared methods for the 'Leukaemia', 'Lung', 'Srbct' and 'GSE4045' datasets.

For the k Nearest Neighbor classifier, POS outperformed all other methods on 'Leukaemia', 'Breast', 'Lung' and 'GSE27854'. While it shows a comparable performance to the MaskedPainter method on the 'Srbct'. On average across all considered datasets, POS approach improves the best performance of the compared methods by up to 20% of the misclassification error rates achieved using their selections at small set sizes less than 20 features.

POSr provides the minimum error rates among the compared methods on the 'Leukaemia', 'Lung', 'Breast' and 'GSE4045' datasets.

For the Support Vector Machine classifier, POS and POSr outperformed all other methods on 'Leukaemia', 'Breast', 'Srbct', 'Lung', 'GSE4045' and 'GSE24514' datasets. Whereas the MaskedPainter provides the minimum error rates on the 'GSE14333' dataset whilst the Wilcoxon Rank Sum is the best method for 'GSE27854' data. On average across all considered datasets, POS and POSr approaches improves the best performance of the compared methods by up to 26% of the misclassification error rates achieved using their selections at different set sizes.

POS is an effective feature selection approach for identifying discriminative genes in respect of a considered classification problem. Experimental results demonstrate that it achieves the best performance, compared with the other feature selection methods, with the three different classifiers. POSr is an effective approach in enhancing the prediction classification performance of the considered classifier models using less number of features, by minimizing selection redundancy, compared to the other studied gene selection methods.

The stability of the selections yielded by the compared feature selection methods using the cross validation technique has been highlighted. Stability scores computed at different set sizes of the selected features show that the proposed approaches have a stable performance for different sizes of selected features. The analysed relationship between classification accuracies yielded by the three different classifiers and stability confirms that the proposal can provide a good trade-off between stability and classification accuracy.

All procedures described in this thesis have been programmed into an R package named

'propOverlap'. It is publicly available for download from the Comprehensive R Archive Network (CRAN) repository (Mahmoud et al. 2014b). The reference manual is reported in Appendix C.

7.2 Future Plans

There are many ideas that are briefly discussed here and will provide directions for future research.

- This work focuses on analysing the overlapping between gene expressions for binary classification problems. One can investigate the possibility of extending POS approach to handle multi-class situations.

- Constructing a framework for POS in which mutual information between genes are considered in the final gene set might be another useful direction. Such a framework could be effective in selecting the discriminative genes with a lower degree of dependency.

- The defined score *POS* measures the overlapping degree by the means of a uni-variate basis. It treats independently each gene. One of the future plans is to examine the possibility of measuring the overlap between expressions of different classes using a multivariate approach

- All the work presented in this thesis is related to feature selection procedures within functional genomic experiments and their effects in enhancing statistical learning. Applying the proposal on datasets from different domains as well as different kinds

of features can also be another direction.

- The defined gene masks characterize genes according to their role in the classification problem. Clustering genes based on their masks can be one of the future plans. The idea may be then extended for applications of other statistical learning tasks.

Appendix A

Availability of Supporting Data

The datasets supporting the results of this thesis are publicly available on various database sources. These datasets are briefly described in the following sections.

A.1 The Lung Dataset

Lung Cancer Classification between malignant pleural mesothelioma (MPM) and adenocarcinoma (ADCA) of the lung. There are 181 observations (tissue samples), 31 MPM and 150 ADCA. Each observation is described by 12533 features (genes). The dataset can be downloaded from [http://cilab.ujn.edu.cn/datasets.htm].

A.2 The Leukaemia Dataset

It was taken from a collection of leukaemia patient samples reported by Golub et al. (1999*b*). This dataset often serves as benchmark for microarray analysis methods. It contains measurements corresponding to Acute Lymphoblast Leukaemia (ALL) and Acute Myeloid

Leukaemia (AML) samples from bone marrow and peripheral blood. It consisted of 72 observations: 25 samples of AML; 47 samples of ALL. Each observation is measured over 7129 features (genes). The dataset can be downloaded from [http://cilab.ujn.edu.cn/datasets.htm].

A.3 The Srbct Dataset

The small, round blue cell tumors (SRBCTs) of childhood, which include NeuroBlastoma (NB), RhabdoMyoSarcoma (RMS), Non-Hodgkin Lymphoma (NHL) and the EWing family of tumors (EWS) are so named because of their similar appearance on routine histology. The dataset consists of 83 observations, 29, 25, 11 and 18 observations of NB, RMS, NHL and EWS respectively described by 2308 genes. Since this thesis considers only binary classification problems, the two classes with the topmost number of observations, i.e. NB and RMS, are only considered for the analysis. The dataset is available in [http://www.gems-system.org/].

A.4 The Prostate Dataset

Microarray gene expressions of 102 patients, 52 patients with prostate tumors and 50 normal patients, are reported. The dataset includes 10509 genes. It can be downloaded from [http://www.gems-system.org/].

A.5 The Carcinoma Dataset

The dataset contains measurements of expression levels in colon adenocarcinomas for 18 patients. Expression levels of the same RNAs were also measured in 18 normal colon tissues. It observes 7457 genes for the considered 36 colon tissues. The dataset is available in [http://genomics-pubs.princeton.edu/oncology/].

A.6 The Colon Dataset

The data contains expression measurements for 62 colon tissues (40 tumors and 22 normal samples) observed over 2000 genes. It is available in the [Bioconductor] repository, [http://www.bioconductor.org/] from the R package 'ColonCA'.

A.7 The All Dataset

The dataset consists of microarrays from 128 different individuals with Acute Lymphoblastic Leukaemia (ALL). It classifies patients into two classes based on the type and stage of the disease: 95 observations from B-cell ALL; 33 observations from T-cell ALL. Measurements of 12625 genes are reported. The dataset is available in the [Bioconductor] repository, [http://www.bioconductor.org/] from the R package 'ALL'.

A.8 The Breast Dataset

Gene expression data from breast cancer study are reported in this dataset. It includes 4948 genes measured in 78 patients: 34 with Distant Metastases (DM); 44 without distant Metas-

tases (NODM). It is available in the [Bioconductor] repository, [http://www.bioconductor.org/] from the R package 'cancerdata'.

A.9 The GSE24514 Dataset

Expression profiling of 34 MicroSatellite Instability (MSI) colorectal cancers and 15 normal colonic mucosas was performed using Affymetrix Human Genome U133A Array (HG-U133A). It is available in the [Gene Expression Omnibus (GEO)] repository [http://www.ncbi.nlm.n id's:GSE24514].

A.10 The GSE4045 Dataset

The dataset includes gene expressions of 37 colorectal cancer (CRC) tumors: 29 serrated CRC; 8 conventional CRC. The study was performed using Affymetrix Human Genome U133A Array (HG-U133A). It is available in the [Gene Expression Omnibus (GEO)] repository [http://www.ncbi.nlm.nih.gov/geo/][accession id's:GSE4045].

