2014-09-29

# Feature selection for cancer classification using microarray gene expression data

Zhong, Wenyan

UNIVERSITY OF CALGARY

# Feature selection for cancer classification using microarray gene expression data

by

Wenyan Zhong

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF MATHEMATICS AND STATISTICS

CALGARY, ALBERTA

September, 2014

**Abstract**

The rapid development of DNA microarray technology enables researchers to measure the expression levels of thousands of genes simultaneously and allows biologists easily gain insight into the complex interaction in tumours on gene expression levels. Its application in cancer studies has been shown great success in both diagnosis and elucidating the pathological mechanism. However, DNA microarray data usually contains thousands of genes and most of them are proved to be uninformative and redundant. Meanwhile, small size of samples of microarray data undermines the diagnosis accuracy of statistical models. Thereby, selecting highly discriminative genes from raw gene expression data can improve the performance of cancer classification and cut down the cost of medical diagnosis. This M.Sc. thesis proposes and investigates a new method of selecting highly discriminative genes for cancer classification based on DNA microarray data. For two-group classification problem, the Bhattacharyya distance is proposed to measure the dissimilarity in gene expression levels between the two groups. For any particular gene, we calculate the Bhattacharrya distance between the two groups based on the expression levels of that particular gene. We use the calculated distances, one for each gene, as a criteria to rank all the genes. Finally, support vector machine is utilized to obtain the optimal subset of genes achieving the lowest misclassification rate. Compared with the other two methods, SWKC (supervised weighted kernel clustering) (Shim et al., 2009) and SVM-RFE (support vector machine with recursive feature elimination) (Guyon et al., 2002), the proposed method is shown to be more effective and sensitive to differentially expressed genes. In the simulation study, the proposed method has much higher recovery rate than the other two methods. Comparisons among these three gene selection methods are also made through two real DNA microarray datasets, the colon dataset and the leukemia dataset, that are publicly available. Based on three classification performance indexes, i.e. average number of genes selected, average number of classification errors in test set and misclassification rate, the proposed method gets slightly better clas-

sification results than SVM-RFE for the colon dataset while at a much less computation cost. It also achieves better classification results than the SWKC methods in both datasets. Finally, we discuss that in future work improvement in performance could be achieved by introducing kernel density estimators and replacing Bhattacharyya distance with Hellinger distance as a feature selection criteria. Since kernel density estimation is free of distribution assumptions, under which the classification results would be more robust than that obtained by the Bhattacharyya distance under normal assumption.

**Acknowledgements**

The research would not have been accomplished without the valuable assistance and support from those who have contributed to the preparation and completion of this thesis.

In the first place, I would like to take this opportunity to express my sincere gratitude to my supervisor Dr. Jingjing Wu for her enthusiastic guidance, continuous support and kind encouragement throughout the whole period of my M.Sc program and study at University of Calgary. In the preparation stage, she spent plenty time discussing referenced papers with me in order to help me acquire a comprehensive and solid understanding of this research. As my research progressing, Dr. Wu provided me with many valuable suggestions and inspirations for this thesis. Without her patience, support and careful reading of this thesis, I would not have been able to make it in condition.

I would like to express my great thanks to my co-supervisor Dr. Xuewen Lu for his persistent supervision, technical discussions and valuable suggestions during the past two years.

I would also like to give my great appreciation to other members of my committee, Dr. Gemai Chen and Dr. Tak S Fung for serving as members of my committee and providing valuable advice and insightful comments for my thesis.

Many thanks go to the Department of Mathematics and Statistics for providing me a great environment to study and do research. I have learnt so much from seminars and research presentations that were held in the department and also from the valuable TA opportunity provided by the department. In addition, generous helps on study and research are provided by professors and staff in the department. Especially, the facility of Sunray server system allows me to do large scale simulations.

Special thanks should also be expressed to all my friends for their care, support and encouragement to me. We shared our memorable moments together in the past two years.

Last but not the least, I am forever indebted to my parents for their kind understanding,

endless patience and persistent encouragement when they were most desired. I am always grateful to them for their support.

# Table of Contents

# List of Tables

# List of Figures and Illustrations

# List of Symbols, Abbreviations and Nomenclature

| Symbol | Definition |
|---|---|
| U of C | University of Calgary |
| B | Bhattacharyya Distance |
| DNA | Deoxyribonucleic Acid |
| KDE | Kernel Density Estimator |
| MSE | Mean Squared Error |
| SKWC | Supervised Weighted Kernel Clustering |
| SNR | Signal-to-noise ratio |
| SVM | Support Vector Machine |
| SVM-RFE | Support Vector Machine with Recursive Feature Elimination |

# Chapter 1

# INTRODUCTION TO DNA MICROARRAY DATA

This chapter gives some introductory knowledge and background about microarray data. In Section 1.1, we first introduce what microarray data is and how it is obtained. In Section 1.2, two benchmark datasets are presented, i.e. the colon data and the leukemia data, which will be analyzed and discussed throughout this thesis.

## 1.1 Introduction to DNA Microarray

Over the last decades, gene expression has been receiving increasing attention by scientists due to the development and advances in DNA microarray technology. And the DNA microarray technology allows us to measure the expression levels of a large number of genes and to genotype multiple regions of a genome simultaneously. Meanwhile, the gene expression levels are believed to have a profound effect on how human body behaves. Therefore, DNA microarray techniques that have been applied in genome-wide gene expression and genome mutation analysis shed light for scientists on the understanding of pathophysiological mechanisms and help physicians in diagnoses, prognoses and choosing treatment plans (Musa et al., 2006).

A DNA microarray, also commonly known as DNA chip or biochip, is a collection of microscopic DNA spots attached to a solid surface. Each DNA spot contains picomoles ($10^{-12}$ moles) of a specific DNA sequence, known as probes (or reporters or oligos). These can be a short section of a gene or other DNA element that are used to hybridize a cDNA or cRNA sample called target under high-stringency conditions. Probe-target hybridization is usually detected and quantified by detection of fluorophore-, silver-, or chemiluminescence-labelled targets to determine relative abundance of nucleic acid sequences in the target.

Among all the methods for massively parallel measurements of gene expression data, two of them have become widely accepted and used. The first one is known as the spotted cDNA microarray technology, which is pioneered at the Stanford University. This technology is open to the public and is widely used for academic purpose. This technology involves robotic spotting of aliquots of purified cDNA clones, polymerase chain reaction products from clones or oligonucleotides onto glass slides that can contain thousands of arrayed elements (Shalon et al., 1996). The second approach is known as oligonucleotide microarray which is a patented commercial product developed by Affymetrix, Inc (www.affymetrix.com). It employs photolithography, the technology used in the manufacturing of computer chips, for embedding DNA probes on silicon chips.

The procedure for obtaining microarray data is as follows: RNA from experimental cells is removed at different sequential time points and then is reversely transcribed in the presence of fluorescent dye Cy5. A reference sample, such as one taken at time 0, is also reversely transcribed but in the presence of Cy3 and is subsequently mixed with experimental samples containing dye Cy5. Following hybridization of these samples to the DNA microarray, the Cy5/Cy3 ratio of each spot is measured, with the ratio being expressed as a log odds ratio in the base 2. The color red denotes over-expression of a gene relative to the control state, green signifies under-expression, and black indicates no change in which case the log ratio is 0 as the Cy5/Cy3 ratio is 1 (equal absolute expression during experimental and control states).

Although the working philosophies of the two technologies are different, according to current literatures, there is no significant difference between them in terms of their performance. In this thesis, the gene expression data we used here was generated by *Affymetrix*.

## 1.2 Description of the Datasets

In this section, we introduce two datasets that we will analyze in later chapters using the proposed method. Both datasets consist of a matrix of gene expression vectors obtained from DNA microarray of patients. And both datasets include thousands of variables but with small sample sizes. The first one was obtained from normal versus cancerous colon tissues. The second dataset was obtained from leukemia patients with two different types of leukemia.

### 1.2.1 Colon cancer dataset

Colon tumor is a disease in which cancerous growths are found in the tissues of colon. This colon cancer dataset was first presented in Alon et al. (1999). It contains 62 colon tissues samples with 2000 genes: 40 colon cancer tissues (labelled as "negative") and 22 normal tissues (labelled as "positive"). The original data is available online at *http://microarray.princeton.edu/oncology/affydata/index.html*. After pre-processing, it is a (62 tissues) $\times$ (2000 genes) data matrix that is ready for use in R, along with a vector of class index for each tissue.

With this dataset, the task here is to classify tissues as either cancerous or normal. Alon et al. (1999) provided a method of unsupervised learning to do the analysis, that is hierarchical clustering on 2000 genes. They indicated that most cancerous samples cluster together and most normal samples cluster together. One year later, Ben-Dor et al. (2000) presented the classification of colon cancer tissues and described how to select marker genes. Since then, this dataset has been widely used and analyzed by an increasing number of statisticians, which makes it one of the benchmark datasets in the area of gene expression data analysis. The same is for the leukemia dataset that will be presented below.

### 1.2.2 Leukemia dataset

Leukemias are primary disorders of bone marrow. They are malignant neoplasms of hematopoietic stem cells. In general, leukemia can be mainly classified into four types: acute lymphoblastic leukemia (ALL), acute myelogenous leukemia (AML), chronic lymphocytic leukemia (CLL) and chronic myelogenous leukemia (CML). The acute leukemia data was first published in Golub et al. (1999). In this dataset, the total number of genes tested is 7129 and the number of patients tested is 72. Those 72 are all acute leukemia patients, either ALL or AML. In more detail, this dataset contains two subsets: a training set and an independent test set. In the training set, there are 38 acute leukemia patients among which 27 are ALL patients and 11 are AML patients. The independent test set includes 34 acute leukemia patients with 20 ALL and 14 AML.

The leukemia data we used here is slightly different from that presented and studied in Golub et al. (1999). In Dettling (2004), this data has been pre-feature selected and standardized through an logarithm (base 10) transformation. As a result, there were actually 3571 genes after data preprocessing that were analyzed in Dettling (2004). Following Dettling (2004), the dataset used here consists of a data matrix for 3571 gene expression values on 72 individuals (3571 by 72) along with a vector of class index (72 by 1).

Note that a number of references have worked on this leukemia data after logrithm transformation, centering and standardization in various ways and at different stages of analysis.

The rest of this thesis is organized as follows. In Chapter 2, we provide a review of various approaches available in literature for microarray data analysis and some related topics to this thesis. In Chapter 3, we first propose to use Bhattacharyya distance to measure the dissimilarity in gene expression levels between two different classes. Then we aim to construct a classification model with minimum classification error. Finally we present the other two available feature selection methods for comparison. In Chapter 4, with the use of

Bhattacharyya distance on both the simulated data and the two benchmark real microarray datasets, we construct SVM (support vector machine) classifiers to select significant marker genes that discriminate between case group and control group. Based on the selected marker genes, we compare the classification results with the other two feature selection methods, SWKC (supervised weighted kernel clustering) and SVM-RFE (SVM with recursive feature elimination). Chapter 5 is devoted to a summary and conclusion of the comparison among the feature selection methods for both the simulated dataset and real datasets. Chapter 5 also presents a discussion on future work to generalize the proposed feature selection method by introducing kernel density estimators.

