



139
831
THS

This is to certify that the dissertation entitled

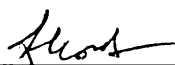
LARGE DIMENSION AND SMALL SAMPLE SIZE
PROBLEMS: CLASSIFICATION, GENE SELECTION AND
ASYMPTOTICS

presented by

JUN LUO

has been accepted towards fulfillment
of the requirements for the

Ph.D degree in Statistics and Probability



Major Professor's Signature

12-04-2006

Date

Doctoral Dissertation

MSU is an Affirmative Action/Equal Opportunity Institution

LIBRARY
Michigan State
University

PLACE IN RETURN BOX to remove this checkout from your record.
TO AVOID FINES return on or before date due.
MAY BE RECALLED with earlier due date if requested.

DATE DUE	DATE DUE	DATE DUE

**LARGE DIMENSION AND SMALL SAMPLE SIZE
PROBLEMS: CLASSIFICATION, GENE
SELECTION AND ASYMPTOTICS**

By

Jun Luo

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Statistics and Probability

2006

ABSTRACT

LARGE DIMENSION AND SMALL SAMPLE SIZE PROBLEMS: CLASSIFICATION, GENE SELECTION AND ASYMPTOTICS

By

Jun Luo

Classification of patient samples is an important aspect of cancer diagnosis and treatment. The support vector machine (SVM) and penalized logistic regression (PLR) have been successfully applied to microarray cancer diagnosis problems. The two methods treat equal penalty on each loss. That may lead to misclassification on unbalanced data. So we propose ν -ridge regression (ν -RR), which puts a generalized weight on the loss of each sample and optimizes the weight vector by the model itself, as an alternative method to the SVM and PLR for classification in microarray cancer diagnosis. Often a primary goal in microarray analysis is to identify the genes which are most responsible for classification in microarray. Two gene selection methods are considered, univariate ranking (UR) and recursive feature elimination (RFE).

Simulation on the well known leukemia data and breast cancer prognosis data indicates that ν -RR combined with either UR or REF tends to select less significant genes than other methods. Meanwhile, ν -RR performs superior to SVM and PLR with a lower rate in both cross-validation error and test error.

One of the weaknesses of the SVM is that given a tumor sample, it only predicts a cancer class label but does not provide any estimation of the underlying probability. The penalized logistic regression has the advantage of additionally providing an estimate of the underlying probability of being assigned to each class, but in fact it does not offer any estimate for the

probability of the outcome class, conditional on an individual gene variable. We propose the conditional logistic regression (CLR) model, which is an alternative for the microarray cancer diagnosis classification, for the underlying probability of the response given any gene variable. In addition, since a primary goal in microarray cancer diagnosis is gene selection, we propose a new method called modified univariate ranking (MUR) as a new choice for dimension reduction.

We show that when applied to a microarray data for classification, CLR performs similarly to SVM, PLR and BMA, but CLR has the advantage of providing the probability of the outcome class, conditional on any individual gene variable. Empirical results on leukemia and breast cancer data indicate that the CLR method combined with one gene selection method (MUR, BSS/WSS or RFE) tends to perform superior on both CV-error and test error rate.

Microarray data typically have very high dimension p and much smaller sample size n . Classical asymptotic theory deals with p fixed and n goes to infinity, which is no longer appropriate for microarray data analysis. There are discussions in the literature about the behavior of estimations when both p and n tend to infinity, but very few dealing with n fixed and p tends to infinity. The latter situation seems more relevant to microarray data in practice. Here we outline and describe the asymptotical behavior of ridge regression estimations when sample size n is fixed and dimension p tends to infinity. Given certain data, mean squared error consistency is established under certain regularity conditions. When there are only finite number of important genes that are actually related to the outcome, we propose a variable screening method to eliminate genes which are unrelated to the outcome and prove the asymptotic consistency of the procedure. After screening, the dimension-reduced microarray data can be further analyzed via a well-known variable selection method such as AIC and BIC. Some simulation results for testing the performance of the screening method are also presented.

Copyright by

Jun Luo

2006

To my deeply loved mother Laxiang Liu, my lenient father Youxin Luo and
my supportive sister Yan Luo.

To all my friends at Michigan State University. They have kept me
accompany in the past 4 years.

ACKNOWLEDGMENTS

First of all, I sincerely appreciate my PhD advisor, Professor Yijun Zuo, for what he has done for me in the program. His excellent guidance and thoughtful lectures sparked my passion in statistics and life. My dissertation is an allusion without Professor Yijun Zuo's effort and time spent on me.

I also want to give my special thanks to Professor Hira Koul, who has been very strict and very careful to be a substitute of Professor Yijun Zuo. I am happy to have him as my committee chair. I value his advice in the rehearsal and writing of the thesis more than he could imagine.

I would like to thank Professors Sarat Dass, Robert Tempelman and Lijian Yang for serving on my thesis committee and for their valuable advice and kind cooperation. I am also grateful to Professor James Stapleton for his generous help and spiritual support.

I have made some lifetime friends at Michigan State University, such as Winny Chiang, Kirk D Dolan, Hongwen Guo, Wenmei Huang, Fang Li, Rong Liu, Aaron Thomas Porter, Yu Sun, Weixing Song, Tony Wang, Lily Wang, Jing Wang, Hui Zhang, Yongfang Zhu and Yanwei Zhang. I thank them for their moral support and friendship.

In the last, I am grateful to my family in China. My loved mother, father and sister have been supportive of me all the way. They tried every possible means to make sure I am happy and healthy. No words can describe my feelings at this moment. I will just say "I Love You All, My Dear Family!"!

TABLE OF CONTENTS

LIST OF TABLES	ix
LIST OF FIGURES	x
1 Introduction	1
1.1 Standard SVM for 2-class classification	4
1.2 ν -SVM formulation	7
1.3 Equivalence proof	8
1.4 L_1 -norm Support Vector Machine	9
2 Classification and Gene selection of Cancer Microarrays by ν-Ridge Regression	11
2.1 Approach	13
2.1.1 ν -Ridge Regression	13
2.1.2 Quadratic Programming	14
2.1.3 Computation of w and b	15
2.1.4 Feature selection	16
2.2 Results	17
2.2.1 Choosing λ and ν	17
2.2.2 Leukemia data	19
2.2.3 Breast cancer prognosis data	20
2.3 Discussion	21
2.4 Conclusion	22
3 Density Curve Estimation and Classification of Microarray by Conditional Logistic Regression Model	23
3.1 Approach	25
3.1.1 Conditional Logistic Regression Formulation	25
3.1.2 Algorithm	26
3.1.3 Feature Selection	29
3.1.4 Cut-off statistics a_p	31
3.2 Results	32
3.2.1 Breast cancer prognosis data	32
3.2.2 Leukemia data	33
3.3 Conclusion	34

4 Asymptotics and Variable Screening for Microarrays with Fixed Small Sample Size and Large Dimension	35
4.1 Mean Squared Error Consistency	37
4.2 Variable Screening Procedure	40
4.3 Simulation Result	43
4.4 Conclusions and Discussions	48
5 Gene Selection Methods for Microarray Data	50
5.1 Univariate Ranking	51
5.2 Recursive Feature Elimination	52
5.3 Ratio of Between-group sum of squares to Within-group sum of squares . . .	53
5.4 Clustering method	53
5.5 Pairwise Ranking Method	55
BIBLIOGRAPHY	57