A.11 The GSE14333 Dataset

The mRNA from 290 primary colorectal tumour samples were extracted and hybridized to Affymetrix Human Genome U133 Plus 2.0 Array (HG-U133 Plus 2). The dataset contains 44, 94, 91 and 61 observations with colorectal cancer of tumor Duck stages A, B, C and D respectively. Patients with stages A and B are combined in a single class representing non-invasive tumors, against patients with stage C, which represents invasive tumors.

While, stage D is excluded from the analysis. The data is available in the [Gene Expression Omnibus (GEO)] repository [http://www.ncbi.nlm.nih.gov/geo/][accession id's:GSE14333].

A.12 The GSE27854 Dataset

Expression profiles in 115 patients with colorectal cancer tumors were invistigated using an Affymetrix Human Genome U133 Plus 2.0 Array (HG-U133 Plus 2). The data consists of 16, 41, 35 and 23 observations of tumor 'Union Internationale Contre le Cancer' (UICC) stages I, II, III and IV respectively. A class composed of patients with stages I and II is defined against another class involving patients with III and IV stages. It is available in the [Gene Expression Omnibus (GEO)] repository [http://www.ncbi.nlm.nih.gov/geo/][accession id's:GSE27854].

Appendix B

Classification Error Rates

B.1 Classification Error Rates Obtained by Random Forest Using different feature selection methods

Average classification error rates yielded by the Random Forest classifier using Wilcoxon rank sum (Wil-RS), Minimum redundancy maximum relevance (mRMR), MaskedPainter (MP) and proportional overlapping scores (POS) feature selection techniques on 'Breast', 'Srbct', 'Prostate', 'All', 'Lung', 'Carcinoma', 'GSE4045', 'GSE14333' and 'GSE27854' datasets over 50 repetitions of 10-fold cross validation are presented in nine tables, a table for each dataset. Each row provides the average classification error rate at a specific number of selected genes (reported in the first column).

Table B.1: *'Breast' dataset*

N.genes	Wil-RS	mRMR	MP	POS
2	0.437	0.483	0.499	0.333
3	0.439	0.475	0.474	0.330
4	0.433	0.471	0.463	0.349
5	0.428	0.474	0.453	0.342
6	0.422	0.462	0.443	0.350
7	0.421	0.458	0.436	0.340
8	0.426	0.458	0.433	0.336
9	0.418	0.459	0.432	0.331
10	0.417	0.459	0.425	0.338
20	0.396	0.448	0.374	0.331
30	0.375	0.442	0.367	0.318
40	0.384	0.425	0.365	0.317
50	0.371	0.407	0.357	0.310

Table B.2: *'Srbct' dataset*

N.genes	Wil-RS	mRMR	MP	POS
2	0.083	0.414	0.046	0.085
3	0.095	0.369	0.013	0.064
4	0.103	0.335	0.029	0.041
5	0.081	0.315	0.026	0.028
6	0.079	0.281	0.026	0.025
7	0.088	0.272	0.028	0.022
8	0.091	0.246	0.026	0.018
9	0.092	0.229	0.027	0.022
10	0.097	0.222	0.026	0.022
20	0.085	0.153	0.012	0.006
30	0.077	0.097	0.011	0.006
40	0.081	0.095	0.012	0.006
50	0.090	0.077	0.014	0.006

Table B.3: *'Prostate' dataset*

N.genes	Wil-RS	mRMR	MP	POS
2	0.435	0.415	0.117	0.117
3	0.481	0.390	0.098	0.083
4	0.470	0.366	0.097	0.088
5	0.452	0.345	0.092	0.086
6	0.433	0.324	0.089	0.091
7	0.432	0.313	0.087	0.088
8	0.433	0.294	0.090	0.092
9	0.445	0.274	0.088	0.092
10	0.440	0.265	0.085	0.088
20	0.218	0.192	0.070	0.071
30	0.215	0.157	0.071	0.068
40	0.216	0.158	0.071	0.065
50	0.200	0.140	0.069	0.062

Table B.4: *'All' dataset*

N.genes	Wil-RS	mRMR	MP	POS
2	0.314	0.315	0.020	0.027
3	0.295	0.287	0.012	0.011
4	0.284	0.284	0.016	0.010
5	0.269	0.269	0.010	0.005
6	0.262	0.261	0.007	0.005
7	0.262	0.250	0.006	0.006
8	0.260	0.239	0.005	0.002
9	0.272	0.236	0.006	0.005
10	0.274	0.229	0.005	0.005
20	0.246	0.114	0.002	0.000
30	0.225	0.036	0.001	0.000
40	0.143	0.024	0.000	0.000
50	0.152	0.011	0.000	0.000

Table B.5: *'Lung' dataset*

N.genes	Wil-RS	mRMR	MP	POS
2	0.202	0.144	0.043	0.034
3	0.179	0.120	0.032	0.029
4	0.155	0.106	0.033	0.026
5	0.135	0.092	0.026	0.019
6	0.126	0.079	0.021	0.015
7	0.103	0.071	0.014	0.016
8	0.085	0.067	0.013	0.013
9	0.081	0.062	0.013	0.015
10	0.081	0.059	0.013	0.014
20	0.068	0.044	0.010	0.013
30	0.040	0.041	0.009	0.009
40	0.047	0.031	0.008	0.008
50	0.048	0.016	0.008	0.007

Table B.6: *'Carcinoma' dataset*

N.genes	Wil-RS	mRMR	MP	POS
2	0.472	0.347	0.038	0.073
3	0.508	0.280	0.025	0.056
4	0.290	0.243	0.022	0.063
5	0.270	0.221	0.019	0.049
6	0.120	0.199	0.022	0.056
7	0.114	0.182	0.023	0.049
8	0.097	0.157	0.025	0.036
9	0.095	0.140	0.026	0.038
10	0.081	0.131	0.023	0.035
20	0.028	0.065	0.027	0.026
30	0.011	0.039	0.025	0.027
40	0.004	0.023	0.026	0.027
50	0.004	0.021	0.027	0.029

Table B.7: *'GSE4045' dataset*

N.genes	Wil-RS	mRMR	MP	POS
2	0.201	0.266	0.208	0.186
3	0.195	0.245	0.180	0.152
4	0.193	0.236	0.194	0.156
5	0.178	0.223	0.172	0.126
6	0.182	0.228	0.169	0.129
7	0.177	0.218	0.160	0.130
8	0.176	0.213	0.155	0.136
9	0.172	0.217	0.151	0.132
10	0.172	0.211	0.147	0.132
20	0.143	0.193	0.141	0.125
30	0.140	0.197	0.149	0.122
40	0.141	0.192	0.152	0.121
50	0.143	0.192	0.154	0.134