# Chapter 2

# REVIEW OF MICROARRAY DATA ANALYSIS TECHNIQUES

In this chapter, we give a literature review of microarray data analysis techniques. Section 2.1 gives a brief introduction to microarray data analysis and its two types of problems. Section 2.2 is devoted to the first type of problem, i.e. feature selection, while Section 2.3 concerns the second type of problem, i.e. molecular classification. Particularly, Section 2.2 presents an overview of various approaches that are currently available for feature selection in classification of heterogeneous diseases. Section 2.3 reviews several methods of classification, including the SVM and the SVM-RFE algorithms that will be studied in this thesis.

## 2.1 Introduction

The rapid development of microarray technology has given rise to a wealth of statistical studies that aimed at detecting significantly differentially expressed genes. DNA microarray data has been extensively studied. Currently, there already exist a variety of data analysis methods for DNA microarray to deal with different research of interest. In general, the data gathered from microarrays broadly prompts two types of questions: (1) those about variables themselves, such as, which genes or subsets of genes are associated with a specific phenotype, biological mechanism, or outcome; and (2) those regarding biological samples, such as, what prediction can be made about a specific tissue (Olshen et al., 2002; Clarke et al., 2008). Most statistical methods, including clustering, easily address the first question. Pattern classifiers, such as SVMs and other machine learning systems, however, are much better for the second question. It is vital for scientists to use the most appropriate analysis techniques in order

to extract the maximum amount of information from the obtained samples. In this thesis, we mainly focus on gene selection for cancer classification. Here, gene selection, or generally speaking feature selection, belongs to the first type of questions, while classification falls into the second type.

As a pre-processing step for machine learning, feature selection is a process of selecting a subset of original features so that the feature space is optimally reduced according to a certain evaluation criterion. Feature selection has been proven to be an effective way for removing redundant and irrelevant features, increasing efficiency in learning tasks, improving learning performance like predictive accuracy, and enhancing comprehensibility of learned results (Dash and Liu, 1997; Kohavi and John, 1997). In statistics, feature selection, also known as variable selection, is the process of selecting a subset of relevant variables for constructing statistical models. Feature selection techniques are used under the central assumption that the data contains many redundant or irrelevant features. Redundant features are those which provide no more information than the currently selected features, and irrelevant features provide no useful information in any context. Feature selection techniques are a subset of the more general field of feature extraction. Feature extraction creates new features from functions of the original features, whereas feature selection returns a subset of the features. Feature selection techniques are often used in domains where there are many features and comparatively few samples such as DNA microarray data. It is also useful as part of the data analysis process since it shows which features are important for prediction and how these features are related. In next section, we give a detailed review of feature selection techniques.

## 2.2 Feature Selection

Due to the high-dimension nature of microarray data and their small sample sizes, microarray data pose a great challenge to computational techniques. In order to tackle the difficulties in

analyzing microarray data, the apparent need of feature selection techniques was realized by researchers; see, e.g. Alon et al. (1999), Golub et al. (1999), Ross et al. (2000) and Ben-Dor et al. (2000) among many others. This has led to a recent surge of dimension reduction approaches presented in the areas of both bioinformatics and statistics. The literature on dimension reduction has already abounded.

Compared to other dimension reduction approaches, such as those based on compression (e.g. using information theory) or projection (e.g. principal component analysis), feature selection does not change the original representation of features (also called variables), but exclusively select a subset of them. Therefore, the original semantics of features have been preserved by feature selection techniques (Saeys et al., 2007). According to the literature, it is wildly believed that in most microarray gene expression data, only some relevant genes play an important role in classification and the rest of genes are irrelevant to classification. Thus, feature selection techniques are needed in order to find out those most important ones among all the genes that have been measured. With microarray data, the selection of important genes for classification of different phenotypes, such as cancer types, aims at providing a better understanding of the underlying biological system and improving the prediction performance of classifiers (Ramaswamy et al., 2001; Tibshirani et al., 2002; Guyon et al., 2002; Liu et al., 2005).

The objectives of feature selection are manifold. Those involved in this thesis are to improve prediction performance of classifier, to avoid over-fitting and to provide more computationally effective models. For classification problems, feature selection techniques could be classified into three types, i.e. filter methods, wrapper methods and embedded methods, depending on how they combine the feature selection with the construction of classification model.

### 2.2.1 Filter methods

Filter methods assess the relevance of features by looking only at the intrinsic properties of the data. In general, they perform feature selection in two steps. In the first step, based on a scoring criteria, an index measuring the relevance of feature is calculated. In the second step, the low-scoring features are excluded or features falling beyond some threshold criterion are eliminated. Afterwards, the subset of remaining features is used as the input to the classification algorithm. Scoring methods generally focus on measuring the differences between distributions of features. The resulting score is intended to reflect the quality of each feature in terms of its discriminative power. Many scoring criteria exist. For example, similar to Fisher's discriminant criterion, Pavlidis et al. (2001) used

$$V(i) = \frac{(\mu_+(i) - \mu_-(i))^2}{\sigma_+^2(i) + \sigma_-^2(i)}, \tag{2.1}$$

Where

$$\mu_+(i): \text{ mean of } i^{th} \text{ feature of class } (+)$$

$$\mu_-(i): \text{ mean of } i^{th} \text{ feature of class } (-)$$

$$\sigma_+(i): \text{ standard deviation of } i^{th} \text{ feature of class } (+)$$

$$\sigma_-(i): \text{standard deviation of } i^{th} \text{ feature of class } (-).$$

Here $V(i)$ is an index of the $i^{th}$ feature that is expressed in terms of the difference among empirical means of the two distributions and is normalized by the sum of their variances. Other popular filter methods, which are extensions of two-class methods, include SNR (signal-to-noise ratio) (Dudoit et al., 2002; Golub et al., 1999), Wilcoxon rank-sum rest (Thomas et al., 2001), Student's $t$-statistics (Dudoit et al., 2002; Liu et al., 2002), and BW (ratio of between-groups to within-groups sum of squares) (Dudoit et al., 2002).

A great advantage of filter methods is that they are independent of the classification algorithm. As a result, the feature selection procedure only needs to be performed once, and then it can be evaluated by different classifiers. Thereby they have a short computational

running time, and they often perform well in combinations with more robust classification methods such as the SVM. However, there are also some limitations associated with filter methods. Since the search in the feature subset space is independent of the search in the hypothesis space, filter methods ignore the interaction with the classifier. Another common disadvantage of filter methods is that most proposed methods are univariate. This means that each feature is considered individually and independently, which may lead to worse classification performance when compared to other types of feature selection techniques due to ignoring feature dependencies (Li et al., 2004; Statnikov et al., 2005). To overcome this downside, a number of multivariate filter methods were proposed.

### 2.2.2 Wrapper methods

Whereas filter methods select a good feature subset separately from the classification model selection step, wrapper methods search for the best feature subset in combination with a fixed classification method. This means wrapper methods embed the model hypothesis search within the feature subset search. Following this idea, a search procedure in the space of all possible feature combinations is defined and various subsets of features are generated and evaluated. However, as the number of all combinations is in exponential with the number of features, heuristic optimization frameworks have been applied to search for the best subset. These include: forward selection, backward elimination (Blum and Langley, 1997), hill climbing, beam search (Russel and Norvig, 1995), and randomized algorithms such as genetic algorithms (Koza, 1995). In general, these methods explore the space of all feature subsets (the search space) starting with no features, all features, or a random selection of features. A search algorithm is then "wrapped" around the classification model.

Compared with filter methods, wrapper methods have the advantage of interacting between feature subset search and model selection and have the ability to take into account feature dependency. A common drawback of wrapper methods is that they are very computationally intensive, due to the fact that many feature subsets need to be assessed. In addition,

they have a higher risk of over-fitting than filter methods, which generates non-reproducible gene subsets (Li et al., 2004).

Evaluating a feature subset in any wrapper method is done by internal validation methods, such as $k$-fold cross-validation or leave-one-out validation (Krus and Fuller, 1982). Since in our proposed method, cross-validation is used in the process of choosing optimal parameters to construct classification model, we give below a brief introduction to the concept of cross-validation.

Cross-validation is a model validation method for evaluating how the built statistical model will perform when generalized to an independent data set. It is often used in settings where the goal is to predict and/or estimate how accurately a predictive model will perform in practice. Suppose we have a model with some unknown parameters and one part of a dataset, the training set, is used to fit the model. The unknown model parameters are optimized in the fitting process to make the model fit the training data as well as possible. If we then take an independent sample of validation data from the same population as the training data, it will generally turn out that the model does not fit the validation data as well as it fits the training data. This phenomenon is called over-fitting and it is particularly likely to happen when the size of the training data is small or when the number of parameters in the model is large. Cross-validation is a way to predict the fit of a model to a hypothetical validation set when an explicit validation set is not available and to limit the problem of over-fitting.

One round of cross-validation involves partitioning a sample of data into two complementary subsets, performing the analysis on one subset (called the training set), and validating the analysis on the other subset (called the validation set or testing set). To reduce variability, multiple rounds of cross-validation are performed using different partitions and the validation results are averaged over the rounds. For example, in $k$-fold cross-validation, the original sample is randomly partitioned into $k$ equal size subsamples. Of the $k$ subsamples,

a single subsample is retained as the validation data for testing the model and the remaining $k-1$ subsamples are used as training data. The cross-validation process is then repeated $k$ times with each of the $k$ subsamples used exactly once as the validation data. The $k$ results from the folds can then be averaged to produce a single estimation. The advantage of this method over repeated random sub-sampling is that all observations are used for both training and validation and each observation is used exactly once for validation. Here $k$ remains an unfixed parameter (Geisser, 1993). When $k = n$, where $n$ is the number of observations, the $k$-fold cross-validation is exactly the leave-one-out cross-validation.

### 2.2.3 Embedded methods

Similar to wrapper methods, embedded methods are specific to a fixed learning algorithm. They incorporate feature selection as part of the model building process. This means that the search for an optimal subset of features is built into the construction of classifier and it can be seen as a search in the combined space of feature subsets and hypotheses (Saeys et al., 2007). In this sense, embedded methods include the interaction with classification model selection and meanwhile they are far less computationally intensive than wrapper methods. Some classical embedded methods include CART (classification and regression tree) (Breiman et al., 1984), RLR (regularized logistic regression) (Hastie et al., 2001) and SVM (Burges, 1998). Each of these methods handles features differently and consequently leads to different classification accuracies.

Regularization or shrinkage methods (Hastie et al., 2001; Xing et al., 2001) provide an alternative way for classification with high-dimensional but small sample size data. These methods trim the space of features directly during classification and thus effectively eliminate the non-informative features for classification. Regularization can be incorporated either into the error criterion or directly into the model. Let $\mathbf{w}$ be the parameter vector defining a classification model (e.g., the weights of a logistic regression model) and $Error(\mathbf{w}; \mathbf{D})$ be an error function measuring the fit of a model to the data $\mathbf{D}$ (e.g., least-squares as likelihood-

based error). Then a regularized error function is then defined as

$$Error_{Reg}(\mathbf{w};\mathbf{D}) = Error(\mathbf{w};\mathbf{D}) + \lambda \parallel \mathbf{w} \parallel, \tag{2.2}$$

where $\lambda > 0$ is a regularization constant and $\parallel \cdot \parallel$ denotes either the $L_1$ or $L_2$ norm. Intuitively, the regularization term penalizes the model for non-zero weights so that the optimization of the new error function drives all unnecessary parameters to 0. Examples of regularization methods includes LASSO (least absolute shrinkage and selection operator) (Tibshirani, 1996), RF-RFE (random forest with RFE) (Uriarte et al., 2006) and RRF (regularized random forest) (Deng et al., 2011).