LIST OF TABLES

2.1	Parameter selection.	18
2.2	Comparison of leukemia classification methods	19
2.3	Published results of leukemia classification methods	19
2.4	Comparison of breast cancer classification methods.	20
3.1	Comparison of classification methods for breast cancer prognosis data.	32
3.2	Output of CLR on breast cancer data.	32
3.3	Comparison of CLR on leukemia data.	33
3.4	Comparison of leukemia classification methods.	33
3.5	Parameters for the chosen genes in the final classifier from CLR MUR.	33
3.6	Parameters for the selected genes in the final classifier from CLR RFE.	34
4.1	Simulation results for MSE consistence based on 1,000 runs (x fixed), $h_p = p^{0.25}$, $n = 60$, X is from the flowchart	46
4.2	Simulation results for close-to-0 nonzero β_i values based on 1,000 runs (x fixed)	47
4.3	Simulation results for distinguished nonzero β_i values based on 1,000 runs (x fixed)	48

LIST OF FIGURES

1.1	Two class classification	5
1.2	Support vector machine	5
4.1	Flowchart of data generation	44

CHAPTER 1

Introduction

The aim of the dissertation is to provide an up to date review of some different approaches to classification (mainly machine learning methods), propose two new methods of classification, compare their performance on some challenging data sets, and draw conclusions on their applicability to realistic microarray problems.

The task of classification occurs in a wide range of human activity. At its broadest, the term could cover any context in which some decision or forecast is made on the basis of currently available information, and a classification procedure is then some formal method for repeatedly making such judgements in new situations. In this dissertation we shall consider a more restricted interpretation. We shall assume that the problem concerns the construction of a procedure that will be applied to a continuing sequence of cases, in which each new case must be assigned to one of a set of pre-defined classes on the basis of observed attributes or features. The construction of a classification procedure from a set of data for which the true classes are known has also been variously termed as pattern recognition, discrimination, or supervised learning (in order to distinguish it from unsupervised learning or clustering in which the classes are inferred from the data).

Microarray experiments raise numerous statistical questions in image analysis, experimental design, cluster and discriminant analysis, and multiple hypothesis testing. Here we focus on the classification of tumors using gene expression data. Three main types of statistical problems are associated with tumor classification: (1) identification of new tumor classes using gene expression profiles, cluster analysis/unsupervised learning; (2) classifica-

tion of malignancies into known classes, discriminant analysis/supervised learning; and (3) identification of "marker" genes that characterize the different tumor classes, called variable selection. In a two class microarray classification problem, we are given a training data set $(x_1, y_1) \dots (x_n, y_n)$, where the input x_i is a p -vector corresponding to the gene expression values of the i th experiment (or samples), $x_i = (x_{i1}, \dots, x_{ip})$, see Eisen (1998), Getz (2000), Slonim (2000), Yang (2001), van't Veer (2002) and Chen and Chen (2003). The output y_i is a binary class label and assumed taking values in $\{-1, +1\}$, see Xiong (2000) and Yeung (2001). The problem of interest is to find a classification rule from the training data, so that we can actually assign a class label from $\{-1, +1\}$ when given a new sample x with p gene expression measurements.

Data from these new types of experiments present a "large p , small n " problem; that is, a very large number of variables (genes) relative to the number of observations (tumor samples). In statistics when the number of variables is much larger than the number of samples, one is said to be facing the problem of the so called "curse of dimensionality", and the function estimated (in here, it is classifier of the microarray data) may be over-fitting (i.e. very high accuracy in fitting the training samples but very low accuracy in assigning labels for the test data). This problem is mitigated by using some gene selection methods and making sure the classifier is smooth. So that the new samples similar to those in the training set will be labeled similarly.

Machine learning is a scientific field that addresses the question of how to program systems to automatically learn and to improve with experience. Vapnik (1995) successfully invented the application of machine learning methods (called support vector machine or SVM) to two class classification problems. As a consequence, SVM has been applied for classification in cancer microarray data. Besides SVM, past publications on cancer classification using gene expression data have focused mainly on the cluster analysis of both tumor samples and genes and include applications of hierarchical clustering (Alon *et al.* (1999) and Perou *et al.* (1999)) and partitioning methods such as self-organizing maps (Golub (1999)). Dudoit *et al.* (2002) compared the performance of a bunch of different discrimination methods for the classification of tumors based on gene expression profiles. These methods include traditional ones, such as nearest-neighbor and linear discriminant analysis, as well as more

modern ones, such as classification trees, bagging and boosting.

Statistical approaches are generally characterized by having an explicit underlying probability model, which provides a probability of being in each class. In addition, it is usually assumed that the techniques will be used by statisticians, and hence some human intervention is assumed with regard to variable selection and transformation, and overall structuring of the problem. Unfortunately, SVM does not offer any underlying probability in the classification. Zhu (2004) proposed penalized logistic regression (PLR) for classification in cancer microarray and an estimator of the underlying probability.

But neither SVM nor PLR can avoid the misclassification on unbalanced data set: the larger the training sample size for one class, the smaller its corresponding classification error rate. The main cause is that the penalty of misclassification is taken to be the same for each training sample. Wang and Yang (2004) used weighted support vector machine on the prediction of membrane protein types. Inspired by this, we proposed a new non-parametric classification method called ν -ridge regression (ν -RR), where each sample contributes differently. Since SVM, PLR and any discriminant methods do not study the conditional probability of the outcome, conditional upon each gene expression level, our conditional logistic regression is indeed very useful in classification and providing underlying distribution. The detailed work is in chapter 2 and 3.

DNA microarrays now permit scientists to screen thousands of genes simultaneously and determine whether these genes are active, hyperactive or silent in normal or cancerous tissue. Because these new microarray devices generate bewildering amounts of raw data, new analytical methods must be developed to sort out whether cancer tissues have distinctive features. In this dissertation, we address the problem of selection of a small subset of genes from broad patterns of gene expression data, recorded on DNA microarrays. Using available training examples from cancer and normal patients, we build a classifier suitable for genetic diagnosis, as well as drug discovery. Previous attempts to address gene selection in cancer microarray are listed in chapter 5, including correlation techniques and pairwise selection. A new method for gene selection based on univariate ranking method is also proposed. We demonstrate experimentally that the genes selected by this technique yield better classification performance and are biologically relevant to cancer. These findings

are consistent with the published results in the literature. In contrast with the published methods, our method eliminates gene redundancy automatically and yields better and more compact gene subsets. Simulations are done with well-known golden standard data sets Leukemia and Breast Cancer Prognosis data. For gene selection in asymptotic case, existing statistical methods deal with a single gene at a time (see Chen *et al.* (2003)), or deal with the case in which the sample size $n \rightarrow \infty$ (see Shao (2006)). Chapter 4 describes insight for the analysis of microarray data with the more realistic case that sample size is fixed and the dimension of variables $p \rightarrow \infty$.