Table B.8: *'GSE14333' dataset*

N.genes	Wil-RS	MP	POS
2	0.454	0.491	0.474
3	0.440	0.485	0.481
4	0.439	0.490	0.476
5	0.435	0.483	0.478
6	0.428	0.475	0.474
7	0.426	0.472	0.474
8	0.424	0.472	0.475
9	0.423	0.471	0.475
10	0.421	0.471	0.474
20	0.438	0.460	0.461
30	0.449	0.438	0.445
40	0.453	0.445	0.441
50	0.452	0.442	0.445

Table B.9: *'GSE27854' dataset*

N.genes	Wil-RS	MP	POS
2	0.446	0.475	0.465
3	0.431	0.467	0.467
4	0.420	0.466	0.465
5	0.418	0.462	0.452
6	0.408	0.457	0.451
7	0.410	0.459	0.452
8	0.407	0.459	0.462
9	0.405	0.448	0.459
10	0.409	0.450	0.463
20	0.405	0.456	0.473
30	0.414	0.451	0.471
40	0.408	0.450	0.472
50	0.416	0.444	0.484

B.2 Classification Error Rates Obtained by k Nearest Neighbor Using different feature selection methods

Average classification error rates yielded by the k Nearest Neighbor classifier using Wilcoxon rank sum (Wil-RS), Minimum redundancy maximum relevance (mRMR), MaskedPainter (MP) and proportional overlapping scores (POS) feature selection techniques on 'Breast', 'Srbct', 'Lung', 'GSE4045', 'GSE14333' and 'GSE27854' datasets over 50 repetitions of 10-fold cross validation are presented in six tables, a table for each dataset. Each row provides the average classification error rate at a specific number of selected genes (reported in the first column).

Table B.10: *'Breast' dataset*

N.genes	Wil-RS	mRMR	MP	POS
1	0.425	0.486	0.510	0.417
2	0.463	0.474	0.480	0.368
3	0.471	0.465	0.468	0.360
4	0.457	0.452	0.447	0.356
5	0.467	0.456	0.434	0.360
6	0.470	0.453	0.432	0.348
7	0.470	0.450	0.418	0.351
8	0.454	0.449	0.415	0.342
9	0.451	0.441	0.417	0.340
10	0.441	0.439	0.406	0.337
20	0.430	0.423	0.348	0.340
30	0.438	0.431	0.354	0.341
40	0.417	0.413	0.359	0.362
50	0.438	0.404	0.354	0.377

Table B.11: *'Srbct' dataset*

N.genes	Wil-RS	mRMR	MP	POS
1	0.422	0.452	0.064	0.135
2	0.180	0.435	0.094	0.098
3	0.157	0.398	0.054	0.063
4	0.185	0.376	0.043	0.050
5	0.189	0.361	0.043	0.051
6	0.252	0.345	0.042	0.046
7	0.299	0.332	0.040	0.044
8	0.364	0.321	0.041	0.040
9	0.394	0.308	0.041	0.040
10	0.424	0.308	0.041	0.037
20	0.367	0.265	0.018	0.007
30	0.383	0.246	0.005	0.008
40	0.477	0.192	0.008	0.009
50	0.460	0.099	0.011	0.009

Table B.12: *'Lung' dataset*

N.genes	Wil-RS	mRMR	MP	POS
1	0.430	0.162	0.062	0.033
2	0.426	0.139	0.051	0.035
3	0.416	0.125	0.041	0.034
4	0.370	0.112	0.039	0.024
5	0.335	0.105	0.035	0.021
6	0.301	0.095	0.033	0.019
7	0.290	0.087	0.031	0.017
8	0.281	0.081	0.028	0.015
9	0.257	0.080	0.027	0.013
10	0.234	0.078	0.024	0.012
20	0.239	0.064	0.017	0.018
30	0.239	0.065	0.020	0.022
40	0.217	0.057	0.023	0.025
50	0.213	0.028	0.026	0.021

Table B.13: *'GSE4045' dataset*

N.genes	Wil-RS	mRMR	MP	POS
1	0.180	0.235	0.227	0.213
2	0.132	0.228	0.172	0.165
3	0.125	0.231	0.153	0.142
4	0.124	0.227	0.144	0.166
5	0.115	0.225	0.148	0.160
6	0.115	0.231	0.153	0.172
7	0.114	0.229	0.147	0.171
8	0.111	0.231	0.149	0.171
9	0.106	0.229	0.148	0.166
10	0.110	0.221	0.150	0.165
20	0.083	0.211	0.147	0.157
30	0.070	0.221	0.147	0.154
40	0.070	0.209	0.144	0.158
50	0.078	0.214	0.137	0.166

Table B.14: *'GSE14333' dataset*

N.genes	Wil-RS	MP	POS
1	0.469	0.472	0.472
2	0.443	0.480	0.476
3	0.434	0.488	0.474
4	0.431	0.488	0.478
5	0.429	0.489	0.482
6	0.422	0.487	0.483
7	0.419	0.483	0.483
8	0.419	0.480	0.482
9	0.420	0.475	0.482
10	0.423	0.476	0.483
20	0.443	0.459	0.470
30	0.454	0.468	0.456
40	0.453	0.482	0.456
50	0.453	0.497	0.456

Table B.15: *'GSE27854' dataset*

N.genes	Wil-RS	MP	POS
1	0.432	0.491	0.492
2	0.427	0.463	0.483
3	0.420	0.460	0.472
4	0.426	0.463	0.463
5	0.434	0.459	0.439
6	0.431	0.460	0.436
7	0.437	0.467	0.431
8	0.436	0.469	0.431
9	0.441	0.463	0.428
10	0.442	0.464	0.428
20	0.453	0.480	0.431
30	0.449	0.476	0.425
40	0.454	0.474	0.432
50	0.456	0.475	0.437

B.3 Classification Error Rates Obtained by Support Vector Machine Using different feature selection methods

Average classification error rates yielded by the Support Vector Machine classifier using Wilcoxon rank sum (Wil-RS), Minimum redundancy maximum relevance (mRMR), MaskedPainter (MP) and proportional overlapping scores (POS) feature selection techniques on 'Breast', 'Srbct', 'Lung', 'GSE4045', 'GSE14333' and 'GSE27854' datasets over 50 repetitions of 10-fold cross validation are presented in six tables, a table for each dataset. Each row provides the average classification error rate at a specific number of selected genes (reported in the first column).