Regularization effects can also be applied to SVMs, which is one of the most popular classifier (Burges, 1998; Smola, 2002). SVM defines a linear decision boundary called hyperplane that separates subjects into two classes. The boundary maximizes the distance (also called margin) between the two sample groups. The effect of margin optimization is twofold: only a small subset of data points, called support vector, are important for the separation; the dimensions unnecessary for separation are penalized. Both aspects help to avoid over-fitting. As a result, SVM offers a robust classification framework that performs very well for data with a moderately large number of features and relatively small sample size (Hauskrecht et al., 2006). In Section 2.3, we will give a more detailed review of SVM and an embedded method based on SVM known as SVM-RFE (Guyon et al., 2002).

The advantage of embedded methods is that they select high-quality feature subsets for a particular classifier or a specific model. Supervised learning methods that incorporate aspects of regularization, e.g. regularized logistic regression or SVMs, can build very good prediction models even in presence of high-dimensional data. However, as applications and extensions of RFE, some methods are relatively computationally expensive.

Since in this thesis we study the feature selection for cancer classification, in next section we present a review of molecular classification as the other main part of analysing DNA microarray data after feature selection stage being finished.

## 2.3 Molecular Classification

### 2.3.1 Classification problems

In this thesis the classification problem under our consideration is limited to two classes. Without loss of generality, we use (+) and (-) to denote the two classes. The input, as a "pattern", is a vector of $p$ components, called variables or features, measured on each of $n$ subjects. The output is the observed class labels for all $n$ subjects. We use $F$ to denote the $p$-dimensional feature space. In our situation, the features are expression levels of thousands of different genes. Thus, a given dataset consist of $n$ vectors of features $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k, ...\mathbf{x}_n\}$ with corresponding known class labels $\{y_1, y_2, ...y_k, ...y_n\}$. Here $y_k \in \{-1, 1\}$ with $y_k = -1$ indicating $k^{th}$ subject belongs to class(-) and $y_k = 1$ class(+). The entry for the $k^{th}$ subject is then $\{\mathbf{x}_k, y_k\}$. We randomly select a certain proportion of all subjects to form the training data set. The training data is used to construct a classifier, more specifically, to build a scalar discriminant/decision function $D(\mathbf{x})$ of an input pattern $\mathbf{x}$. New patterns are classified according to the sign of the decision function as

$$D(\mathbf{x}) > 0 \Rightarrow \mathbf{x} \in \text{class}(+)$$

$$D(\mathbf{x}) < 0 \Rightarrow \mathbf{x} \in \text{class}(-)$$

$$D(\mathbf{x}) = 0 \text{ decision boundary.}$$

Decision functions that are simply weighted sums of the training patterns plus a bias are called linear discriminant functions (Duda, 1973; Guyon et al., 2002). In this case, the decision function can be written as

$$D(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b, \tag{2.3}$$

where $\mathbf{w}$ is the weight vector and $b$ is a bias value. A data set is called *linearly separable* if a linear discriminant function can separate it without error.

### 2.3.2 Classification approaches

The advence of gene expression microarrays makes it possible to take a genomewide approach for disease prognosis, diagnosis, and prediction of therapeutic responsiveness (Wang et al., 2008; Clarke et al., 2008). With machine learning algorithms applied in studying the molecular signature, new subtypes of disease are identified and new insights into biological mechanisms and diagnostic targets emerge (Olshen et al., 2002; Clarke et al., 2008). For example, there are plenty of literature that have demonstrated that global gene expression profiling of a certain kind of human diseases can provide molecular classifications that reveal distinctive disease subtypes not evident with traditional histopathological approaches (Golub et al., 1999; Ramaswamy et al., 2001; Shedden et al., 2003; Wang et al., 2006).

Classification methods can be generally summarized into two branches: supervised learning and unsupervised learning. Here, we will focus on the supervised learning where the class labels are known beforehand. While molecular classification falls neatly within supervised learning, high dimension and small size of microarray samples facilitate new developments in not only feature selection techniques but also classifier design (Wang et al., 2008). With supervised learning methods for gene expression data, various classifiers with promising performance have been constructed. These classifiers include kNN ($k$-nearest neighbor rule) (Golub et al., 1999), LDA (Fisher Linear Discriminant Analysis), weighted gene voting (Golub et al., 1999; Tibshirani et al., 2002), SVM (Ramaswamy et al., 2001), NBC (naive Bayes classifier) (Liu et al., 2002), linear regression (Fort and Lambert-Lacroix, 2005), artificial neural networks (Wang et al., 2006), CART (Breiman et al., 1984) and random forest (Breiman et al., 2001). Many comparative reviews and studies indicate that SVM-based classifiers outperform other methods on most benchmark microarray data sets (Li et al., 2004; Statnikov et al., 2005), but in general no one classifier uniformly outperforms others.

### 2.3.3 Support vector machines (SVM)

In this section, we introduce some basic concepts of SVM (Cortes and Vapnik, 1995). In machine learning, SVMs are supervised learning models with associated learning algorithms that analyze data and recognize patterns and they are used for classification and regression analysis. Based on a set of training samples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new samples into one category or the other and this makes it a non-probabilistic binary linear classifier. An SVM model is a representation of the samples, as points in space, mapped so that samples of separate categories are divided by a clear gap as wide as possible. New samples are then mapped into that same space and predicted to belong to a category according to which side of the gap they fall on. In brief, SVM is a binary classification method that discriminates a set of data points from another. It can be roughly sketched as follows: SVM constructs a hyperplane as the decision surface that perfectly separates each class samples with a maximum margin, where the margin is defined as the distance from the hyperplane to the nearest sample. The concept of SVM for classification is illustrated in Figure 2.1.

Unfortunately, for real life data, it is often difficult to clearly discriminate between positive and negative samples. SVM tackles this problem by mapping data points into a higher dimension space, called *feature space*, instead of into the *input space* where we find the training samples. The feature space is so named because we name each entry in the expression vector as a feature. Furthermore, algorithms determining the hyperplane in the feature space can be expressed exclusively in terms of vectors in the input space and dot products in the feature space. Consequently, the SVM, by defining a kernel function that assumes the role of the dot product in the feature space, can identify the hyperplane without ever having to actually represent the feature space. However, Eisen et al. (1998) mentioned that the dot product of two normalized vectors is the simplest way to measure the similarity in expression

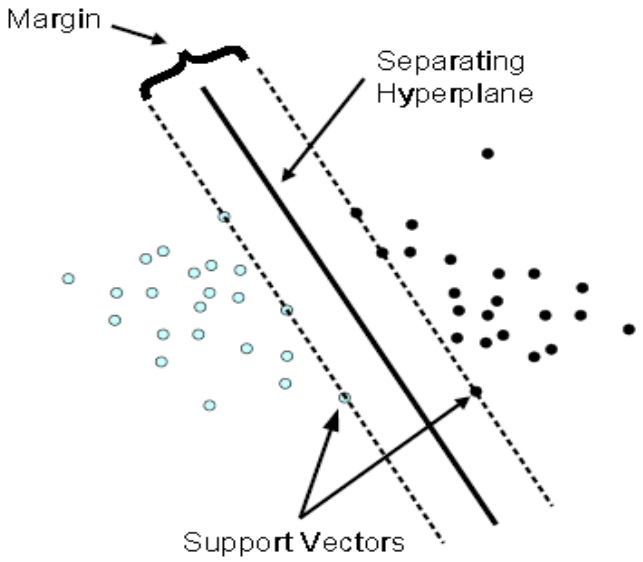


Figure 2.1: SVM for classification: linear separable case.

vectors between two samples, i.e.

$$K(\mathbf{X}, \mathbf{Y}) = \overrightarrow{X} \cdot \overrightarrow{Y} = \sum_{i=1}^{p} X_i Y_i.$$

By raising the kernel function to a higher power, say $d_f$, i.e. $\sum_{i=1}^{p}(X_i Y_i + 1)^{d_f}$, one obtains a hyperplane of higher degrees in the input space, and for each gene there now exist $d_f$-fold interactions between microarray measurements $(X_{i1}, X_{i2}, \ldots, X_{id_f})$ in this kernel's feature space. In case that a linear SVM is unable to create an effective separating hyperplane, we can implement a *soft margin* that allows some training samples to be misclassified. This is also a way to control the trade-off between false positives and false negatives.

There are mainly two types of SVM in practice for classification, namely L1-SVM and L2-SVM. Given a data set $\mathbf{D} = \{(\mathbf{x_i}, y_i) : i = 1, \ldots, n\}$, where sample $\mathbf{x_i}$ has binary class label $y_i$ such that $y_i \in \{-1, 1\}$. The SVM with $L_1$-norm of the slack variables, named as

L1-SVM, is the solution to the following constrained minimization:

$$\begin{gathered}\min_{\mathbf{w},\xi_1,\ldots,\xi_n}\left\{\frac{1}{2}\|\mathbf{w}\|^2+C\sum_{i=1}^{n}\xi_i\right\}\\ \text{subject to } y_i(w_i\cdot\Phi(x_i)+b)\geq 1-\xi_i \text{ and } \xi_i\geq 0, i=1,\ldots,n,\end{gathered}\tag{2.4}$$

where $\xi_i$ denotes the slack variables and $C$ is a regularization parameter adjusting the training errors and controlling over-fitting.

The dual form of above minimization is to solve the following quadratic optimization problem:

$$\begin{gathered}\max_{\alpha_1,\ldots,\alpha_n}\left\{\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K(\mathbf{x_i},\mathbf{x_j})\right\}\\ \text{subject to } 0\leq\alpha_i\leq C \text{ and } \sum_{i=1}^{n}\alpha_i y_i=0, i=1,\ldots,n,\end{gathered}\tag{2.5}$$

where $\{\alpha_i\}_{i=1}^{n}$ are the Lagrange multipliers and $K(\mathbf{x_i},\mathbf{x_j})$ is the inner product of $\Phi(\mathbf{x_i})$ and $\Phi(\mathbf{x_j})$ with $\Phi(\cdot)$ a nonlinear function that projects the input pattern into a linearly separable feature space. Function $\Phi$ is also viewed as a feature mapping function which maps the input space to the higher dimensional feature space. In practice, $K(\mathbf{x_i},\mathbf{x_j})$ is calculated through a kernel function instead. Some popular kernel functions are available in Table 2.1.

Table 2.1: Some commonly used kernel functions in SVM

| Name | Formula | Parameters |
|---|---|---|
| Linear | $K(\mathbf{u},\mathbf{v})=\mathbf{u}^T\mathbf{v}$ | (none) |
| Radial Basis Function | $K(\mathbf{u},\mathbf{v})=exp\left\{-\gamma\lvert\mathbf{u}-\mathbf{v}\rvert^2\right\}$ | $\gamma$ |
| Sigmoid | $K(\mathbf{u},\mathbf{v})=tanh\left\{\gamma\mathbf{u}^T\mathbf{v}+c_0\right\}$ | $\gamma, c_0$ |
| Polynomial | $K(\mathbf{u},\mathbf{v})=\gamma(\mathbf{u}^T\mathbf{v}+c_0)^d$ | $\gamma, c_0, d$ |

L2-SVM uses the $L_2$-norm of slack variables $\xi_i$ in the objective function. In another word,

L2-SVM is formulated as

$$\min_{\mathbf{w},\xi_1,\ldots,\xi_n} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{2}\sum_{i=1}^{n}\xi_i^2 \right\} \\ \text{subject to } y_i(w_i \cdot \Phi(x_i) + b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0, i = 1, \ldots, n. \tag{2.6}$$

By introducing Lagrange multipliers $\{\alpha_i\}_{i=1}^n$, one can obtain its dual form

$$\max_{\alpha_1,\ldots,\alpha_n} \left\{ \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \left[ K(\mathbf{x_i}, \mathbf{x_j}) + \frac{1}{C} \right] \right\} \\ \text{subject to } 0 \leq \alpha_i \leq C \text{ and } \sum_{i=1}^{n}\alpha_i y_i = 0, i = 1, \ldots, n. \tag{2.7}$$

Given a new test sample $\mathbf{x}$ to be classified, with either type of SVM, its label can be predicted according to the decision function

$$D(\mathbf{x}) = \text{sign}\left( \sum_{i=1}^{n}\alpha_i y_i K(\mathbf{x_i}, \mathbf{x}) + b \right), \tag{2.8}$$

where $b$ is a bias value.