1.1 Standard SVM for 2-class classification

The original SVM was motivated by the idea of maximizing the distance between the separation hyperplane and the closest point in the training samples, also of making the prediction function as smooth as possible. See Vapnik (1995, 1998, 2001 and 2002), Ripley (1996), Bradley (1998), Allwein (2000), Mukherjee (2000), Scholkopf (2000) and Evgeniou (2000). Discriminant methods have been used for classification in cancer microarray data, see Dudoit (2002). Based on the n training data $(x_1, y_1) \dots (x_n, y_n)$, where x_i is a p -vector in the dot product space (through the chapter, we will focus on the linear separation hyperplane and use x_i as the basis function), y_i is the label from $\{+1, -1\}$ for the i th sample. The classification rule is given by:

$$f(x) = \text{sign}(\langle w, x \rangle + b). \quad (1.1)$$

To address the classifier for the microarray data with maximal margin distance, some statistical methods have been applied. Throughout the dissertation, $\langle x, y \rangle$ is the inner product of the two vectors x and y .

For a two class classification problem, the linear SVM fits a model $f(x) = \langle w, x_i \rangle + b$, where w and b are such that

$$\min_{w, b, \xi} \left[\frac{1}{2} \|w\|_2^2 + C \sum_{i=1}^n \xi_i \right], \quad (1.2)$$

subject to the condition

$$y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \text{for all } i=1, 2, \dots, n, \quad (1.3)$$

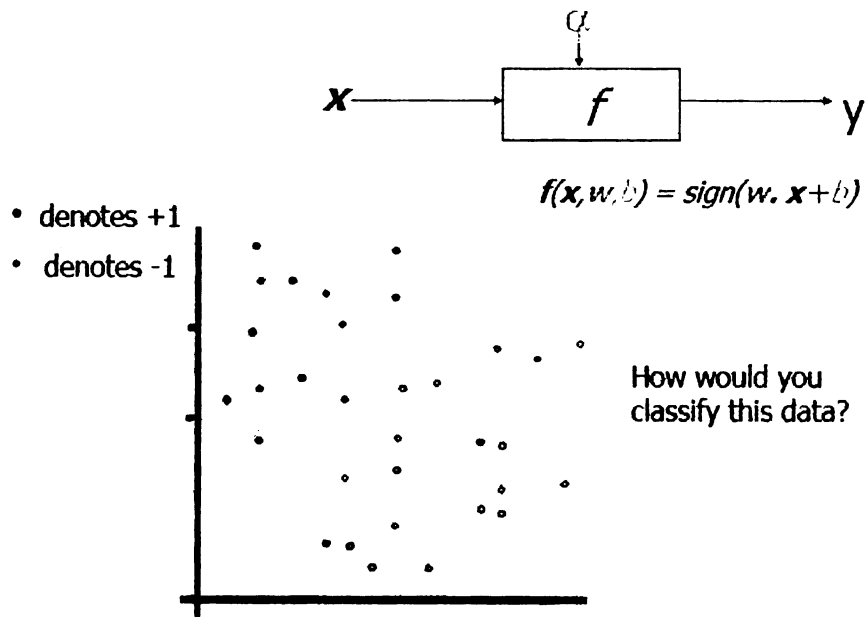


Figure 1.1. Two class classification

Linear Classifiers

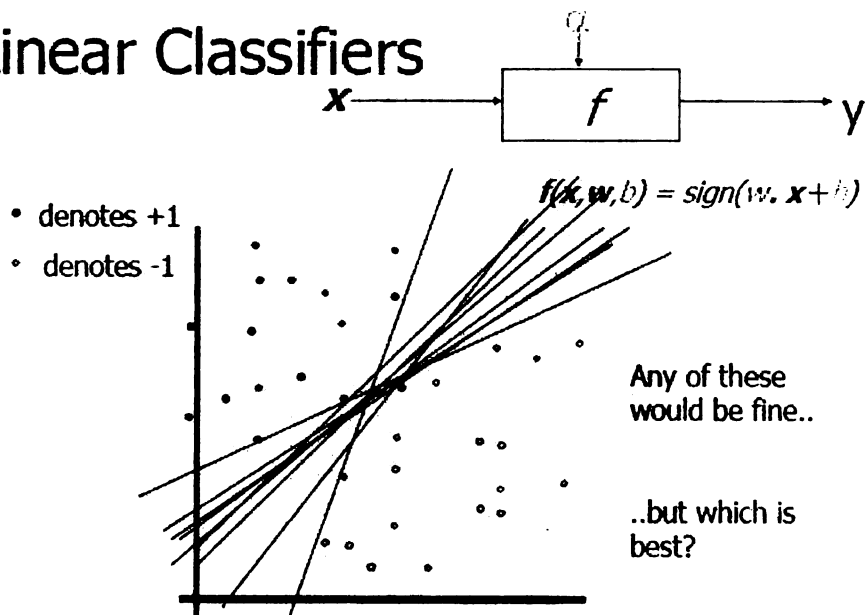


Figure 1.2. Support vector machine

where ξ'_i s is the soft margin which allows the possibility of violation on the distance constraint for non-separable data (i.e. when $\xi_i > 1, \forall i = 1, 2, \dots, n$). This new formulation trades off two goals of finding a hyperplane with large margin (minimizing $\|w\|$) and finding a hyperplane that separates the data well (minimizing the loss). And the constant $C > 0$ controls the trade-off. In practice, C is determined by cross validation. The classifier is given by $f(x) = \text{sign}(\langle w, x \rangle + b)$. The above set-up is called C-SVM.

Ideally, we are interested in the number of nonzero ξ_i , as that is the count of errors made by our classifier on the set of training examples. This count is the L_0 norm, and is discontinuous and non-convex. We would like to stay close to L_0 , while maintaining convexity. We take the Lagrangian in the usual manner:

$$L(w, \xi, b, \alpha, \beta) = \frac{1}{2} w' w + C \sum_{i=1}^n \xi_i + \sum_{i=1}^n \alpha_i [1 - \xi_i - y_i(x'_i w + b)] - \sum_{i=1}^n \beta_i \xi_i. \quad (1.4)$$

To find the dual form of the problem, we first need to minimize $L(w, \xi, b, \alpha)$ with respect to w, ξ and b (for fixed α and β), i.e. $\min_{w, \xi, b} L(w, \xi, b, \alpha, \beta)$. Since the Lagrangian function is linear in α and β , we can not set the gradient with respect to α and β to zero. We obtain the following dual optimization problem:

$$\max_{\alpha} \left[\sum_{i=1}^n -\frac{1}{2} \sum_{i,j=1}^n y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle \right], \quad (1.5)$$

subject to the constraint $0 \leq \alpha \leq C, \sum_{i=1}^n \alpha_i y_i = 0$. The introduction of the box constraint on α_i is necessary to ensure that the Lagrangian will be bounded (i.e., that we can not drive the cost to $-\infty$). This will occur in the case in which $(C - \alpha_i)$ is negative and ξ_i goes to ∞ , the summed expression $\sum_{i=1}^n (C - \alpha_i) \xi_i$ goes to $-\infty$. The box constraint prevents this from happening.