Table B.16: *'Breast' dataset*

N.genes	Wil-RS	mRMR	MP	POS
1	0.417	0.481	0.488	0.398
2	0.422	0.470	0.458	0.324
3	0.432	0.466	0.445	0.359
4	0.485	0.454	0.432	0.338
5	0.489	0.461	0.424	0.339
6	0.497	0.457	0.415	0.334
7	0.496	0.456	0.410	0.343
8	0.486	0.447	0.407	0.342
9	0.480	0.446	0.404	0.332
10	0.468	0.444	0.405	0.328
20	0.426	0.443	0.363	0.321
30	0.401	0.440	0.366	0.321
40	0.413	0.426	0.365	0.324
50	0.411	0.407	0.361	0.329

Table B.17: *'Srbct' dataset*

N.genes	Wil-RS	mRMR	MP	POS
1	0.421	0.454	0.068	0.134
2	0.176	0.419	0.041	0.099
3	0.174	0.397	0.014	0.060
4	0.152	0.361	0.017	0.029
5	0.143	0.343	0.012	0.018
6	0.149	0.311	0.011	0.006
7	0.150	0.303	0.015	0.003
8	0.165	0.286	0.015	0.003
9	0.184	0.268	0.016	0.004
10	0.180	0.266	0.017	0.004
20	0.177	0.213	0.011	0.018
30	0.152	0.161	0.023	0.022
40	0.162	0.147	0.025	0.019
50	0.131	0.126	0.029	0.018

Table B.18: *'Lung' dataset*

N.genes	Wil-RS	mRMR	MP	POS
1	0.173	0.153	0.065	0.040
2	0.175	0.133	0.054	0.040
3	0.155	0.122	0.048	0.032
4	0.144	0.110	0.041	0.025
5	0.136	0.100	0.034	0.020
6	0.133	0.090	0.030	0.020
7	0.116	0.084	0.027	0.018
8	0.103	0.080	0.026	0.018
9	0.102	0.074	0.026	0.017
10	0.103	0.072	0.024	0.018
20	0.081	0.049	0.022	0.015
30	0.073	0.050	0.022	0.013
40	0.080	0.043	0.023	0.012
50	0.066	0.026	0.024	0.011

Table B.19: *'GSE4045' dataset*

N.genes	Wil-RS	mRMR	MP	POS
1	0.201	0.330	0.221	0.249
2	0.201	0.266	0.186	0.186
3	0.195	0.245	0.166	0.152
4	0.193	0.236	0.153	0.156
5	0.178	0.223	0.149	0.126
6	0.182	0.228	0.154	0.129
7	0.177	0.218	0.143	0.130
8	0.176	0.213	0.143	0.136
9	0.172	0.217	0.139	0.132
10	0.172	0.211	0.134	0.132
20	0.143	0.193	0.114	0.125
30	0.140	0.197	0.106	0.122
40	0.141	0.192	0.098	0.121
50	0.143	0.192	0.098	0.134

Table B.20: *'GSE14333' dataset*

N.genes	Wil-RS	MP	POS
1	0.447	0.412	0.431
2	0.453	0.447	0.455
3	0.440	0.463	0.461
4	0.437	0.465	0.464
5	0.431	0.465	0.471
6	0.429	0.461	0.471
7	0.430	0.454	0.475
8	0.427	0.452	0.474
9	0.427	0.451	0.475
10	0.431	0.454	0.470
20	0.460	0.440	0.472
30	0.468	0.432	0.460
40	0.472	0.423	0.463
50	0.468	0.423	0.462

Table B.21: *'GSE27854' dataset*

N.genes	Wil-RS	MP	POS
1	0.435	0.465	0.494
2	0.458	0.483	0.480
3	0.447	0.479	0.470
4	0.443	0.477	0.479
5	0.445	0.485	0.466
6	0.440	0.476	0.470
7	0.443	0.477	0.459
8	0.443	0.483	0.456
9	0.438	0.477	0.459
10	0.445	0.475	0.464
20	0.436	0.477	0.468
30	0.440	0.482	0.466
40	0.434	0.475	0.472
50	0.439	0.472	0.475

Appendix C

Reference Manual for the developed R Package ‘propOverlap’

Package 'propOverlap'

February 20, 2015

Type Package

Title Feature (gene) selection based on the Proportional Overlapping Scores

Version 1.0

Date 2014-09-15

Author
Osama Mahmoud, Andrew Harrison, Aris Perperoglou, Asma Gul, Zardad Khan, Berthold Lausen

Maintainer Osama Mahmoud <ofamah@essex.ac.uk>

Description A package for selecting the most relevant features (genes) in the high-dimensional binary classification problems. The discriminative features are identified using analyzing the overlap between the expression values across both classes. The package includes functions for measuring the proportional overlapping score for each gene avoiding the outliers effect. The used measure for the overlap is the one defined in the ``Proportional Overlapping Score (POS)" technique for feature selection. A gene mask which represents a gene's classification power can also be produced for each gene (feature). The set size of the selected genes might be set by the user. The minimum set of genes that correctly classify the maximum number of the given tissue samples (observations) can be also produced.

Depends R (>= 2.10), Biobase

LazyLoad yes

License GPL (>= 2)

Repository CRAN

NeedsCompilation no

Date/Publication 2014-09-15 17:06:03

R topics documented:

`propOverlap-package` *Feature (gene) selection based on the Proportional Overlapping Scores.*

Description

A package for selecting the most relevant features (genes) in the high-dimensional binary classification problems. The discriminative features are identified using analyzing the overlap between the expression values across both classes. The package includes functions for measuring the proportional overlapping score for each gene avoiding the outliers effect. The used measure of the overlap is the one defined in the "Proportional Overlapping Score (**POS**)" technique for feature selection, see 'References' section below. A gene mask which represents a gene's classification power can also be produced for each gene (feature). The set size of the selected genes might be set by the user. The minimum set of genes that correctly classify the maximum number of the given tissue samples (observations) can be also produced.

Details

Package:	propOverlap
Type:	Package
Version:	1.0
Date:	2014-09-15
License:	GPL (>= 2)

Author(s)

Osama Mahmoud, Andrew Harrison, Aris Perperoglou, Asma Gul, Zardad Khan, Berthold Lausen
Maintainer: Osama Mahmoud <ofamah@essex.ac.uk>

References

Mahmoud O., Harrison A., Perperoglou A., Gul A., Khan Z., Metodiev M. and Lausen B. (2014) *A feature selection method for classification within functional genomics experiments based on the proportional overlapping score*. BMC Bioinformatics, 2014, 15:274

CI.emprical *Computing the Core Intervals for Both Classes.*

Description

CI.emprical is used to compute the core interval boundaries for each class.

Usage

```
CI.emprical(ES, Y)
```

Arguments

ES	gene (feature) matrix: P, number of genes, by N, number of samples(observations).
Y	a vector of length N for samples' class label.

Value

CI.emprical returns an object of class "data.frame" which has P rows and 4 columns. The first two columns represent a1, the minimum boundary of the first class, and b1, the maximum boundary of the first class, respectively. Whereas, the last two columns represent a2, the minimum boundary of the second class, and b2, the maximum boundary of the second class, respectively.

Author(s)

Osama Mahmoud <ofamah@essex.ac.uk>

References

Mahmoud O., Harrison A., Perperoglou A., Gul A., Khan Z., Metodiev M. and Lausen B. (2014) *A feature selection method for classification within functional genomics experiments based on the proportional overlapping score*. BMC Bioinformatics, 2014, 15:274.