### 2.3.4 SVM with recursive feature elimination (SVM-RFE)

RFE is an instance when backward feature elimination is used to remove features based on a criterion related to decision/discrimination function by an iterative procedure (Kohavi, 2000). SVM-RFE, an embedded feature selection technique, uses the weight magnitude as the ranking criterion. The SVM-RFE algorithm proposed by Guyon et al. (2002) returns a ranking of the features in a classification problem by training a SVM with a linear kernel and removing the features with smallest rankings. This ranking criterion is the $\mathbf{w}$ values of the decision hyperplane given by the SVM. Soon SVM-RFE, as an application of SVM, became a *de facto* standard in the field of molecular classification (Saeys et al, 2007).

SVM is re-trained at each step to eliminate redundant features and yield better feature subsets. It has four steps as follows.

1. Train an SVM with the training set;

2. Order features using the weights of the decision hyperplane given by the resulting SVM;

3. Eliminate features with the smallest weights;

4. Repeat the steps 1-3 with the training set restricted to the remaining features.

The detailed algorithm is given below.

*Input:*

training samples $\mathbf{X}_0 = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^T$

class labels $\mathbf{y} = [y_1, y_2, \ldots, y_n]^T$

*Initialize:*

subset of surviving features $\mathbf{s} = [1, 2, \ldots, p]$

ranked feature list $\mathbf{r} = [\ ]$ ( Set empty space for vector $\mathbf{r}$.)

*Repeat until* $\mathbf{s} = [\ ]$ ( The algorithm stop when vector $\mathbf{s}$ becomes an empty set.)

{

*Restrict* training samples to good feature indices: $\mathbf{X} = \mathbf{X}_0\,(:, \mathbf{s})$

*Train* the classifier to obtain the Lagrange multipliers $\alpha$: $\alpha =$ SVM-train$(\mathbf{X}, \mathbf{y})$

*Compute:*

weight vector of dimension $d =$length$(\mathbf{s})$: $\mathbf{w} = \sum_k \alpha_k y_k \mathbf{x}_k$

ranking criteria: $c_i = (w_i)^2$, $i = 1, \ldots, d$

*Find* the feature with smallest ranking criterion: $f = argmin(\mathbf{c})$

*Update* ranked feature list: $\mathbf{r} = [\mathbf{s}(f), \mathbf{r}]$ ( At this stage, we obtain the updated ranked feature list by adding the feature $\mathbf{s}(f)$ to the old ranked feature list which was generated in the last iteration. )

*Eliminate* the feature with smallest ranking criterion: $\mathbf{s} = \mathbf{s}(1 : f-1, f+1 : d)$

}

*Output:* ranked feature list $\mathbf{r}$

# Chapter 3

# METHODOLOGIES

The proposed method in this thesis follows a similar framework to that of filter method in feature selection for two-category classification problems. Particularly, it selects informative genes according to their discriminative power but without considering any knowledge of the classifier adopted. With use of DNA microarray data, we propose a new approach for marker gene selection in cancer classification with improved classification accuracy. In Section 3.1, we introduce the proposed method B/SVM of this thesis. Section 3.2 presents another feature selection method SWKC proposed by Shim et al. (2009). The SWKC together with the SVM-RFE method presented in Section 2.3.4 are used to compare with the proposed method in terms of classification performance. The comparison results will be presented in Chapter 4.

## 3.1 Bhattacharyya Distance with SVM Classifier (B/SVM)

### 3.1.1 Introduction

The DNA microarray technology allows us to measure the expression levels of thousands of genes simultaneously, providing great chance for cancer diagnosis and prognosis. In this thesis we study gene expression levels to determine the marker genes which are relevant to a given cancer. The number of genes often exceeds tens of thousands, whereas the number of subjects available is often no more than a hundred. Therefore, gene selection is not only needed but also important for classification. A good subset of discriminative genes can improve prediction accuracy of classifiers and save computational cost with reduced dimension of data. So far, the most popularly used gene selection methods are based on gene ranking. T-statistics, SNR, the Fisher's criterion, information gain and statistics, probability of se-

lection, RFE in its various forms are commonly used criteria for gene ranking. Considering the fact that many cancers or classification problems are controlled by only a small number of genes, it is valuable to find these informative genes.

In this chapter, we propose a new method for gene selection and tissue classification based on SVM. In order to accelerate the gene selection process, we firstly calculate, for each gene, the Bhattacharyya distance between two classes. Afterwards, we rank the genes according to their corresponding Bhattacharyya distances. Then we evaluate certain subsets, starting with the top-ranked gene with largest Bhattacharyya distance and progressively adding the next one on the list until all genes are included. At last, we choose, by forward selection method, the final gene subset based on the subsets' individual classification ability with SVMs. In order to improve the performance of SVM with the final gene subset, parameter optimization of SVM is conducted. Then the SVM is trained using the parameter optimized with the final optimal gene subset and is used to predict the testing data.

### 3.1.2 Bhattacharyya distance

In statistics, the Bhattacharyya distance is often used to determine the similarity of two probability distributions, discrete or continuous. In classification, it measures the separability of classes and is considered to be more robust and reliable than the Mahalanobis distance, as the latter is a special case of the former when the standard deviations of the two classes are the same. In cases when two classes have similar means but different standard deviations, the Mahalanobis distance would be close to zero whereas the Bhattacharyya distance would grow depending on the difference between the two standard deviations. The Bhattacharyya distance is closely related to the Bhattacharyya coefficient which is a measure of the amount of overlap between two statistical samples or populations (Bhattacharyya, 1943). This coefficient can be used to measure the relative closeness of two samples under consideration.

For discrete probability distributions $p$ and $q$ over the same domain $X$, the Bhattacharyya

Distance is defined as

$$D_B(p,q) = -\ln\left(BC(p,q)\right), \tag{3.1}$$

where $BC(p,q) = \sum_{x\in X}\sqrt{p(x)q(x)}$ is the Bhattacharyya coefficient. When $p$ and $q$ are continuous density functions, the Bhattacharyya coefficient is defined as

$$BC(p,q) = \int \sqrt{p(x)q(x)}\,dx. \tag{3.2}$$

In either case, $0 \le BC \le 1$ and $0 \le D_B \le \infty$. Particularly when the two populations $p$ and $q$ are normal, the Bhattacharyya distance can be calculated by extracting the means and variances of the two separate distributions or classes; specifically,

$$D_B(p,q) = \frac{1}{4}\ln\left[\frac{1}{4}\left(\frac{\sigma_p^2}{\sigma_q^2} + \frac{\sigma_q^2}{\sigma_p^2} + 2\right)\right] + \frac{1}{4}\left[\frac{(\mu_p - \mu_q)^2}{\sigma_p^2 + \sigma_q^2}\right], \tag{3.3}$$

where $\mu_p$ and $\sigma_p^2$ are the mean and variance of $p$ and $\mu_q$ and $\sigma_q^2$ are those of $q$.

Note that the Bhattacharyya distance does not obey the triangle inequality. However, there is a relationship

$$D_H(p,q) = \sqrt{2}\cdot\sqrt{1 - BC(p,q)} \tag{3.4}$$

between the Bhattacharyya distance and the Hellinger distance which does obey the triangle inequality, where $D_H(p,q)$ is the Hellinger distance between $p$ and $q$ defined as $D_H(p,q) = \{\int[\sqrt{p(x)} - \sqrt{q(x)}]^2 dx\}^{1/2}$ for continuous case and $D_H(p,q) = \{\sum_{i=1}^{k}(\sqrt{p_i} - \sqrt{q_i})^2\}^{1/2}$ for discrete case.

### 3.1.3 B/SVM algorithm for marker genes selection

The B/SVM algorithm that we propose for marker genes selection is based on the Bhattacharyya distance along with SVM. Within this algorithm, the Bhattacharyya distance is used to obtain the ranking list of genes for classification, where genes with larger Bhattacharyya distances are suggested to be more discriminative on the differentiation of two categories than those with smaller ones. Then SVM is used to evaluate each subset of genes in terms of classification performance. Eventually, we identify a subset of discriminative

genes that provides the lowest classification error rate as marker genes that will be used for the validation data. Denote $p_i$ and $q_i$ the distribution functions of the $i^{th}$ gene for the two classes, say class(+) and class(-), $i = 1, \ldots, p$. Then in general, the proposed B/SVM algorithm includes the following four steps:

1. Compute the Bhattacharyya distance $D_B(p_i, q_i)$ given in (3.3) for the $i^{th}$ gene, $i = 1, \ldots, p$;

2. Order genes according to the magnitude of their Bhattacharyya distances;

3. Evaluate subsets of important genes using SVM in terms of classification power, starting with the gene(s) with the largest Bhattacharyya distance:

   (a) Randomly divide the given data into training data and testing data;

   (b) Use the training data to train the SVM with optimal hyperparameters selected by 10-fold cross-validation criterion for any subset of important genes;

   (c) Obtain the classification error rate of the trained SVM applied to the testing data;

4. Repeat Step 3 progressively by adding the next important gene until all genes are added once. We identify as marker genes the subset of important genes that provides the lowest classification error rate for testing data.

The illustration chart of the proposed B/SVM method is given in Figure 3.1. In the first step, we assume for simplicity that for any fixed gene, the expression levels of subjects in the two classes are from two (possibly the same) normal distributions, i.e. we assume that both $p_i$ and $q_i$ are normal distributions, $i = 1, \ldots, p$. Based on the data, we can calculate the sample mean and variance for both $p_i$ and $q_i$. Then we plug in these sample means and variances into (3.3) to get the estimated Bhattacharyya distance for each gene. In order to evaluate and compare the effectiveness of this feature selection approach, the given dataset is divided into two, training set and testing set. The training set is used to train the model while the testing set is used as an independent validation set to evaluate the performance of

the model through prediction results.

## 3.2 Supervised Weighted Kernel Clustering (SWKC)

### 3.2.1 Clustering

As one of the most widely used unsupervised learning method, clustering groups data components, either data points or features, according to their similarity. Different from supervised learning methods that rely heavily on class label information, clustering is unsupervised and the information about the target classes is not used. Every data component is assigned to one of the clusters; components falling into the same cluster are assigned the same value in the new representation, that is, a new cluster label (Ben-Dor et al., 2000; Slonim et al., 2000).