The dual still has a quadratic objective, and differs from the optimal margin classifier only in the introduction of the box constraint. Indeed, we can still use $w = \sum_{i=1}^n \alpha_i x'_i y_i$ to give us the optimal value of w in terms of the optimal value of α . We must also verify that the KKT dual-complementary conditions are still satisfied in this optimization problem:

$$\alpha_i = 0 \Rightarrow y_i[x'_i w + b] \geq 1,$$

$$\alpha_i = C \Rightarrow y_i[x'_i w + b] \leq 1,$$

$$0 < \alpha_i < C \Rightarrow y_i[x'_i w + b] = 1.$$

As before, α_i will be nonzero only for the support vectors, where the set of support vectors now includes all data points on the margin boundary as well as those on the wrong side of the margin boundary.

Most often, one uses the regularized optimization problem which is so called standard L_2 -norm SVM:

$$\min_{w,b} \left[\sum_{i=1}^n (1 - y_i(\langle w, x_i \rangle + b))_+ + \lambda \|w\|_2^2 \right] \quad (1.6)$$

where $\lambda \geq 0$ is a tuning parameter, and the classifier is defined in (1.1). Model (1.2) and (1.3) is basically equivalent to (1.6) in the sense that for any given $C > 0$, there exists a $\lambda = 1/(2C)$ so that these two models share the same optimal value w, b , i.e. we will get the exactly same classification rule. Note that (1.6) has the form loss and penalty, and λ is the tuning parameter that controls the trade-off between the loss and penalty. The loss $(1 - yf)_+$ is called the hinge loss, and the L_2 -norm penalty is called the ridge penalty.

1.2 ν -SVM formulation

Motivated by the idea of maximizing the margin, another approach to get the classifier is considering

$$\min_{w,b,\rho,\xi} \left[\frac{1}{2} \|w\|_2^2 + C(-\nu\rho + 1/n \sum_{i=1}^n \xi_i) \right], \quad (1.7)$$

subject to the constraint

$$y_i(\langle w, x_i \rangle + b) \geq \rho - \xi_i, \quad \xi_i \geq 0, \quad \rho \geq 0, \quad \text{for all } i=1,2,\dots,n, \quad (1.8)$$

where $C > 0$ and $\nu \in [0, 1]$ are parameters. To understand the role of ρ , note that for $\xi = 0_{n \times 1}$, the constraint simply means the two classes are separated by the margin $\rho/\|w\|$. The classifier for (1.7) is defined in (1.1). Applying the Lagrangian dual variable method on model (1.8), it turns out that C in (1.7) does not matter at all and (1.7) obtains the same decision function as

$$\min_{w,b,\rho \geq 0} \left[\frac{1}{2} \|w\|_2^2 - \nu\rho + \frac{1}{n} \sum_{i=1}^n \xi_i \right] \quad (1.9)$$

under the constraint (1.8), ν is a parameter in $[0,1]$. Model (1.9) is known as ν -SVM, which is proposed by Scholkopf, Smola, Williamson, and Bartlett (2000) for classification. The relationship between SVM and ν -SVM with 2-norm penalty is analyzed below. We leave the comparison with 1-norm to the feature selection section.

The ν -SVM possesses some additional properties than C-SVM besides that both methods provide the classifier using the training data. For example, ν -SVM has the advantage of using a parameter ν on controlling the number of support vectors and the fraction of training errors.

1.3 Equivalence proof

Generally (1.6) and (1.8) and (1.9) are two different problems with the same optimal solution set.

Theorem 1. In model (1.8) with constraint (1.9), for any given $\nu \in [0,1]$ with optimal solution $\tilde{\rho} > 0$, we conclude (1.8) and (1.9) has the same classifier as (1.6) with $\lambda = n\tilde{\rho}/2$.

Proof. Focus on model (1.6), suppose there are \tilde{w}, \tilde{b} such that

$$(\tilde{w}, \tilde{b}) = \arg \min_{w, b, \xi} \left[\frac{1}{2} \|w\|_2^2 - \nu \tilde{\rho} + 1/n \sum_{i=1}^n \xi_i \right],$$

subject to the condition

$$y_i(< w, x_i > + b) \geq \tilde{\rho} - \xi_i, \xi_i \geq 0, \text{ for all } i=1,2,\dots,n.$$

Now we do the following transformation of variables:

$$\tilde{w} = \tilde{\rho} w', \tilde{b} = \tilde{\rho} b', \tilde{\xi}_i = \tilde{\rho} \xi'_i.$$

Then the classifier for (1.8) and (1.9) becomes

$$f(x) = \text{sign}(< \tilde{\rho} w', x >) + \tilde{\rho} b' = \text{sign}(< w', x >) + b'.$$

Recall (w', b') satisfies

$$(w', b') = \arg \min_{w, b, \xi_i} \left[1/2 \tilde{\rho}^2 \|w\|_2^2 - \nu \tilde{\rho} + \tilde{\rho}/n \sum_{i=1}^n \xi_i \right],$$

subject to the condition

$$y_i(< w, x_i > + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \text{for all } i=1,2,\dots,n.$$

The above optimization problem is the same as

$$\min_{w', b', \xi'_i} \left[\frac{n\tilde{\rho}}{2} \|w'\|_2^2 + \sum_{i=1}^n \xi'_i \right],$$

subject to the condition

$$y_i(< w', x_i > + b') \geq 1 - \xi'_i, \quad \xi'_i \geq 0, \quad \text{for all } i=1,2,\dots,n.$$

That is model (1.6) with $\lambda = n\tilde{\rho}/2$, so the classifier is $f(x) = \text{sign}(< w', x > + b')$. Obviously we get the same decision function from the two models under the condition $\tilde{\rho} > 0$.

1.4 L_1 -norm Support Vector Machine

The standard 2-norm SVM is known for its good performance in two-class classification. Hastie (2004) studied the regularization path for SVMs. In the section, we talk about 1-norm SVM. 1-norm SVM has some advantage over the standard 2-norm SVM, especially 1-norm SVM functions as one variable selection method and a classification method.

In standard 2-class classification problems, we are given a set of training data $(x_1, y_1), \dots, (x_n, y_n)$, where the input $x_i \in R^p$, and the output $y_i \in \{1, -1\}$ is binary. We wish to find a classification rule from the training data, so that when given a new input x , we can assign a class y from $\{1, -1\}$ to it. To handle this problem, 1-norm support vector machine (1-norm SVM) was considered:

$$\min_{\beta_0, \beta} \sum_{i=1}^n [1 - y_i(\beta_0 + \sum_{j=1}^q \beta_j h_j(x_i))]_+, \quad (1.10)$$

subject to the condition $\|\beta\|_1 = |\beta_1| + \dots + |\beta_q| \leq s$, where $D = h_1(x), \dots, h_q(x)$ is a dictionary of basis functions, and s is a tuning parameter. The solution is denoted as $\hat{\beta}_0(s)$ and $\hat{\beta}(s)$; the fitted model is

$$\hat{f}(x) = \hat{\beta}_0 + \sum_{j=1}^q \hat{\beta}_j h_j(x).$$

The classification rule is given by $\text{sign}[\hat{f}(x)]$.