Examples

```
data(lung)
GenesExpression <- lung[1:12533,]   #define the features matrix
Class           <- lung[12534,]     #define the observations' class labels
CoreIntervals   <- CI.emprical(GenesExpression, Class)
CoreIntervals[1:10,]              #show classes' core interval for the first 10 features
```

GMask *Producing Gene Masks.*

Description

GMask produces the masks of features (genes). Each gene mask reports the samples that can unambiguously be assigned to their correct target classes by this gene.

Usage

```
GMask(ES, Core, Y)
```

Arguments

ES	gene (feature) matrix: P, number of genes, by N, number of samples(observations).
Core	a data.frame of the core interval boundaries for both classes. It should have the same number of rows as ES and 4 columns (the minimum and the maximum of the first class's core interval followed by the minimum and the maximum of the second class's core interval). See the returned value of the CI.emprical.
Y	a vector of length N for samples' class label.

Details

GMask gives the gene masks that can represent the capability of genes to correctly classify each sample. Such a mask represents a gene's classification power. Each element of a mask is set either to 1 or 0 based on whether the corresponding sample (observation) could be unambiguously assign to its correct target class by the considered gene or not respectively.

Value

It returns a P by N matrix with elements of zeros and ones.

Author(s)

Osama Mahmoud <ofamah@essex.ac.uk>

References

Mahmoud O., Harrison A., Perperoglou A., Gul A., Khan Z., Metodiev M. and Lausen B. (2014) *A feature selection method for classification within functional genomics experiments based on the proportional overlapping score*. BMC Bioinformatics, 2014, 15:274.

See Also

CI.emprical for the core interval boundaries.

Examples

```
data(leukaemia)
GenesExpression <- leukaemia[1:7129,] #define the features matrix
Class           <- leukaemia[7130,]   #define the observations' class labels
Gene.Masks      <- GMask(GenesExpression, CI.emprical(GenesExpression, Class), Class)
Gene.Masks[1:100,]                    #show the masks of the first 100 features
```

`leukaemia` *Leukaemia data set.*

Description

The leukemia dataset was taken from a collection of leukemia patient samples reported by Golub et. al., (1999). This dataset often serves as a benchmark for microarray analysis methods. It contains gene expressions corresponding to acute lymphoblast leukemia (ALL) and acute myeloid leukemia (AML) samples from bone marrow and peripheral blood. The dataset consisted of 72 samples: 49 samples of ALL; 23 samples of AML. Each sample is measured over 7,129 genes.

Usage

```
data(leukaemia)
```

Format

A matrix with 7130 rows (7129 rows show the gene expressions while the last row reports the corresponding sample's class label), and 72 columns represent the samples. The samples class's label coded as follows:

`1` acute lymphoblast leukemia sample (ALL).

`2` acute myeloid leukemia sample (AML).

Source

http://cilab.ujn.edu.cn/datasets.htm

References

Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh ML, Downing JR, Caligiuri MA, Bloomfield CD, Lander ES. (1999) *Molecular classification of cancer: class discovery and class prediction by gene expression monitoring*. Science: 286 (5439), 531-537.

Examples

```
data(leukaemia)
str(leukaemia)
```

lung *Lung cancer data set.*

Description

Gene expression data for lung cancer classification between two classes: adenocarcinoma (ADCA); malignant pleural mesothe-lioma (MPM). The lung data set contains 181 tissue samples (150 ADCA and 31 MPM). Each sample is described by 12533 genes.

Usage

```
data(lung)
```

Format

A matrix with 12534 rows (12533 rows show the gene expressions for 181 tissue samples, reported in columns, while the last row reports the corresponding sample's class label). The samples class's label coded as follows:

1 adenocarcinoma sample (ADCA).

2 malignant pleural mesothe-lioma sample (MPM).

Source

http://cilab.ujn.edu.cn/datasets.htm

References

Gordon GJ, Jensen RV, Hsiao L-L, Gullans SR, Blumenstock JE, Ramaswamy S, Richards WG, Sugarbaker DJ, Bueno R. (2002) *Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma*. Cancer research: 62(17), 4963-4967.

Examples

```
data(lung)
str(lung)
```

POS *Calculating the proportional Overlapping Scores.*

Description

POS computes the proportional overlapping scores of the given genes (features). This score measures the overlap degree between gene expression values across various classes. It produces a value lies in the interval [0,1]. A lower score denotes gene with higher discriminative power for the considered classification problem.

Usage

```
POS(ES, Core, Y)
```

Arguments

ES	gene (feature) matrix: P, number of genes, by N, number of samples(observations).
Core	a data.frame of the core interval boundaries for both classes. It should have the same number of rows as ES and 4 columns (the minimum and the maximum of the first class's core interval followed by the minimum and the maximum of the second class's core interval). See the returned value of the CI.emprical.
Y	a vector of length N for samples' class label.

Details

For each gene, POS computes a measure that estimates the overlapping degree between the expression intervals of different classes. For estimating the overlap, POS measure takes into account three factors: the length of the overlapping region; number of the overlapped samples (observations); the proportion of each class's overlapped samples to the total number of overlapping samples.

Value

It returns a vector of length P for 'POS' measures of all genes (features).

Author(s)

Osama Mahmoud <ofamah@essex.ac.uk>

References

Mahmoud O., Harrison A., Perperoglou A., Gul A., Khan Z., Metodiev M. and Lausen B. (2014) *A feature selection method for classification within functional genomics experiments based on the proportional overlapping score*. BMC Bioinformatics, 2014, 15:274.

See Also

CI.emprical for the core interval boundaries and GMask for the gene masks.

Examples

```
data(leukaemia)
Score <- POS(leukaemia[1:7129,], CI.emprical(leukaemia[1:7129,],
leukaemia[7130,]), leukaemia[7130,])
Score[1:5]      #show the proportional overlapping scores for the first 5 features
summary(Score)  #show the the summary of the scores of all features.
```

RDC *Assiging the Relative Dominant Class.*

Description

RDC associates genes (features) with the class which it is more able to distingish. For each gene, a class that has the highest proportion, relative to classes' size, of correctly assigned samples (observations) is reported as the relative dominant class for the considered gene.

Usage

```
RDC(GMask, Y)
```

Arguments

GMask	gene (feature) mask matrix: P, number of genes, by N, number of samples(observations) with elements of zeros and ones. See the returned value of the GMask.
Y	a vector of length N for samples' class label.

Value

RDC returns a vector of length P. Each element's value is either 1 or 2 indicating which class label is reported as the relative dominant class for the corresponding gene (feature).