Clustering algorithms are frequently applied to DNA microarray data. Defining appropriate metrics to measure the similarity in an input space is a very important component of clustering algorithms such as hierarchical and $K$-mean algorithms. Input vectors can be either expression profiles across different genes for clustering subjects with similar microarray data or expression profiles across different arrays for grouping genes with similar expression patterns across different subjects. Thus, clustering methods rely on the similarity matrix between data components. Similarity matrix can be built with use of one of the standard distance metrics, such as Euclidean distance, Mahalanobis distance, Minkowski distance or commonly used Pearson correlation, but more complex distances based on, for example, functional similarity of genes (Speer et al., 2005), are also possible. With these metrics, the hierarchical and $K$-mean algorithms can be implemented for clustering. Clusters are often presented as dendrograms or color-coded representation of similarly expressed genes.

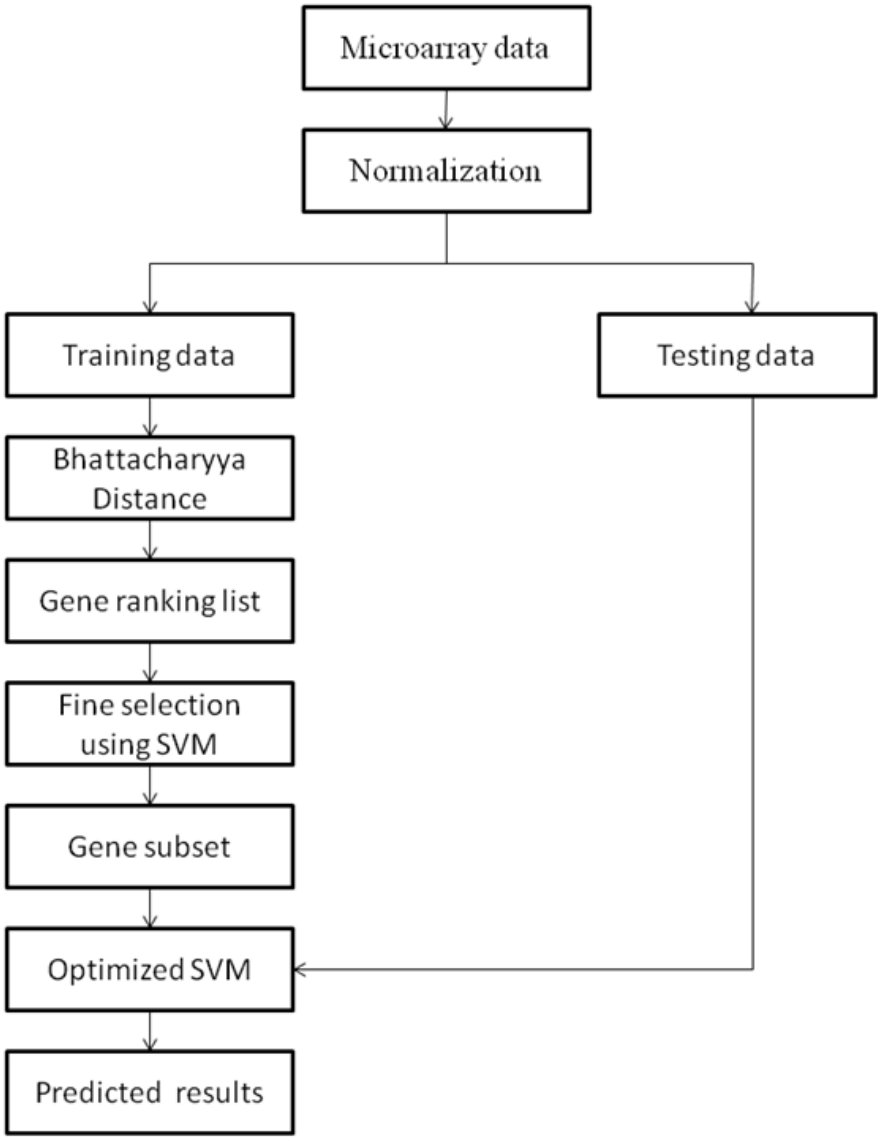


Figure 3.1: Schematic illustration of the proposed B/SVM method.

### 3.2.2 SWKC

The supervised weighted kernel clustering method, referred as SWKC, for marker gene selection is proposed by Shim et al. (2009). Different from canonical clustering methods that treat all the genes/variables equally important in clustering, SWKC deals with genes differently in clustering according to their relevance. Since genes playing a more important role in clustering are referred as marker genes in classification, SWKC can be utilized for marker genes selection. The detailed algorithm of SWKC is given as follows.

Consider an $m$-cluster dataset of $n$ subjects $\{(\mathbf{x}_j, \mathbf{u}_j)\}_{j=1}^{n}$. The input $\mathbf{x}_j = (x_{j1}, \ldots, x_{jp})^T$ is a $p \times 1$ column vector consisting of the expression levels of $p$ genes measured on the $j^{th}$ subject. The other input $\mathbf{u}_j = (u_{j1}, \ldots, u_{jm})$ is a $m \times 1$ column vector of hard membership index with $u_{ji} \in \{0, 1\}$ indicating whether the $j^{th}$ subject belongs to the $i^{th}$ cluster ($u_{ji} = 1$) or not ($u_{ji} = 0$). The $\mathbf{u}_j$ is of a more general form than $y_j$ (a scalar) used in previous chapters to denote class index, and thus it is a generalization of notation from two-class to $m$-class label index. Consider the objective function

$$L^* = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{p} u_{ji} w_{ik}^2 d^2(x_{jk}, v_{ik}) \tag{3.5}$$

subject to the constraint $\sum_{k=1}^{p} w_{ik} = 1$, $i = 1, \ldots, m$, where $w_{ik}$ is a weight factor associated with the $k^{th}$ gene in the $i^{th}$ cluster, $v_{ik}$ is the $k^{th}$ gene expression level of the $i^{th}$ cluster center, and $d^2(x_{jk}, v_{ik})$ is a squared distance between $x_{jk}$ and $v_{ik}$ whose summation with respect to $k$ measures the dissimilarity between the $j^{th}$ subject and the $i^{th}$ cluster center. The optimal weights $\{w_{ik} : i = 1, \ldots, m, k = 1, \ldots, p\}$ can be obtained by minimizing the constrained objective function (3.5). Considering the fact that the number of genes $p$ in microarray data may exceed several thousands, this optimization may suffer from the "curse of dimensionality" which was coined by Bellman R.E. in 1961, in another word, its performance deteriorates as the number of genes increases. To overcome this, Shim et al. (2009) proposed a more effective clustering scheme. Their idea is to cluster subjects in a

high dimensional feature space by introducing feature mapping function, where the inner product of these functions can be given as a specified form of kernel function. In Shim et al. (2009), the most widely used Gaussian kernel is utilized which satisfies Mercer (1909)'s conditions for kernels illustrated in Vapnik (1998). Then $d^2(x_{jk}, v_{ik})$ can be written using a kernel function $K$ as

$$d^2(x_{jk}, v_{ik}) = K(x_{jk}, x_{jk}) - 2K(x_{jk}, v_{ik}) + K(v_{ik}, v_{ik}) \tag{3.6}$$

To minimize the objective function $L^*$ given in (3.5) in terms of $w_{ik}$, we write the Lagrangian function as

$$L = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{p} u_{ji} w_{ik}^2 d^2(x_{jk}, v_{ik}) - \sum_{i=1}^{m}\eta_i(\sum_{k=1}^{p} w_{ik} - 1), \tag{3.7}$$

where $\eta_i$ is the Lagrange multiplier. To obtain the $w_{ik}$, we differentiate the objective function $L$ with respect to $w_{ik}$ and $\eta_i$, respectively, and set the equations to zero. As a result, the optimal $w_{ik}$ can be written in closed form as

$$w_{ik} = \left( \sum_{l=1}^{p} \frac{\sum_{j=1}^{n} u_{ji} d^2(x_{jk}, v_{ik})}{\sum_{j=1}^{n} u_{ji} d^2(x_{jl}, v_{il})} \right)^{-1}, i = 1, \ldots, m,\ k = 1, \ldots, p. \tag{3.8}$$

For a given $i^{th}$ cluster, we infer that genes attached to larger values of $w_{ik}$ play more important roles in clustering in the $i^{th}$ cluster. Therefore, genes attached to larger values of $\{w_{ik}\}_{i=1}^{m}$ are considered more important and thus $\{\sum_{i=1}^{m} w_{ik}\}_{k=1}^{p}$ can be used as the feature ranking or gene selection criterion (Shim et al., 2009).

# Chapter 4

# SIMULATION STUDIES AND REAL DATA APPLICATIONS

In this chapter, we illustrate how well B/SVM performs in marker gene selection through both simulation studies and real data applications. Specifically, Section 4.1 presents the results of simulation studies for both the proposed B/SVM and the well known SWKC/SVM and SVM-RFE. In Section 4.2 we applied and compared the three methods for the two benchmark datasets, i.e. the colon cancer dataset and leukemia dataset. Both simulation studies and real data applications suggest that the proposed B/SVM performs competitively in the sense that it produces relatively lower misclassification rate compared with SWKC/SVM and SVM-RFE.

## 4.1 Simulation Studies

To evaluate how well the proposed B/SVM method perform in feature selection for classification, a simulation study is carried out. As in Broberg (2002) and Koo et al. (2006), the simulated data are generated from normal distributions assumed as the populations of gene expression levels after log transformation.

### 4.1.1 Simulated data

The means and standard deviations of normal distributions for the simulated data are given in Table 4.1. In this table, the first three rows, designated as the null cases, represent the distributions of irrelevant genes. The last three rows in the table, designated as the significant cases, represent the distributions of marker genes for the two groups of subjects. When generating observations from either an irrelevant gene or a marker gene, the popula-

tion distribution of the gene is randomly chosen from either the first three rows or the last three rows depending on whether it's an irrelevant gene or a marker gene. The numbers of subjects, in another word the sample sizes, in the normal group and the case group are chosen to be 47 and 25, respectively, as in Broberg (2002) and Koo et al. (2006). These particularly chosen sample sizes are intended to match those of the real leukemia data introduced in Chapter 1. In the simulation, we consider 1000 genes in total among which 1%, i.e. 10 genes, are assumed differentially expressed. As a result, the simulated data for a normal subject consists of 1000 genes randomly selected from the null cases, while the simulated data for a case subject contains 10 differentially expressed genes randomly selected from the significant cases and the rest 990 genes are all from the null cases. Note that the data are generated gene by gene across all 72 subjects, rather than subject by subject with all 1000 genes.

Table 4.1: Normal distributions used to generate simulated data

| Genes expression levels | Mean 1 | Standard deviation 1 | Mean 2 | Standard deviation 2 |
|---|---|---|---|---|
| Null cases | -8.0 | 0.2 | -8.0 | 0.2 |
| | -10.0 | 0.4 | -10.0 | 0.4 |
| | -12.0 | 1.0 | -12.0 | 1.0 |
| Significant cases | -6.0 | 0.1 | -6.1 | 0.1 |
| | -8.0 | 0.2 | -8.5 | 0.2 |
| | -10.0 | 0.4 | -11.0 | 0.7 |

### 4.1.2 Simulation results

The simulated data is randomly split into a training set of 67% and a testing set of 33% of the whole dataset. Based on the training set, the Bhattacharyya distance is calculated for each gene according to which all the genes are ranked in decreasing order. Then the SVM classifier

is built based on the gene rankings and classification error rate is calculated for the testing set. Here, the SVM is used as a method of marker gene selection. The simulation procedure is repeated 50 times. In other words, 50 independent datasets are generated. For each generated dataset, we also randomly split it 50 times into the training set and testing set. Thus, for each generated dataset the indexes are averaged over these 50 splittings. Afterwards, we increase the number of simulations from 50 to 200. We compare our proposed B/SVM method with the well known SWKC/SVM (Shim et al., 2009) and SVM-RFE (Guyon et al., 2002). For comparison purpose, we mainly focus on the following indexes: average number of genes selected, average number of true marker genes selected, average recovery rate, and average misclassification rate of classification models based on the constructed classifier. Table 4.2 presents the summarized results for the simulated data under 50 and 200 simulations.