The 1-norm SVM has been successfully used in classification problem for cancer microarray data. To get a good fitted model $\hat{f}(x)$ that performs well on future data, we also need to select an appropriate tuning parameter s . In practice, people usually pre-specify a finite set of values of s that covers a wide range, then either use a separate validation data set or use cross-validation to select a value for s that gives the best performance among the given set. In Zhu (2004), the chapter illustrates that the solution path $\hat{\beta}(s)$ is piece-wise linear as a function of s (in the R^q space); it also proposes an efficient algorithm to compute the exact whole solution path $\{\hat{\beta}(s), 0 \leq s \leq \infty\}$, hence help us understand how the solution changes with s and facilitate the adaptive selection of the tuning parameter s . Under some mild assumptions, Zhu showed that the computational cost to compute the whole solution path $\hat{\beta}(s)$ is $O(nq \min(n, q)^2)$ in the worst case and $O(nq)$ in the best case.

1-norm SVM replaces the ridge penalty in 2-norm SVM with the L_1 -norm penalty on β , i.e., the lasso penalty (See Tibshirani 1996), and considers the optimization below:

$$\min_{\beta_0, \beta} \left[\sum_{i=1}^n [1 - y_i(\beta_0 + \sum_{j=1}^q \beta_j h_j(x_i))]_+ + \lambda \|\beta\|_1 \right], \quad (1.11)$$

which is equivalent Lagrangian version of the constrained optimization problem. The lasso penalty was first proposed by Tibshirani (1996) for regression problems, where the response y is continuous rather than categorical. It has also been used in Bradley (1998) for classification problems under the framework of SVMs. Knight (2000) and Fu (2004) studied the asymptotics for lasso-type estimators. Similar to the ridge penalty, the lasso penalty also shrinks the fitted coefficients $\hat{\beta}$'s towards zero, hence 1-norm SVM also benefits from the reduction in fitted coefficients' variances. Another property of the lasso penalty is that because of the L_1 nature of the penalty, making λ sufficiently large, or equivalently s sufficiently small, will cause some of the coefficients $\hat{\beta}_j$'s to be exactly zero. Thus the lasso penalty does a kind of continuous feature selection, while this is not the case for the ridge penalty in 2-norm SVM, where none of the $\hat{\beta}_j$'s will be equal to zero.

CHAPTER 2

Classification and Gene selection of Cancer Microarrays by ν -Ridge Regression

There has been a recent explosion in the use of microarray data for classification in a variety of diagnostic areas, see Golub (1999) and Hastie (1998). The prediction of the diagnostic category of a tissue sample from its expression array phenotype given the availability of similar data from tissues in classified categories is known as classification or supervised learning. In the context of gene expression data, for example, different tumor types (Golub *et al.* 1999; Ramaswamy *et al.* 2001), response to therapy (van't Veer *et al.* 2003). A challenge in predicting the diagnostic categories using microarray data is that the number of genes is much greater than the number of tissue samples available, and we assume only a subset of genes is relevant in distinguishing different classes. Selection of relevant genes for classification is known as feature selection, which is a primary goal in microarray data analysis. A small set of relevant genes is essential for the development of inexpensive diagnostic tests.

The support vector machine (SVM) is one of the leading methods that has been successfully applied to classification of the cancer diagnosis (Lee & Lee 2002, Mukherjee *et al.* 1999, and Ramaswamy *et al.* 2001). In two-class classification, the linear SVM fits a model

$f(x) = b_0 + x_{1 \times p} b_{p \times 1}$ that minimizes

$$\sum_{i=1}^n (1 - y_i f(x_i))_+ + \frac{\lambda}{2} \|b\|^2.$$

The classification decision is then made according to $\text{sign}[f(x)]$. Inspired by the gene variable selection, Zhu (2003) proposed a L_1 -norm SVM using lasso penalty. The model can select variables by shrinking the coefficients toward 0 and actually setting some of them to be exactly 0.

Besides the non-parametric classification methods, Zhu (2004) proposed a penalized logistic regression (PLR) model and addressed the estimator of probability of interest $p(x)$, where the $p(x) = P(\text{Class} = +1 | X = x)$ is the conditional probability of a point being in class $\{+1\}$, given the gene expression measurements x . PLR performs comparably to SVM in classification besides providing an estimator of the underlying probability. However, these three classification methods may lead to misclassification on unbalanced data due to equal penalty on the loss of each sample. Wang and Yang (2004) proposed a weighted SVM to deal with classification on unbalanced data. They tried to put a constant weight on the loss of samples from class $\{+1\}$ and another constant weight on the loss of samples from class $\{-1\}$ so that the classifier will not benefit the class with more training samples.

It is possible that each sample contributes differently to the final classification rule. To capture the characteristic, in the thesis, we propose a new optimization method called ν -ridge regression (ν -RR) method. It can optimize the generalized weight for each training sample instead of putting a pre-identified constant weight. We define an upper bound of the weight so that the model is very flexible and obtain the best performance on a new gene expression data set. We also give a new and efficient computational insight for computing coefficients and constant in the model.

Besides predicting the correct cancer class for a given tumor sample, a primary challenge in microarray cancer diagnosis is to identify the relevant genes with contribute most to the outcome. We apply three gene selection methods here, univariate ranking (UR) (Dudoit *et al.* 2002, Golub *et al.* 1999 and Zhu 2005) and recursive feature elimination (RFE) (Guyon *et al.* 2002). The comparison of SVM, PLR and ν -RR with external gene selection methods are done with two frequently studied microarray data sets: leukemia introduced by Golub

et al. (1999) and breast cancer prognosis data which appeared on Nature in 2002 by van't Veer *et al.*.

Our simulation results on the real data sets indicates ν -RR with RFE tends to select less genes with exceptional error rates than SVM, PLR and BMA. More detailed report of simulation appears in section 2.3. The formulation of ν -RR, the description of UR, RFE and computation are described in section 2.2.

2.1 Approach

2.1.1 ν -Ridge Regression

Weighted-SVM has been discussed for unbalanced 2-class classification problem. See Chew (2001). He puts a common weight on all training samples from one class and puts another common weight on the rest of training samples. In Wahba *et al.* (2005), the technical report states a modified penalized likelihood for weighted data. Basically, each simulated multivariate data point has two associated weights. In this section, we propose a new model with a generalized weight, which may be negative, on the loss of each sample, and try to extract the best classifier among all the weighted models.