Author(s)

Osama Mahmoud <ofamah@essex.ac.uk>

References

Mahmoud O., Harrison A., Perperoglou A., Gul A., Khan Z., Metodiev M. and Lausen B. (2014) *A feature selection method for classification within functional genomics experiments based on the proportional overlapping score*. BMC Bioinformatics, 2014, 15:274.

See Also

GMask for gene (feature) mask matrix.

Examples

```
data(lung)
Class           <- lung[12534,]   #define the observations' class labels
Gene.Masks      <- GMask(lung[1:12533,], CI.emprical(lung[1:12533,], Class), Class)
RelativeDC      <- RDC(Gene.Masks, Class)
RelativeDC[1:10]              #show the relative dominant classes for the first 10 features
table(RelativeDC)                 #show the number of assignments for each class
```

`Sel.Features` *Gene (Feature) Selection.*

Description

`Sel.Feature` selects the most discriminative genes (features) among the given ones.

Usage

```
Sel.Features(ES, Y, K = "Min", Verbose = FALSE)
```

Arguments

`ES`	gene (feature) matrix: P, number of genes, by N, number of samples (observations).
`Y`	a vector of length N for samples' class label.
`K`	the number of genes to be selected. The default is to give the minimum subset of genes that correctly classify the maximum number of the given tissue samples (observations). Alternatively, `K` should be a positive integer.
`Verbose`	logical. If `TRUE`, more information about the selected genes are returned.

Details

`Sel.Feature` selects the most relevant genes (features) in the high-dimensional binary classification problems. The discriminative genes are identified using analyzing the overlap between the expression values across both classes. The "**POS**" technique has been applied to produce the selected set of genes. A proportional overlapping score measures the overlapping degree avoiding the outliers effect for each gene. Each gene is described by a robust mask that represents its discriminative power. The constructed masks along with the gene scores are exploited to produce the selected subset of genes.

Value

If `K` is specified as 'Min' (the default), a list containing the following components is returned:

`Features`	A matrix of the indices of selected genes with their POS measures. See `POS`.
`Covered.Obs`	A vector showing the indices of the observations that have been covered by the returned minimum subset of genes.

If K is specified as a positive integer, a list containing the following components is returned:

features	A vector of the indices of the selected genes.
nMin.Features	The number of genes included in the minimum subset.
Measures	Available only when Verbose is TRUE. It is an object with class "data.frame" which contains 3 columns: the indices of the selected genes; the POS measures of the selected genes (see POS); the status that reports on which basis a gene is selected ("Min.Set": the gene is a member of the selected minimum subset, 1: the gene has a low POS score and its relative dominant class is the first class or 2: the gene has a low POS score and its relative dominant class is the second class), see RDC.

Note

Verbose is only needed when K is specified. If K is set to "Min" (default), all information are automatically returned.

Author(s)

Osama Mahmoud <ofamah@essex.ac.uk>

References

Mahmoud O., Harrison A., Perperoglou A., Gul A., Khan Z., Metodiev M. and Lausen B. (2014) *A feature selection method for classification within functional genomics experiments based on the proportional overlapping score*. BMC Bioinformatics, 2014, 15:274.

See Also

POS for calculating the proportional overlapping scores and RDC for assigning the relative dominant class.

Examples

```
data(leukaemia)
GenesExpression <- leukaemia[1:7129,] #define the features matrix
Class           <- leukaemia[7130,]   #define the observations' class labels
## select the minimum subset of features
Selection       <- Sel.Features(GenesExpression, Class)
attributes(Selection)
(Candidates      <- Selection$Features)   #return the selected features
(Covered.observations <- Selection$Covered.Obs) #return the covered observations by the selection
## select a specific number of features
Selection.k      <- Sel.Features(GenesExpression, Class, K=10, Verbose=TRUE)
Selection.k$Features
Selection.k$nMin.Features   #return the size of the minimum subset of genes
Selection.k$Measures        #return the selected features' information
```

Index

Bibliography

Alhopuro, P., Sammalkorpi, H., Niittymäki, I., Biström, M., Raitila, A., Saharinen, J., Nousiainen, K., Lehtonen, H. J., Heliövaara, E., Puhakka, J. et al. (2012), 'Candidate driver genes in microsatellite-unstable colorectal cancer', *International Journal of Cancer* **130**(7), 1558–1566.

Altman, D. G., Lausen, B., Sauerbrei, W. & Schumacher, M. (1994), 'Dangers of using optimal cutpoints in the evaluation of prognostic factors', *Journal of the National Cancer Institute* **86**(11), 829–835.

Apiletti, D., Baralis, E., Bruno, G. & Fiori, A. (2007*a*), The painter's feature selection for gene expression data, *in* 'Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE', IEEE, pp. 4227–4230.

Apiletti, D., Baralis, E., Bruno, G. & Fiori, A. (2007*b*), The painter's feature selection for gene expression data, *in* 'Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE', IEEE, pp. 4227–4230.

Apiletti, D., Baralis, E., Bruno, G. & Fiori, A. (2012), 'Maskedpainter: Feature selection for microarray data analysis', *Intelligent Data Analysis* **16**(4), 717–737.

Baralis, E., Bruno, G. & Fiori, A. (2008), Minimum number of genes for microarray feature selection, *in* 'Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE', IEEE, pp. 5692–5695.

Bishop, C. M. et al. (2006), *Pattern recognition and machine learning*, Vol. 1, Springer New York.

Boyd, S. & Vandenberghe, L. (2009), *Convex optimization*, Cambridge university press.

Breiman, L. (1984), 'Classification and regression trees'.

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

Breiman, L. (2001), 'Random forests', *Machine Learning* **45**(1), 5–32.

Breiman, L., Friedman, J., Stone, C. & Olshen, R. (1984), *Classification and regression trees*, Chapman & Hall/CRC.

Brown, G. (2009), 'Ensemble learning', *Encyclopedia of Machine Learning* .

Chen, K.-H., Wang, K.-J., Tsai, M.-L., Wang, K.-M., Adrian, A. M., Cheng, W.-C., Yang, T.-S., Teng, N.-C., Tan, K.-P. & Chang, K.-S. (2014), 'Gene selection for cancer identification: a decision tree model empowered by particle swarm optimization algorithm', *BMC Bioinformatics* **15**(1), 49.

Chiaretti, S., Li, X., Gentleman, R., Vitale, A., Vignetti, M., Mandelli, F., Ritz, J. & Foa, R. (2004), 'Gene expression profile of adult t-cell acute lymphocytic leukemia identifies distinct subsets of patients with different response to therapy and survival', *Blood* **103**(7), 2771–2778.

Chipman, H., George, E. & McCulloch, R. (1998), 'Bayesian cart model search', *Journal of the American Statistical Association* pp. 935–948.

Cho, S. & Kim, J. (1995), 'Multiple network fusion using fuzzy logic', *Neural Networks, IEEE Transactions on* **6**(2), 497–501.

Cortes, C. & Vapnik, V. (1995), 'Support-vector networks', *Machine Learning* **20**(3), 273–297.