Table 4.2: Comparison of the proposed B/SVM with SWKC and SVM-RFE using simulated data

| Simulations | Indexes | B/SVM | SWKC/SVM | SVM-RFE |
|---|---|---|---|---|
| 50 | Average number of genes selected | 6.995 | 8.548 | **3.512** |
| | Average number of true marker genes selected | **6.008** | 2.800 | 2.970 |
| | Average recovery rate (%) | **95.744** | 83.884 | 94.057 |
| | Average misclassification rate (%) | **1.055** | 7.308 | 1.960 |
| 200 | Average number of genes selected | 6.613 | 8.000 | **3.466** |
| | Average number of true marker genes selected | **5.834** | 2.789 | 2.920 |
| | Average recovery rate (%) | **96.561** | 83.946 | 94.235 |
| | Average misclassification rate (%) | **1.090** | 7.703 | 2.034 |

In Table 4.2, "Average number of genes selected" denotes the averaged number of genes selected out of 1000 over 50 generated datasets; "Average number of true marker genes selected" denotes the averaged number of those, among all the selected genes, that are desig-

nated as significant cases; "Average recovery rate" denotes the averaged ratio of the number of true marker genes selected to the total number of genes selected; "Average misclassification rate" denotes the averaged proportion of misclassified subjects calculated as the ratio of the number of prediction errors to the size of testing set.

Take the results for 50 simulations as an example for the following discussion and similar observations are true for the 200 simulations. Compared with SVM-RFE and SWKC/SVM, the proposed B/SVM returns a relatively higher average recovery rate of 95.7%. This indicates that B/SVM has higher power of detecting marker genes than the other two methods. B/SVM also gives the lowest misclassification rate among the three methods. SVM-RFE selects the smallest size, 3.512 on average, of gene subset among the three, at the price of expensive computation time. B/SVM as a filter method takes much less computing time than both SVM-RFE and SWKC/SVM. Specifically, in the simulation study, with 50 times replications, B/SVM takes 31 mins 48 secs and SWKC/SVM takes 49 mins 52 secs while SVM-RFE takes 13 hrs 29 mins 24 secs. Note that the smallest subset size doesn't mean the best since we know this simulation is designed with 10 truly differentially expressed genes. SVM-RFE can only identify 2.97 of the 10 while B/SVM identify 6.008 on average. In this sense, SWKC/SVM performs the worst, selecting the largest size of gene subset but identifying the fewest true marker genes. These observations suggest that the proposed B/SVM method outperforms both SWKC/SVM and SVM-RFE.

The numerical studies given in this thesis are conducted with the use of R programming. The R package, *e*1071, has provisions for performing C-classification which corresponds to a soft-margin classifier (C here is the same as that in (2.4)) using a Gaussian kernel or linear kernel. In this simulation study, 10-fold cross-validation on each training set was used to estimate the cost parameter C in SVM classifier.

## 4.2 Application to Benchmark Datasets

In this section, we analyze two real microarray datasets

(1) Colon cancer dataset

(2) Leukemia dataset

to evaluate the performance of our proposed B/SVM method. These two datasets are publicly available and are described in Chapter 1. All the original observations in the two datasets were transformed to the base 10 log scale. Table 4.3 displays the description of the two benchmark microarray datasets.

Table 4.3: Description of the two benchmark microarray datasets

| Dataset | Sample | Gene | Class | Publication |
|---|---|---|---|---|
| Colon cancer | 62 | 2000 | 2 | Alon et al. (1999) |
| Leukemia | 72 | 3571 | 2 | Golub et al. (1999) |

Each dataset is randomly split into a training set and a testing set 50 times and the results are the averages over the 50 repetitions. Since we don't actually know which are the true marker genes in the real datasets, average number of genes selected, average number of errors in testing set and average misclassification rate for testing set are used for comparing the proposed B/SVM with SWKC/SVM and SVM-RFE.

### 4.2.1 Colon cancer dataset

The Colon data contains 62 samples with 2000 genes (Alon et al., 1999). Among the 62 samples, 40 are tumor tissues and the rest 22 are normal ones.

Similar to the simulated data, we randomly select 20 samples (33%) of the total 62 samples to form an independent testing/validation set and the rest 42 samples (67%) form

an training set. That is,

$$\text{Colon data (62 samples)} = \begin{cases} \text{training set (42 samples)} \\ \text{testing set (20 samples).} \end{cases}$$

The training set is used to train a classifier and the testing set is used to evaluate the classification performance of trained classifier.

Table 4.4 gives the analysis results from all the three methods. It presents average number of the genes selected, average number of errors in testing set and average misclassification rate. Note that the misclassification rate is simply the number of errors in testing set divided by the size of testing set 20. From Table 4.4 we can see that the proposed B/SVM perform best among the three in terms of either index. Comparatively, SWKC/SVM performs the worst among the three methods in the sense that it uses the largest gene subset for classification that nevertheless results in the highest misclassification rate. This is consistent with our observation about SWKC/SVM in the simulated data. From the result based on B/SVM we can infer that on average, with the use of around 6 genes out of the total 2000, the trained classifier will achieve classification accuracy over 90%.

Table 4.4: Analysis result of the colon cancer microarray data

| Splittings | Indexes | B/SVM | SWKC/SVM | SVM-RFE |
|---|---|---|---|---|
| 50 | Average number of genes selected | **6.36** | 15.42 | 7.62 |
| | Average number of errors in testing set | **1.90** | 2.96 | 2.30 |
| | Average misclassification rate (%) | **9.50** | 14.80 | 11.50 |
| 200 | Average number of genes selected | **6.365** | 15.575 | 8.270 |
| | Average number of errors in testing set | **1.980** | 2.985 | 2.335 |
| | Average misclassification rate (%) | **9.900** | 14.925 | 11.720 |

### 4.2.2 Leukemia dataset

The leukemia dataset consists of 3571 genes and 72 samples (Golub et al., 1999). It is a gene expression dataset with a binary response indicating type of leukemia: type 1 ALL (acute lymphoblastic leukemia) and type 2 AML (acute myeloid leukemia). Among the 72 patients, 47 are ALL patients and 25 are AML patients. The classifier built for this dataset aims at differentiating these two types of leukemia patients.

Here, we randomly select 24 samples (33%) of the total 72 samples to form an independent testing/validation set used to evaluate the prediction accuracy. The rest 48 samples (67%) form an training set. That is,

$$\text{Leukemia data (72 samples)} = \begin{cases} \text{training set (48 samples)} \\ \text{testing set (24 samples).} \end{cases}$$

Table 4.5 presents the analysis results from all the three methods. From Table 4.5 we can see that the three methods are very competitive for this leukemia data in terms of misclassification rate or equivalently the number of errors in testing set, with SVM-RFE a little bit better than B/SVM followed by SWKC/SVM. For this leukemia data, B/SVM selects the smallest gene subset closely followed by SVM-RFE. For the leukemia data, SWKC/SVM performs the worst among the three methods in the sense that it uses the largest gene subset for classification that nevertheless results in the highest misclassification rate. From the result based on B/SVM we can infer that on average the top 10 genes can provide enough information for classification. More specifically, with use of the top 9 or 10 ranked genes out of the total 3571, the trained classifier will achieve classification accuracy 96.917%; in another word, there is only about one patient out of the total 24 in testing set that will be misclassified.

As a summary for both datasets, Tables 4.4 and 4.5 show that B/SVM always selects a relatively smaller gene subset for classification than SVM-RFE and SWKC/SVM. Especially,

Table 4.5: Analysis result of the leukemia microarray data

| Splittings | Indexes | B/SVM | SWKC/SVM | SVM-RFE |
|---|---|---|---|---|
| 50 | Average number of genes selected | **9.54** | 13.48 | 9.68 |
| | Average number of errors in testing set | 0.70 | 0.78 | **0.62** |
| | Average misclassification rate (%) | 3.083 | 3.251 | **2.583** |
| 200 | Average number of genes selected | 10.510 | 14.790 | **10.415** |
| | Average number of errors in testing set | 0.715 | 0.764 | **0.687** |
| | Average misclassification rate (%) | 2.979 | 3.229 | **2.633** |

the performance of B/SVM for the colon dataset is outstanding. With the number of genes increases, the advantage of computational efficiency in B/SVM is impressive.

## 4.3 R Code for Numerical Studies

### 4.3.1 R code for simulation studies

Here only the B/SVM algorithm in simulation studies is presented below due to space limit.

```
set.seed(2014)
sink("sim BSVM.txt", append=TRUE)
cat("The program started: ", date(), "\n \n")
for (k in 1:50){
#cat("This is the", k, "th simulation", "\n")
id_n=1:990
id_s=991:1000
index_n=sample(1:3,990,replace=TRUE)
index_s=sample(1:3,10,replace=TRUE)
n=72
p=1000
Xm=matrix(0,n,p)
Xm[,id_n[which(index_n==1)]]
    =matrix(rnorm(72*length(id_n[which(index_n==1)]),-8.0,0.2),nrow=n)
Xm[,id_n[which(index_n==2)]]
    =matrix(rnorm(72*length(id_n[which(index_n==2)]),-10.0,0.4),nrow=n)
Xm[,id_n[which(index_n==3)]]
    =matrix(rnorm(72*length(id_n[which(index_n==3)]),-12.0,1.0),nrow=n)
Xm[1:47,id_s[which(index_s==1)]]
```

```
    =matrix(rnorm(47*length(id_s[which(index_s==1)]),-6.0,0.1),nrow=47)
Xm[1:47,id_s[which(index_s==2)]]
    =matrix(rnorm(47*length(id_s[which(index_s==2)]),-8.0,0.2),nrow=47)
Xm[1:47,id_s[which(index_s==3)]]
    =matrix(rnorm(47*length(id_s[which(index_s==3)]),-10.0,0.4),nrow=47)
Xm[48:72,id_s[which(index_s==1)]]
    =matrix(rnorm(25*length(id_s[which(index_s==1)]),-6.1,1.0),nrow=25)
Xm[48:72,id_s[which(index_s==2)]]
    =matrix(rnorm(25*length(id_s[which(index_s==2)]),-8.5,0.2),nrow=25)
Xm[48:72,id_s[which(index_s==3)]]
    =matrix(rnorm(25*length(id_s[which(index_s==3)]),-11.0,0.7),nrow=25)
Xm <- as.data.frame(Xm)
label=c(rep(0,47),rep(1,25))
for(j in 1:50){
#cat("This is the", j, "th simulation", "\n")
index <- 1:nrow(Xm)
testindex <- sample(index,trunc(length(index)/3))
test.x <- Xm[testindex,]
test.y <- label[testindex]
train.x <- Xm[-testindex,]
train.y <- label[-testindex]
trainset <- cbind(train.x, train.y)
trG1 <- trainset[which(train.y==0),]
trG1.x <- trG1[,-1001]
trG1.y <- trG1[,1001]
trG2 <- trainset[which(train.y==1),]
trG2.x <- trG2[,-1001]
trG2.y <- trG2[,1001]
################## BD
mu1=colMeans(trG1.x)
mu2=colMeans(trG2.x)
var1=apply(trG1.x, 2, var)
var2=apply(trG2.x, 2, var)
d<-c()
d<-1/4*(log(1/4*((var1/var2)+(var2/var1)+2)))+1/4*((mu1-mu2)^2/(var1+var2))
A=cbind(1:1000,d)
rank=A[order(A[,2],decreasing=TRUE),1]
################ SVM
library("e1071")
EN <- c()
EachSigN <- c()
TrueN <- c()
RecoverRate <- c()
    misclass=c(rep(0,50))
    svm.model <- svm(train.x[,rank[1]],
```

```
        train.y,cachesize=500, scale=F,
        type="C-classification", kernel="linear", cross=10)
    svm.pred <- predict(svm.model, test.x[,rank[1]])
    error=length(test.y[which(test.y!=svm.pred)])
    misclass[1]=error
    for (i in 2:50){
    svm.model <- svm(train.x[,rank[1:i]],
          train.y, cachesize=500, scale=F,
          type="C-classification", kernel="linear", cross=10)
    svm.pred <- predict(svm.model, test.x[,rank[1:i]])
    error=length(test.y[which(test.y!=svm.pred)])
    misclass[i]=error
    }
    output=cbind(c(1:50),misclass)
    SigN=min(output[which.min(output[,2]),1])
EN[j]=min(misclass)
EachSigN[j]=SigN
TrueN[j]=length(intersect(rank[1:SigN],c(991:1000)))
RecoverRate[j]=TrueN[j]/SigN
print(c(SigN,EN[j],TrueN[j],RecoverRate[j]))
}
cat("The program ended:", date(), "\n")
sink()
```