When we use a linear classifier in a two-class microarray data classification problem, to assign the most likely group to the samples in test data, we come up with the idea of putting a penalty on the influence of each gene. This is one way to lower misclassification, since those samples which have very similar gene expression values will be labeled similarly. In another words, if the label of one training sample x_1 is $y_1 \in \{-1, +1\}$, and there is a test sample x_2 which is very close to x_1 in Euclidian distance sense. So x_2 is more likely from class y_1 as x_1 . Therefore, we hope the inner product $| \langle w, (x_1 - x_2) \rangle |$ will be as small as possible. Roughly speaking, we may control $\|w\|^2$ so that the samples in test data will be assigned to the same group as those similar sample tissues in training data. The standard SVM puts the same weight on the loss of each training sample. Here, we put a generalized weight on the loss of each sample and regard the weights as unknown parameters. Consider

the optimization problem below:

$$\max_{t_{1 \times n}} \min_w \left[\sum_{i=1}^n t_i (< w, x_i > - y_i) + \lambda \|w\|_2^2 \right], \quad (2.1)$$

subject to the condition

$$\sum_{i=1}^n t_i = 0, \quad 0 \leq \nu \leq 1, \quad \lambda \geq 0, \quad |t_i| \leq \nu, \quad \forall i = 1, 2, \dots, n. \quad (2.2)$$

Each t_i can be understood as the generalized weight for the loss function of the i th sample. The max – min forces each loss to approach 0. Since our optimization is actually a modification of the dual form of ridge regression, we name (2.1) and (2.2) as ν -ridge regression model (ν -RR). The constraint $\sum_{i=1}^n t_i = 0$ is a regularized condition. The parameters λ and ν are to be chosen. In practice, they are determined by cross-validation. We clearly state the selection process in section 2.2.

The classification function for ν -RR is given by $f(x) = \text{sign}(< w, x > + b)$ for the choice of λ and ν , where b is a constant.

2.1.2 Quadratic Programming

As is well-known, the microarray data typically has small sample size n but large dimension of gene variables p (in the thousands). In the model we proposed above, one of the important roles of the vector t is reducing the number of parameters from $p + 1$ to $n + 1$, which makes the optimization feasible. The objective function $L(w, b, t) = \sum_{i=1}^n t_i (< w, x_i > + b - y_i) + \lambda \|w\|_2^2$ is convex for w, b when the vector t is fixed. For any given $t_{1 \times n}$ satisfying the constraints, notice that the value of b is not a function of t . In fact, any value of b will not change the optimal solution for the whole optimization problem (2.1) and (2.2). So we propose a rule to get the value of b in section 2.3. To reduce the computation, we simplify the part $\min_{w, b} L(w, b, t)$ by getting the score equation for w , whose expression is given by

$$\frac{\partial L(w, b, t)}{\partial w} = 0.$$

That leads to

$$2\lambda w + \sum_{i=1}^n (t_i x_i) = 0. \quad (2.3)$$

Therefore, the optimization problem becomes searching for the weight vector which

$$\max_{t_{1 \times n}} \left[-\lambda \|w\|_2^2 - \sum_{i=1}^n t_i y_i \right], \quad (2.4)$$

subject to the condition

$$\sum_{i=1}^n t_i = 0, \quad |t_i| \leq \nu, \quad \forall i = 1, 2, \dots, n.$$

Plug (2.3) into (2.4), then we have

$$\min_t \left[\frac{1}{4\lambda} \sum_{i=1}^n \sum_{k=1}^n t_i t_k < x_i, x_k > + \sum_{i=1}^n t_i y_i \right], \quad (2.5)$$

subject to the condition

$$\sum_{i=1}^n t_i = 0, \quad |t_i| \leq \nu, \quad \forall i = 1, 2, \dots, n. \quad (2.6)$$

Once λ and ν are determined, this is a standard quadratic programming problem, where t is a $1 \times n$ vector with typically small n value. Lately there is a contributed package in R, Kernlab, which can make the computation even easier.

2.1.3 Computation of w and b

For any chosen values of λ and ν , w is given by the expression $w = -t_{1 \times n} x_{n \times p} / 2\lambda$. Obviously, the optimization (2.5) and (2.6) does not provide the optimal solution of b . Thinking of the fact that b is playing a role for samples with $y_i = +1$ and $< w, x_i >$ being less than 1, or for samples with $y_i = -1$ and $< w, x_i >$ being greater than -1 , we consider two groups of samples,

$$S_1 = \{i \in 1, 2, \dots, n | y_i = +1\}, \quad S_2 = \{i \in 1, 2, \dots, n | y_i = -1\}.$$

There exists a constant that is just large enough for all samples from group S_1 being correctly classified, and there exists another constant that is just small enough for samples from group S_2 being correctly classified. For a new sample to be fairly classified, we define b as the average value of the two constants. That is

$$b = (-\min_{i \in S_1} < w, x_i > - \max_{i \in S_2} < w, x_i >)/2. \quad (2.7)$$

This b can avoid the over-fitting on the training data. Actually it works very well on two real gene expression data sets.

2.1.4 Feature selection

Besides predicting the correct cancer class for a given tumor sample, another primary challenge in microarray cancer diagnosis is to identify the relevant genes which contribute the most to the classification. Among all external gene selection methods, univariate ranking (UR) and recursive feature elimination (RFE) are the most often used. Golub *et al.* (1999) first introduced UR for each gene in the two class classification problem. The criterion is defined as:

$$s_j = \frac{\overline{x_j^+} - \overline{x_j^-}}{\sigma_j^+ + \sigma_j^-}, \quad (2.8)$$

where $\overline{x_j^+}$ and σ_j^+ indicate the average and standard deviation of the gene expression values of gene j for all samples from class $+1$. Similarly, $\overline{x_j^-}$ and σ_j^- indicate the average and standard deviation of the gene expression values of gene j for all samples from class $\{-1\}$. Genes that give the most positive values are supposed to be most correlated with class $\{+1\}$, and genes that give the most negative values are supposed to be most correlated with class $\{-1\}$. This ranking criteria implicitly assumes orthogonality among the genes, because each s_j is computed with information about a single gene and does not take into account mutual information between genes. Later on Dudoit *et al.* (2002) used the ratio of between-group to within-group sum of squares (BSS/WSS) to determine the initial gene order for the multi-class case.

Opposite to the ranking method, RFE recursively removes genes based upon the absolute magnitude of the hyperplane elements. For the linear kernel, given microarray data with p genes per sample, a classification method will output w , which is a vector with p components, each corresponding to a particular gene. The absolute magnitude of each element $|w_j|$ determines its importance in classifying a sample. The idea behind RFE is to eliminate elements of w that have small magnitude. This screening procedure is iterative until a desired number of genes is obtained.

2.2 Results

In this section, we fit ν -RR to leukemia data set (Golub *et al.* (1999)) and breast cancer prognosis data set (van't Veer *et al.* (2002)). UR and RFE are applied to reduce the number of genes at each step. For the UR method, we first use (2.8) to compute s_j for all genes and rank the genes in the descending order of $|s_j|$. Then we apply an iterative procedure that goes as following: start with fitting ν -RR using all current genes, next remove 10% of the genes in the model that are at the bottom 10% of the ranking, then refit the model with the remaining genes, and iterate. For the RFE applied in ν -RR, according to the recursive procedure, at the k th step of the iteration, we fit the model with the remaining k_p genes,

$$f_k(x) = \langle w, x \rangle + b_k = b_k + \sum_{j=1}^{k_p} w_j x_j ,$$

and eliminate the genes with the overall smallest 10% $|w_j|$ values. So at each step of RFE we may eliminate different number of genes. Notice that once the gene is removed from the model, it will never come back to the model again. So the parameters used in the RFE at the very beginning have a significant influence on the whole procedure of keeping genes at each step. The number of genes in the final model is selected by cross-validation and the performance of the final model is evaluated on the test samples.