Cover, T. & Hart, P. (1967), 'Nearest neighbor pattern classification', *Information Theory, IEEE Transactions on* **13**(1), 21–27.

Cunningham, P. & Carney, J. (2000), 'Diversity versus quality in classification ensembles based on feature selection', *Machine Learning: ECML 2000* pp. 109–116.

Cutler, A. & Stevens, J. (2006), '[23] random forests for microarrays', *Methods in enzymology* **411**, 422–432.

De Jay, N., Papillon-Cavanagh, S., Olsen, C., El-Hachem, N., Bontempi, G. & Haibe-Kains, B. (2013), 'mrmre: an r package for parallelized mrmr ensemble feature selection', *Bioinformatics* **29**(18), 2365–2368.

Díaz-Uriarte, R. & De Andres, S. (2006), 'Gene selection and classification of microarray data using random forest', *BMC bioinformatics* **7**(1), 3.

Dietterichl, T. (2002), 'Ensemble learning', *The handbook of brain theory and neural networks* pp. 405–408.

Ding, C. & Peng, H. (2005), 'Minimum redundancy feature selection from microarray gene expression data', *Journal of Bioinformatics and Computational Biology* **3**(02), 185–205.

Dramiński, M., Rada-Iglesias, A., Enroth, S., Wadelius, C., Koronacki, J. & Komorowski, J. (2008), 'Monte carlo feature selection for supervised classification', *Bioinformatics* **24**(1), 110–117.

Dudoit, S. & Fridlyand, J. (2003), 'Bagging to improve the accuracy of a clustering procedure', *Bioinformatics* **19**(9), 1090–1099.

Fan, J., Feng, Y., Saldana, D. F., Samworth, R. & Wu, Y. (2014), *SIS: Sure Independence Screening*. R package version 0.7-2.
URL: *http://CRAN.R-project.org/package=SIS*

Fan, J. & Lv, J. (2008), 'Sure independence screening for ultrahigh dimensional feature space', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **70**(5), 849–911.

Fan, J., Samworth, R. & Wu, Y. (2009), 'Ultrahigh dimensional feature selection: beyond the linear model', *The Journal of Machine Learning Research* **10**, 2013–2038.

Fischer, B. & Buhmann, J. (2003), 'Bagging for path-based clustering', *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **25**(11), 1411–1415.

Freund, Y. & Schapire, R. (1997), A desicion-theoretic generalization of online learning and an application to boosting, *in* 'Journal of Computer and System Sciences', Vol. 55, pp. 119–139.

Friedman, J., Hastie, T. & Tibshirani, R. (2001), *The elements of statistical learning*, Vol. 1, Springer Series in Statistics.

Frossyniotis, D., Likas, A. & Stafylopatis, A. (2004), 'A clustering method based on boosting', *Pattern Recognition Letters* **25**(6), 641–654.

Ghosh, A. K., Chaudhuri, P. & Murthy, C. (2005), 'On visualization and aggregation of nearest neighbor classifiers', *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **27**(10), 1592–1602.

Giacinto, G. & Roli, F. (2001), 'Design of effective neural network ensembles for image classification purposes', *Image and Vision Computing* **19**(9), 699–707.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A. et al. (1999*a*), 'Molecular classification of cancer: class discovery and class prediction by gene expression monitoring', *science* **286**(5439), 531–537.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A. et al. (1999*b*), 'Molecular classification of cancer: class discovery and class prediction by gene expression monitoring', *Science* **286**(5439), 531–537.

Gordon, G. J., Jensen, R. V., Hsiao, L.-L., Gullans, S. R., Blumenstock, J. E., Ramaswamy, S., Richards, W. G., Sugarbaker, D. J. & Bueno, R. (2002), 'Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma', *Cancer Research* **62**(17), 4963–4967.

Hansen, L. & Salamon, P. (1990), 'Neural network ensembles', *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **12**(10), 993–1001.

Hastie, T. J., Tibshirani, R. & Friedman, J. (2009), *The elements of statistical learning: data mining, inference, and prediction*, Springer.

Ho, T., Hull, J. & Srihari, S. (1994), 'Decision combination in multiple classifier systems', *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **16**(1), 66–75.

Jirapech-Umpai, T. & Aitken, S. (2005), 'Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes', *BMC Bioinformatics* **6**(1), 148.

Jorissen, R. N., Gibbs, P., Christie, M., Prakash, S., Lipton, L., Desai, J., Kerr, D., Aaltonen, L. A., Arango, D., Kruhøffer, M. et al. (2009), 'Metastasis-associated gene expression changes predict poor outcomes in patients with dukes stage b and c colorectal cancer', *Clinical Cancer Research* **15**(24), 7642–7651.

Kikuchi, A., Ishikawa, T., Mogushi, K., Ishiguro, M., Iida, S., Mizushima, H., Uetake, H., Tanaka, H. & Sugihara, K. (2013), 'Identification of nucks1 as a colorectal cancer prognostic marker through integrated expression and copy number analysis', *International Journal of Cancer* **132**(10), 2295–2302.

Kim, J. (2009), 'Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap', *Computational Statistics & Data Analysis* **53**(11), 3735–3745.

Kohavi, R., Wolpert, D. et al. (1996), Bias plus variance decomposition for zero-one loss functions, *in* 'MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-', MORGAN KAUFMANN PUBLISHERS, INC., pp. 275–283.

Laiho, P., Kokko, A., Vanharanta, S., Salovaara, R., Sammalkorpi, H., Järvinen, H., Mecklin, J., Karttunen, T., Tuppurainen, K., Davalos, V. et al. (2007), 'Serrated carcinomas form a subclass of colorectal cancer with distinct molecular basis', *Oncogene* **26**(2), 312–320.

Lausen, B., Hothorn, T., Bretz, F. & Schumacher, M. (2004), 'Assessment of optimal selected prognostic factors', *Biometrical Journal* **46**(3), 364–374.

Lausser, L., Müssel, C., Maucher, M. & Kestler, H. A. (2013), 'Measuring and visualizing the stability of biomarker selection techniques', *Computational Statistics* **28**(1), 51–65.

Liaw, A. & Wiener, M. (2002), 'Classification and regression by randomforest', *R News* **2**(3), 18–22.
URL: *http://CRAN.R-project.org/doc/Rnews/*

Liu, H.-C., Peng, P.-C., Hsieh, T.-C., Yeh, T.-C., Lin, C.-J., Chen, C.-Y., Hou, J.-Y., Shih, L.-Y. & Liang, D.-C. (2013), 'Comparison of feature selection methods for cross-laboratory microarray analysis.', *IEEE/ACM Transactions on Computational Biology and Bioinformatics/IEEE, ACM* .

Liu, Y., Jin, R. & Jain, A. (2007), Boostcluster: Boosting clustering by pairwise constraints, *in* 'Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 450–459.