#### 4.3.2 R code for real data applications

Here only the application of B/SVM to the colon cancer dataset is presented below due to space limit. The application of B/SVM to the leukemia data is similar.

```
set.seed(2014)
sink("colon SWBC.txt", append=TRUE)
cat("The program started: ", date(), "\n \n")
options(digits=5)
############ Real data Colon
load("colon.rda")
Xm=as.data.frame(log(colon.x))
label=as.factor(colon.y)
for(j in 1:50){
#cat("This is the", j, "th simulation", "\n")
index <- 1:nrow(Xm)
testindex <- sample(index,trunc(length(index)/3))
test.x <- Xm[testindex,]
test.y <- label[testindex]
```

```
train.x <- Xm[-testindex,]
train.y <- label[-testindex]
trainset <- cbind(train.x, train.y)
trG1 <- trainset[which(train.y==0),]
trG1.x <- trG1[,-2001]
trG1.y <- trG1[,2001]
trG2 <- trainset[which(train.y==1),]
trG2.x <- trG2[,-2001]
trG2.y <- trG2[,2001]
################ BD
mu1=colMeans(trG1.x)
mu2=colMeans(trG2.x)
var1=apply(trG1.x, 2, var)
var2=apply(trG2.x, 2, var)
d<-c()
d<-1/4*(log(1/4*((var1/var2)+(var2/var1)+2)))+1/4*((mu1-mu2)^2/(var1+var2))
A=cbind(1:2000,d)
rank=A[order(A[,2],decreasing=TRUE),1]
################ SVM
library("e1071")
EN=c()
EachSigN=c()
    misclass=c(rep(0,50))
    svm.model <- best.tune(svm, train.x[,rank[1]],
        train.y, cachesize=500, scale=F,
        type="C-classification", kernel="linear", cross=10)
    svm.pred <- predict(svm.model, test.x[,rank[1]])
    error=length(test.y[which(test.y!=svm.pred)])
    misclass[1]=error
    for (i in 2:50){
    svm.model <- best.tune(svm, train.x[,rank[1:i]],
        train.y, cachesize=500, scale=F,
        type="C-classification", kernel="linear", cross=10)
    svm.pred <- predict(svm.model, test.x[,rank[1:i]])
    error=length(test.y[which(test.y!=svm.pred)])
    misclass[i]=error
    }
    output=cbind(c(1:50),misclass)
    SigN=min(output[which.min(output[,2]),1])
EN[j]=min(misclass)
EachSigN[j]=SigN
print(c(SigN,EN[j]))
}
cat("The program ended:", date(), "\n")
sink()
```

# Chapter 5

# CONCLUSIONS AND DISCUSSIONS

## 5.1 Summary of Findings

Many DNA microarray data in practice are represented in form of extremely high dimensional vectors or matrices which brings great challenges in both data mining and further processing. High dimensionality not only increases the learning cost but also deteriorates the learning performance, known as the "curse of dimensionality". Therefore, dimension reduction has attracted great attentions in pattern recognition, machine learning and their applications such as microarray data analysis.

Under this framework, this thesis focuses on dimension reduction, more specifically, feature selection in DNA microarray data analysis. We proposes a new gene selection method for classification based on SVMs. In the proposed method, we first rank all the genes according to the magnitude of their Bhattacharyya distances between the two specified classes. Then the optimal gene subset is selected as the one which achieves the lowest misclassification rate in the constructed SVMs following a forward selection algorithm. Afterwards, the 10-fold cross-validation is applied to find the optimal parameters for SVM with the final optimal gene subset. As a result, the classification model is trained and built. Finally, the classification model is evaluated by its prediction performance for the testing set.

As described in Chapter 4, we compare the performance of our proposed B/SVM method with that of SVM-RFE and SWKC/SVM. SVM-RFE, a widely used and well developed method proposed by Guyon et al. (2002), is proved to be an effective embedded feature selection method. The idea of SWKC/SVM is introduced by Shim et al. (2009). As a filter feature selection method, it sets the ranking rule by a weight vector obtained from clustering approach with a distance measurement defined by using Gaussian kernel function.

The simulation results of these three methods suggest that the proposed B/SVM method outperforms the other two in terms of average misclassification rate (1.055%) and average recovery rate (95.744%). This means that B/SVM appears to be more effective and has more power in finding marker genes, compared with SVM-RFE and SWKC/SVM. B/SVM results in relatively high recovery rate with use of only a very small gene subset selected. The much smaller computational burden is another outstanding advantage of the proposed B/SVM. With B/SVM, 6.995 genes on average are selected out of the total 1000 as marker genes. The dimension of the original data is dramatically reduced. The analysis results of the colon cancer data and the leukemia data also indicate that the B/SVM algorithm can effectively reduce the dimension of data and select a small informative gene subset for classification with low misclassification rates.

As explained in Chapter 2, filter method refers to selecting informative genes according to their discriminative power without considering any knowledge of the classifier adopted, while wrapper method selects the discriminative features dependently on the classifier used. The filter method possesses the advantages of fast computability and capability of dealing with large datasets, but lacks the ability of finding optimal feature subset. Wrapper method can be expected to have good performance, but it is difficult to be scaled to large datasets because of the expensive computation cost. Embedded method can be treated as a special case of wrapper method when feature selection space is exactly the same as the hypothesis space of a classifier. The proposed B/SVM, classified as a filter method, has the drawbacks shared by all filter methods. In another word, B/SVM treats features independently, with the dependency and interaction among features ignored, and fails to take into account the interaction with classifier.

## 5.2 Future Work

Our study in Chapters 4 and 5 shows that the proposed B/SVM algorithm is very promising in marker gene selection and cancer classification. For the proposed B/SVM itself, improvements could be obtained in our future work. Firstly, we can extend the B/SVM algorithm to the more general multi-class classification. The problem we address in this thesis is a two-class classification problem. However, as a classifier SVM can be used for multi-class classification. Secondly, other classifiers besides SVM can be used to obtain potentially more reliable and more precise classification models for cancer classification. Various classifiers are available in practice and here we use SVM to select the optimal feature subset simply for the convenience that the comparison of B/SVM with the other two SVM-based methods SVM-RFE and SWKC/SVM is reasonable. Last but not the least, we can relax the normal assumption imposed on the distributions of gene expression levels by applying nonparametric kernel density estimations. When calculate the Bhattacharyya distance, we assume the gene expression levels for the two classes are normally distributed which in reality is not the case. In what follows, we attempt to explore some further modifications and adjustments of B/SVM to construct an upgraded feature selection algorithm which is free of distribution assumption.

As shown in Section 3.1.2, there is a relationship between the Bhattacharyya coefficient and the Hellinger distance which obeys the triangle inequality. For continuous probability distributions $p$ and $q$, this relationship is given by

$$D_H(p,q) = \sqrt{2} \cdot \sqrt{1 - BC(p,q)}. \tag{5.1}$$

where $D_H(p,q) = \{\int [\sqrt{p(x)} - \sqrt{q(x)}]^2 dx\}^{1/2}$ is the Hellinger distance between $p$ and $q$, and $BC(p,q) = \int \sqrt{p(x)q(x)}\, dx$ is the Bhattacharyya coefficient. According to this relationship, a new algorithm for feature selection may be constructed for DNA microarray data. In this algorithm, nonparametric approach will be used to estimate the probability density function of each gene for the two groups. That is to say, we can obtain the estimated $p$ and $q$ for each

gene by applying nonparametric, such as kernel, estimation. Afterwards, Hellinger distance can be easily calculated based on the estimated $p$ and $q$. Then we use the Hellinger distance as the ranking criterion to rank all genes. Finally, the SVM is implemented to determine the optimal gene subset that will be identified as marker genes.

Nonparametric approach imposes non or much slaker assumptions on the underlying true distribution function than parametric approach, and thus is more robust against model assumption. Kernel density estimation (KDE) is a nonparametric estimation of density function based on a random sample when the density function is known to be continuous. It is a fundamental function smoothing technique based on which inferences about the population can be made, with use of a finite sample. Because the expression level of any particular gene is a continuous random variable, it is appropriate to use kernel estimation to estimate the distribution function of gene expression levels.

Let $X_1, X_2, \ldots, X_n$ denote an independent and identically distributed sample drawn from a population with unknown density function $f$. Then the kernel density estimator $\hat{f}_n$ of $f$ is given by

$$\hat{f}_n(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right), \tag{5.2}$$

where $K$ is a continuous kernel function and $h$ is the bandwidth such that $h > 0$ and $h \to 0$ as $n \to \infty$. Some commonly used kernel functions are listed below in Table 5.2. From definition (5.2) one can see that kernel estimation produces a smoothed estimate of density functions. The smoothness can be tuned via the bandwidth parameter $h$. Thus, in order to give a kernel estimator, both kernel function and bandwidth need to be selected. With appropriately chosen bandwidth, important features of kernel estimator can be guaranteed. The kernel function $K$ usually satisfies $K \geq 0$, $K(u) = K(-u)$ and $\int_{-\infty}^{\infty} K(u)du = 1$. While the kernel function $K$ is a nonnegative real valued weight function determining how each data point contributes to the kernel density estimator, according to the literature all kernels are asymptotically equivalent and different choices of kernel will not affect the large sample

properties of the estimate (Prakasa, 1983).