Before applying ν -RR, we standardize each sample as is usually done in microarray studies, see Dudoit *et al.* (2002) and Guyon *et al.* (2002), so that the mean and standard deviation of the expression levels are 0 and 1, respectively.

2.2.1 Choosing λ and ν

We randomly divide the training data into 10 groups. At each step we use one group as the cross-validation dataset and the other 9 groups are used as training data where the classifier comes from. So each sample in the training data will be regarded as cross-validation set for only one time and will be used as training data 9 times. The RFE method is more sensitive to the parameters, so we determine (λ, ν) by applying RFE and use the same parameter setting in UR. When searching for λ and ν used in the final model of ν -RR, we follow the rule below: among the chosen λ and ν which result in the minimal 10-fold CV-error

Table 2.1. Parameter selection.

λ	ν	CV_1	RFE	UR
25	0.1	2/38	37	41
	0.3	3/38		
	0.5	4/38		
	0.7	3/38		
	0.9	2/38		
100	0.1	2/38	20	42
	0.3	2/38	43	35
	0.5	2/38	30	40
	0.7	3/38	26	40
	0.9	2/38		
200	0.1	4/38		
	0.3	2/38	47	34
	0.5	2/38	37	41
	0.7	4/38		
	0.9	3/38		

on the whole training data without gene selection, we want the parameters with minimal summation of the CV-error across all iterations in gene elimination.

$$(\lambda, \nu) = \arg_{\arg(\min_{\lambda \geq 0, 0 \leq \nu \leq 1} CV_1)} (\min \sum_{k=1}^l Misc_k) ,$$

where l is total number of iterations, CV_1 means the 10-fold CV-error using all genes in the training data, and $Misc_k$ means the number of misclassification based on cross-validation at the k th iteration. We care about the CV-error without gene selection due to the disadvantage of RFE method. If the parameters do not perform well at the beginning, there is a bigger chance that it will not do a good job in the gene elimination procedure. In each iteration, we can update the ν value to fit the data with remaining genes. Through this parameter selection, we can confidently claim that the parameters λ and ν work generally well at each step. In practice, however, we can not try all possible values of λ and ν . Table 1 shows the numerical parameter selection process for the leukemia data based on 12 results. The minimal summation of the CV-error happens at initial value $\lambda = 25$ and $\nu = 0.9$, hence $(25, 0.9)$ are chosen as the regularization parameters in ν -RR model to fit the entire training data. Recall that we can adjust the ν value later on.

In Table 2.1, RFE column is the total number of 10-fold misclassifications across all iterations in gene elimination using RFE. Similarly, UR column is the total number of 10-fold misclassifications across all iterations in gene elimination using UR.

Table 2.2. Comparison of leukemia classification methods

Method	10-fold CV-error	Test error	m
SVM UR	2/38	3/34	22
PLR UR	2/38	3/34	16
ν -RR UR	0/38	2/34	7
SVM RFE	2/38	1/34	31
PLR RFE	2/38	1/34	26
ν -RR RFE	0/38	0/34	11

Table 2.3. Published results of leukemia classification methods

Method	10-fold CV-error	Test error	m
Golub et al	3/38	4/34	50
Tibshirani et al	2/38	2/34	21
L_1 norm SVM	2/38	2/34	17

2.2.2 Leukemia data

Leukemia (Golub *et al.* 1999) consists of 38 training samples and 34 test samples of two types of acute leukemias, acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL). Each sample contains 7129 gene expression levels. 10-fold cross validation is used as a criterion to determine parameters λ and ν . As in table 2.1, $\lambda = 25$ and $\nu = 0.9$ are used in the ν -RR model and the performance of the classifier is evaluated on the test data. For the classification method SVM and PLR, we follow the results from Zhu (2004). We can see that when using the same set of genes (i.e. using UR), ν -RR yields less significant genes with reduced CV-error and test error than SVM and PLR. When using RFE, ν -RR performs the best with least relevant genes and smallest CV-error and test error among the three listed methods. ν -RR with UR obtained a more manageable set of genes than ν -RR with RFE, at the cost of one more test error (which increases the test error from 0 to 1).

The minimal cross-validation error for ν -RR occurs at 7 genes and 11 genes in the UR and RFE, respectively. The list of the 7 genes chosen by ν -RR UR are identified as 1745, 1834, 2020, 2288, 3320, 4847 and 5772. The list of 11 genes chosen by ν -RR RFE are identified as numbers 1249, 1779, 1796, 1834, 1846, 1882, 2288, 2402, 4847, 5039 and 5950. The results of Golub *et al.* (1999), Tibshirani *et al.* (2002) and Zhu (2003) are summarized in Table 2.3. m is the number of selected genes.

In Table 2.2, ν -RR sacrifice two test errors by eliminating 4 genes. It is not necessary that the 7 genes selected by ν -RR with UR are all in the set of 11 genes selected by ν -RR with RFE.

Table 2.4. Comparison of breast cancer classification methods.

Method	Test error	m
Van't Veer <i>et al.</i>	2/19	70
Yeung	3/19	6
ν -RR RFE	2/19	2
ν -RR UR	4/19	2

2.2.3 Breast cancer prognosis data

The breast cancer prognosis data (van't Veer *et al.* (2002) and Shieh (2004)) consists of 97 primary breast tumor samples hybridized to cDNA arrays consisting of 24481 gene expression levels. The two categories are: the good prognosis group (patients who remained disease free for at least 5 years) and the poor prognosis group (patients who developed distant metastases within 5 years). We picked 4918 significantly regulated genes (at least a 2-fold difference and p-value < 0.01 in more than 3 samples) among 24481 genes. We further removed two samples with missing values at the 4918 gene expression levels. Therefore, the breast cancer prognosis data in this chapter consists of 76 training samples and 19 test samples across 4918 genes.

When we apply the ν -RR with RFE to the 76 training samples each with 4918 genes, a cross-validation set formed by 4 samples from the good group and 3 samples from the poor group was randomly taken from the training data to determine the tuning parameters. For each λ value, we choose that $\nu \in (0, 1)$ which leads to the smallest CV-error rate and evaluate the performance of the classifier on the test samples. Due to the randomness of the cross validation set, we repeated the programming 10 times for each pair of λ and ν , and take the average of number of misclassifications across the 10 simulations. The average number of misclassifications at the k th iteration in gene selection was used as $Misc_k$ in the parameter selection criterion. Then our ν -RR method is trained on all 76 training samples to get the classifier. When applying RFE for gene selection, we found out that any $\lambda > 5$ will provide the same gene selection result. Genes identified as 1865 and 2294 are selected with 2 classification error on the test data. The result varies as λ decreases from 5. We compare the published results of Van't Veer *et al.* (2002) and BMA Yeung (2004), to our selected genes and the corresponding classification errors. The comparison is summed up in table 2.4.

There is only one gene expression level identified as AL080059 in the above 4 sets of selected genes.

2.3 Discussion

Among all the above classification methods with UR, ν -RR found 7 genes that distinguished acute myeloid leukemia (AML) from acute lymphoblastic leukemia (ALL) with a lower error rate than SVM and PLR employed in Zhu (2004). That is to say, ν -RR can reduce the test error with even smaller set of genes. This may imply the weights used in ν -RR capture the interactions among genes better than SVM and PLR models.