Loh, W. (2002), 'Regression trees with unbiased variable selection and interaction detection', *Statistica Sinica* **12**(2), 361–386.

Lu, J., Kerns, R. T., Peddada, S. D. & Bushel, P. R. (2011), 'Principal component analysis-based filtering improves detection for affymetrix gene expression arrays', *Nucleic Acids Research* **39**(13), e86–e86.

Ma, C., Dong, X., Li, R. & Liu, L. (2013), 'A computational study identifies hiv progression-related genes using mrmr and shortest path tracing', *PLOS ONE* **8**(11), e78057.

Mahmoud, O., Harrison, A., Gul, A., Khan, Z., Metodiev, M. & Lausen, B. (2015), Minimizing redundancy among genes selected based on the overlapping analysis, *in* 'Proceedings of the European Conference on Data Analysis', Bremen, Germany [ACCEPTED].

Mahmoud, O., Harrison, A., Perperoglou, A., Gul, A., Khan, Z. & Lausen, B. (2014b), *propOverlap: Feature (gene) selection based on the Proportional Overlapping Scores*. R package version 1.0.
URL: *http://CRAN.R-project.org/package=propOverlap*

Mahmoud, O., Harrison, A., Perperoglou, A., Gul, A., Khan, Z., Metodiev, M. & Lausen, B. (2014a), 'A feature selection method for classification within functional genomics experiments based on the proportional overlapping score', *BMC Bioinformatics* **15**(1).

Marczyk, M., Jaksik, R., Polanski, A. & Polanska, J. (2013), 'Adaptive filtering of microarray gene expression data based on gaussian mixture decomposition', *BMC Bioinformatics* **14**(1), 101.

McCall, M. N., Bolstad, B. M. & Irizarry, R. A. (2010), 'Frozen robust multiarray analysis (frma)', *Biostatistics* **11**(2), 242–253.

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A. & Leisch, F. (2014), *e1071: Misc Functions of the Department of Statistics (e1071), TU Wien*. R package version 1.6-3.
URL: *http://CRAN.R-project.org/package=e1071*

Michiels, S., Koscielny, S. & Hill, C. (2005), 'Prediction of cancer outcome with microarrays: a multiple random validation strategy', *The Lancet* **365**(9458), 488–492.

Nilsson, N. (1965), *Learning Machines: Foundations of Trainable Pattern-calssifying Systems*, McGraw-Hill.

Notterman, D. A., Alon, U., Sierk, A. J. & Levine, A. J. (2001), 'Transcriptional gene expression profiles of colorectal adenoma, adenocarcinoma, and normal tissue examined by oligonucleotide arrays', *Cancer Research* **61**(7), 3124–3130.

Olshen, L. B. J. F. R. & Stone, C. J. (1984), 'Classification and regression trees', *Wadsworth International Group* .

Oza, N. & Tumer, K. (2001), 'Input decimation ensembles: Decorrelation through dimensionality reduction', *Multiple Classifier Systems* pp. 238–247.

Partridge, D. & Krzanowski, W. (1997), 'Software diversity: practical statistics for its measurement and exploitation', *Information and software technology* **39**(10), 707–717.

Pavlovic, V. (2004), Model-based motion clustering using boosted mixture modeling, *in* 'Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on', Vol. 1, IEEE, pp. I–811.

Peng, H., Long, F. & Ding, C. (2005), 'Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy', *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **27**(8), 1226–1238.

Saeys, Y., Inza, I. & Larrañaga, P. (2007), 'A review of feature selection techniques in bioinformatics', *bioinformatics* **23**(19), 2507–2517.

Saffari, A. & Bischof, H. (2007), Clustering in a boosting framework, *in* 'Proc. of Computer Vision Winter Workshop (CVWW), St. Lambrecht, Austria', pp. 75–82.

Schapire, R., Freund, Y., Bartlett, P. & Lee, W. (1998), 'Boosting the margin: A new explanation for the effectiveness of voting methods', *The annals of statistics* **26**(5), 1651–1686.

Skalak, D. et al. (1996), The sources of increased accuracy for two proposed boosting algorithms, *in* 'Proc. American Association for Artificial Intelligence, AAAI-96, Integrating Multiple Learned Models Workshop', Vol. 1129, p. 1133.

Statnikov, A., Aliferis, C. F., Tsamardinos, I., Hardin, D. & Levy, S. (2005), 'A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis', *Bioinformatics* **21**(5), 631–643.

Su, X., Wang, M. & Fan, J. (2004), 'Maximum likelihood regression trees', *Journal of Computational and Graphical Statistics* **13**(3), 586–598.

Su, Y., Murali, T., Pavlovic, V., Schaffer, M. & Kasif, S. (2003), 'Rankgene: identification of diagnostic genes based on expression data', *Bioinformatics* **19**(12), 1578–1579.

Talloen, W., Clevert, D.-A., Hochreiter, S., Amaratunga, D., Bijnens, L., Kass, S. & Göhlmann, H. W. (2007), 'I/ni-calls for the exclusion of non-informative genes: a highly effective filtering tool for microarray data', *Bioinformatics* **23**(21), 2897–2902.

Tan, P., Steinbach, M. & Kumar, V. (2007), *Introduction to data mining*, Pearson Education India.

Tibshirani, R. (1996), 'Regression shrinkage and selection via the lasso', *Journal of the Royal Statistical Society. Series B (Methodological)* pp. 267–288.

Tumer, K. & Oza, N. (2003), 'Input decimated ensembles', *Pattern Analysis & Applications* **6**(1), 65–77.

Tusher, V. G., Tibshirani, R. & Chu, G. (2001), 'Significance analysis of microarrays applied to the ionizing radiation response', *Proceedings of the National Academy of Sciences* **98**(9), 5116–5121.

Ultsch, A., Pallasch, C., Bergmann, E. & Christiansen, H. (2010), A comparison of algorithms to find differentially expressed genes in microarray data, *in* A. Fink, B. Lausen, W. Seidel & A. Ultsch, eds, 'Advances in Data Analysis, Data Handling and Business Intelligence', Studies in Classification, Data Analysis, and Knowledge Organization, Springer, Berlin Heidelberg, pp. 685–697.

Vapnik, V. N. & Vapnik, V. (1998), *Statistical learning theory*, Vol. 2, Wiley New York.

Venables, W. N. & Ripley, B. D. (2002), *Modern Applied Statistics with S*, fourth edn, Springer, New York. ISBN 0-387-95457-0.
URL: *http://www.stats.ox.ac.uk/pub/MASS4*

Yu, L. & Liu, H. (2004), 'Efficient feature selection via analysis of relevance and redundancy', *The Journal of Machine Learning Research* **5**, 1205–1224.

Zou, C., Gong, J., Li, H. et al. (2013), 'An improved sequence based prediction protocol for dna-binding proteins using svm and comprehensive feature analysis.', *BMC Bioinformatics* **14**, 90.