Table 5.1: Some commonly used kernel functions in KDE

| Name | Function Form |
|---|---|
| Uniform | $K(u) = \frac{1}{2} I_{\{\lvert u\rvert \leq 1\}}$ |
| Triangular | $K(u) = (1 - \lvert u\rvert) I_{\{\lvert u\rvert \leq 1\}}$ |
| Epanechnikov | $K(u) = \frac{3}{4}(1 - u^2) I_{\{\lvert u\rvert \leq 1\}}$ |
| Quartic (Biweight) | $K(u) = \frac{15}{16}(1 - u^2)^2 I_{\{\lvert u\rvert \leq 1\}}$ |
| Triweight | $K(u) = \frac{35}{32}(1 - u^2)^3 I_{\{\lvert u\rvert \leq 1\}}$ |
| Tricube | $K(u) = \frac{70}{81}(1 - \lvert u\rvert^3)^3 I_{\{\lvert u\rvert \leq 1\}}$ |
| Gaussian | $K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}}$ |
| Cosine | $K(u) = \frac{\pi}{4} \cos\left(\frac{\pi}{2} u\right) I_{\{\lvert u\rvert \leq 1\}}$ |

We redo the simulation studies for B/SVM method as in Section 4.1 with use of kernel density estimations. In DNA microarray data we have relatively small sample size, and thus the kernel $K$ need to be carefully chosen. We used the Gaussian kernel for all kernel estimations. Here, we select the bandwidth that minimizes the mean squared error (MSE) of a kernel estimator for estimating the probability density function, that is to say we set the bandwidth $h = n^{-1/5}$ . As in Section 4.1, we follow the same way splitting data into training set and testing set, and use the same 10-fold cross-validation in classification step. Unfortunately, the results for B/SVM show that the classification of independent validation set is not as good as expected and the recovery rate is not as high as that achieved by B/SVM under normal assumption. The main reason for the un-preference of kernel smoothed nonparametric B/SVM over the parametric B/SVM in Section 4.1 is simply that the simulated samples are from normal populations and the parametric B/SVM in Section 4.1 utilized this extra information while the nonparametric B/SVM here doesn't. This extra information about the populations will for sure increase the efficiency and accuracy of any classifier.

Some other reasons for the un-preference may include

- In the training set, the balance of samples from two classes may be a problem. We randomly select 2/3 of the simulated data as training set. With small sample size of simulated data, the training set is even smaller and it may contain only very few data from one class which makes the resulted kernel estimation very unreliable.

- Due to the small sample size, bandwidth is not appropriately chosen. We arbitrarily set the bandwidth as $n^{-1/5}$. It may be more appropriate to use data-driven bandwidths for different genes rather than a uniform one for all genes.

As discussed above, a more reliable and generalized B/SVM method can be obtained by addressing those problems. Improvements can be achieved in future research to increase classification reliability and decrease misclassification rate.

# Bibliography


[1] Alon, U. and Barkai, N. and Notterman, D.A. and Gish, K. and Ybarra, S., Mack, D. and Levine, A.J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, **96**, 6745-6750.

[2] Ben-Dor,A., Bruhn,L., Friedman,N., Nachman,I., Schummer,M. and Yakhini,Z. (2000). Tissue classification with gene expression profiles. *Journal of Computational Biology*, **7**, 559-583.

[3] Bhattacharyya, A. (1943). On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematical Society* **35** 99-109.

[4] Breiman, L., Friedman, J., Stone, C. J. and Olshen, R.A. Classification and Regression Trees Taylor  Francis Press.

[5] Broberg, P. (2002). Ranking genes with respect to differential expression. *Genome Biology*, **3**, preprint0007.1-0007.23.

[6] Clarke, R., Ressom, H.W., Wang, A., Xuan, J., Liu, M.C., Gehan, E.A. and Wang, Y. (2008). The properties of high-dimensional data spaces: implications for exploring gene and protein expression data. *Nature Reviews Cancer*, **8**, 37-49.

[7] Cristianini, N. and Shawe, T.J. (2000). An introduction to SVM. Cambridge University Press. Cambridge, UK.

[8] Cui, X.Q. and Churchill, G.A. (2003). Statistical tests for differential expression in cDNA microarray experiments. *Genome Biology*, **4**, 210-215.

[9] Dash, M. and Liu, H. Feature selection for classification. (1997). *Intelligent Data Analysis*,**1**, 131-156.

[10] Dettlling, M. (2004). BagBoosting for tumor classification with gene expression data. *Bioinformatics*, **18**, 3583-3593.

[11] Dudoit, S., Fridlyand, J. and Speed, T.P. (2002). Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, **97**, 77-87.

[12] Eisen, M., Spellman, P., Brown, P., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *PNAS*, **95** 14863-14868.

[13] 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., Bloomfield, C.D. and Lander, E.S. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, **286**, 531-537.

[14] Guyon, I., Weston, J. and Barnhill, S. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, **46**, 389-422.

[15] Guyon, I. and Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, **3**, 1157-1182.

[16] Koo, J.Y., Sohn, I., Kim, S., Lee, J.W. (2006). Structured polychotomous machine diagnosis of multiple cancer types using gene expression. *Bioinformatics*, **22**, 950-990.

[17] Li, F. and Yang, Y. (2005). Analysis of recursive gene selection approaches from microarray data. *Bioinformatics*, **21**, 3741-3747.

[18] Li, L.P., Weinberg, C.R., Darden, T.A. and Pedersen, L.G. (2001). Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. *Bioinformatics*, **17**, 1131-1142.

[19] Li, T., Zhang, C. and Ogihara, M. (2004). A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics*, **20**, 2429-2437.

[20] Li, J., Tang, X. and Liu, J. (2008). A novel approach to feature extraction from classification models based on information gene pairs. *Pattern Recognition*, **41**, 1975-1984.

[21] Liu, H., Li, J. and Wong, L. (2002). A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns. *Genome Informatics*, **13**, 51-60.

[22] Loog, M., Duin, R.P.W. and Haeb-Umbach, R. (2001). Multiclass linear dimension reduction by weighted pairwise fisher criteria. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23**, 762-766.

[23] Mercer, J. (1909). Functions of positive and negative type and their connection with theory of integral equations. *Philosophical Transactions of the Royal Society of London - Series A*, **209**, 415-446.

[24] Musa, H. A., Dilek, C., Omer, D. and Mehmet, S. I. (2006). Gene Expression Profile Classification: A Review. *Bioinformatics*, **1**, 55-73.

[25] Olshen, A. B. and Jain, A. N.(2002). Deriving quantitative conclusions from microarray expression data. *Bioinformatics*, **18(7)**, 961-970.

[26] Park, C.Y., Koo, J.-Y., Kim, S., Sohn, I. and Lee, J.W. (2008). Classification of gene functions using SVM for time-course gene expression data. *Computational Statistics and Data Analysis*, **52**, 2578-2587.

[27] B. L. S. Prakasa Rao (1983). Non-Parametric Functional Estimation, Academic Press, Orlando, Florida.

[28] Pomeroy, S., Tamayo, P., Gaasenbeek, M., Sturla, L., Angelo, M., McLaughlin, M., Kim, J., Goumnerova, L., Black, P. and Lau, C.(2002). Prediction of central nervous system embryonal tumor outcome based on gene expression. *Nature*, **415**, 436-442.

[29] Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S., Yeang, C.H., Angelo, M., Ladd, C., Reich, M., Latulippe, E., Mesirov, J.P., Poggio, T., Gerald, W., Loda, M., Lander, E.S. and Golub, T.R. (2001). Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Academy of Sciences of the United States of America*, **98**, 15149-15154.

[30] Ross, D.T., Scherf, U., Eisen, M.B., Perou, C.M., Rees, C., Spellman, P., Iyer, V., Jeffrey, S.S., Van de Rijn, M., Waltham, M., Pergamenschikov, A., Lee, J.C., Lashkari, D., Shalon, D., Myers, T.G., Weinstein, J.N., Botstein, D. and Brown, P.O. (2000). Systematic variation in gene expression patterns in human cancer cell lines. *National Genetics*, **24**, 227-235.

[31] Saeys, Y., Inza, I. and Larranaga, P. (2007). A review if feature selection techniques in bioinformatics. *Bioinformatics*, **19**, 2507-2517.

[32] Shalon, D., Smith, S. J. and Brown, P. O. (1996). A DNA microarray system for analyzing complex DNA samples using two-color fluorescent probe hybridization. *Genome Research.* **6(7)**, 639-645.

[33] Shedden, K.A., Taylor, J.M., Giordano, T.J., Kuick, R., Misek, D.E., Rennert, G., Schwartz, D.R., Gruber, S.B., Logsdon, C., Simeone, D., Kardia, S.L., Greenson, J.K., Cho, K.R., Beer, D.G., Fearon, E.R. and Hanash, S. (2003). Accurate molecular classification of human cancers based on gene expression using a simple classifier with a pathological tree-based framework. *The American Journal of Pathology*, **163**, 1985-1995.

[34] Shim, J., Sohn, I., Kim, S., Lee, S.W., Green, P.E. and Hwang, C. (2009). Selection marker genes for cancer classification using supervised weighted kernel clustering and the support vector machine. *Computational Statistics and Data Analysis*, **53**, 1736-1742.

[35] Slonim, D. K., Tamayo, P., Mesirov, J. P., Golub, T. R. and Lander, E. S. (2000). Class prediction and discovery using gene expression data. *Proceedings of the fourth annual international conference on Computational molecular biology*, 263-272.

[36] Speer, N., Spieth, C. and Zell, A. (2005). Spectral clustering gene ontology terms to group genes by function *In Proceedings of the 5th Workshop on Algorithms in Bioinformatics (WABI)*

[37] Statnikov, A., Aliferis, C.F., Tsamardinos, I., Hardin, D. and Levy, S. (2005). A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. *Bioinformatics*, **21**, 631-643.

[38] Tibshirani, R., Hastie, T., Narasimhan, B. and Chu, G. (2002). Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences of the United States of America*, **99**, 6567-6572.

[39] Tschentscher, F., H*ü*sing, J., H*ö*lter, T., Kruse, E., Dresen, I.G., J*ö*ckel, K.H., Anastassiou, G., Schilling, H., Bornfeld, N., Horsthemke, B., Lohmann, R. and Zeschnigk, M. (2003). Tumor classification based on gene expression profiling shows that uveal melanomas with and without monosomy 3 represent two distinct entities. *Cancer Research*, **63**, 2578-2584.

[40] Tusher, V.G., Tibshirani, R. and Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *PNAS*, **98**, 5116-5121.

[41] Vapnik, V. (1998). Statistical Learning Theory. Wiley, New York.

[42] Wang, Y., Miller, D.J. and Clarke, R. (2008). Approaches to working in high-dimensional data spaces: gene expression microarrays. *British Journal of Cancer*, **98**, 1023-1028.

[43] Wang, Z., Wang, Y., Xuan, J., Dong, Y., Bakay, M., Feng, Y., Clarke, R. and Hoffman, E.P. (2006). Optimized multilayer perceptrons for molecular classification and diagnosis using genomic data. *Bioinformatics*, **22**, 755-761.

[44] Xiong, M.M., Li, W.J., Zhao, J.Y, Jin, L. and Boerwinkle, E. (2001). Feature (gene) selection in gene expression-based tumor classification. *Molecular Genetics and Metabolism*, **73**, 239-247.

[45] Wahba, G. (1999). The bias-variance tradeoff and the randomized GACV. *Advances in Neural Information Processing Systems*, **11**, 620-626

[46] Zhang, H.H., Ahn, J., Lin, X., Park, C. (2006). Gene selection using support vector machines with non-convex penalty. *Bioinformatics*, **22**, 88-95.

[47] Zhou, X. and Tuck, D.P. (2007). MSVM-RFE: extensions of SVM-RFE for multiclass gene selection on DNA microarray data. *Bioinformatics*, **23**, 1106-1114.