Comparing the results of classification methods with RFE, we found that ν -RR dramatically improves all three criterion: CV-error, test error and number of genes, based on the existing gene selection results on leukemia and breast cancer prognosis data. Combined with either UR or RFE, ν -RR obtained a smaller set of related genes without any cost of CV-error and test error.

We can tell from Table 2.2 that ν -RR with UR tends to select less (or same amount of) genes than ν -RR with RFE, but UR leads to a higher test error. It is a trade-off. Through the simulation, we can tell that RFE is more sensitive to the values of regularization parameters than UR. So more attention should be focused in choosing the parameters in RFE than in UR. The set of genes chosen by ν -RR with UR in leukemia data has 3 genes in common with the set of genes chosen by ν -RR with RFE. In breast cancer prognosis data, they have one common selected gene named AL080059, which is also in the list of van't Veer *et al.* (2002) and Yeung (2004).

It is also worth to note that in the simulation, we can update the regularization parameters to get the picture of the overall performance at all iteration steps rather than keep the best performing parameters at an individual iteration step, as we did in the breast cancer prognosis data. However, we can not try out all λ values. In our approach, the summation of the misclassifications over all steps is one criterion for the parameters, so the model with the chosen parameter λ guarantees a good overall performance on the cross-validation data, which is consistent with our simulation results in the leukemia data and breast cancer data.

Here we propose a new method inspired by ridge regression for the two class classification. In going from the two class to the multi-class classification, the one-vs-rest scheme is often used: given K classes, the problem is divided into a series of K one-vs-rest problems, and each one-vs-rest problem is addressed by a different class-specific ν -RR classifier; then a new sample takes the class corresponding to the classifier with largest real valued output. This is one approach for the multi-class classification. But how to extend the ν -RR directly to multi-class case with less work is called for and I am working on it. For general information about multi-class classification, see Dietterich (1991), Ramaswamy (1998) and Lee and Lee (2002).

2.4 Conclusion

We have proposed a ν -ridge regression (ν -RR) model for the two class microarray cancer diagnosis classification. The simulation results on the real leukemia data and breast cancer data show that when using the same set of genes (using UR), ν -RR identifies less significant genes with smaller CV-error and test error than SVM and PLR. When using the recursive feature elimination method to select relevant genes, ν -RR works perfect on leukemia data with 0 CV-error and 0 test error, and it gains comparable test error on breast cancer data. What is more, ν -RR with RFE selects less genes than SVM, PLR, and other published results. This good property makes the set of chosen genes more manageable. Therefore, we claim ν -RR is a good classification method for two class microarray data.

CHAPTER 3

Density Curve Estimation and Classification of Microarray by Conditional Logistic Regression Model

The support vector machine (SVM) has been successfully applied to microarray cancer diagnosis problems. Khan (2001) proposed a classification method using gene expression profiling and artificial neural networks. However, one weakness of the SVM and Khan is that given a tumor sample, it only predicts a cancer class label but does not provide any estimate of the underlying probability. The penalized logistic regression has the advantage of additionally providing an estimate of the underlying probability of being assigned to a class, but it does not offer any estimate for the probability of the class y conditional on an individual gene variable. We propose the conditional logistic regression (CLR) model, which is an alternative for the microarray cancer diagnosis classification, for the underlying probability of the response given any gene variable. Meanwhile, since the gene selection purpose as a primary goal in microarray cancer diagnosis, we propose a new method called modified univariate ranking (MUR) as a new choice for dimension reduction.

We show that when applied on a microarray data for classification, CLR performs comparable to those classification methods, e.g. SVM, PLR and BMA, but CLR has the advantage

of providing the probability of the class y conditional on any individual gene variable. Empirical results on Leukemia and Breast Cancer data indicate the CLR combined with one of the gene selection methods (MUR, BSS/WSS or RFE) tends to perform superior to SVM and PLR on both CV-error and test error rate.

The support vector machine (SVM) is one of the leading methods that has been successfully applied to classification of the cancer diagnosis. See Lee & Lee (2002), Mukherjee *et al.* (1999), and Ramaswamy *et al.* (2001). In two-class classification, the linear SVM fits a model $f(x) = b_0 + x_{1 \times p} b_{p \times 1}$ that minimizes

$$\sum_{i=1}^n (1 - y_i f(x_i))_+ + \frac{\lambda}{2} \|b\|^2.$$

The classification decision is then made according to $\text{sign}[f(x)]$. However, one weakness of the SVM is that it only estimates $\text{sign}[p(x) - 0.5]$, but doesn't offer the probability of interest $p(x)$ or $p_i(x)$, where the $p(x) = P(\text{Class} = 1 | X = x)$ is the conditional probability of a point being in class 1 given the gene measurements x , and $p_i(x) = P(\text{Class} = 1 | \text{ith gene expression level})$. Recently, Zhu (2005) proposed a penalized logistic regression (PLR) classifier and the underlying probability $p(x)$, but it still lacks the investigation on the estimation of $p_i(x)$.

In this chapter, we use CLR model to get an insight of the $p_i(x)$ and take the $p_i(x)$ into consideration for classifier. The classification rule is given by $\text{sign}[\frac{\prod_i p_i}{\prod_i (1-p_i)} - a]$, where a is a cut-off value depending on the gene variables used in CLR. The CLR not only performs as well as SVM, PLR and BMA in two-class classification, but can provide an estimate of the probability of interest $p_i(x)$.

Maximum likelihood and the Newton-Raphson algorithm is the traditional way to solve CLR numerically. However, the computation in microarray data is tedious. Instead, we use a sequential minimal optimization (SMO) algorithm (Platt (1998)) to solve CLR in this chapter. SMO was first proposed in Keerthi *et al.* (2002) for PLR model for two-class classification; we modify it to be applicable to the CLR model.

Besides predicting the correct cancer class for a given tumor sample, a primary challenge in microarray cancer diagnosis is to identify the relevant genes that contribute most to the outcome. We apply three gene selection methods here, univariate ranking (UR) (Dudoit *et*

al. (2002), Golub *et al.* (1999) and Zhu (2004)), recursive feature elimination (RFE) (Guyon *et al.* (2002)), and the ratio of between-group to within-group sum of squares (BSS/WSS) (Dudoit *et al.* (2002)). Furthermore, we propose a modified univariate ranking (MUR) to improve the classification performance and eliminate genes. The comparison of SVM, PLR and CLR with some external gene selection method are done with two frequently studied microarray data sets.

Our simulation results on the real data sets indicates CLR with MUR tends to select less genes with exceptional error rates than SVM, PLR and BMA with some gene selection method. More detailed report of simulation is in section 3.2. The formulation of CLR, the description of UR, RFE, BSS/WSS and MUR, and the cut-off value are described in section 3.1.

3.1 Approach

In standard two class classification problems, we are given a set of training data (x_1, y_1) , (x_2, y_2) , ..., (x_n, y_n) , where the input $x_i = (x_{i1}, x_{i2}, \dots, x_{ip})$, the output y_i is qualitative and assumes values in $\{+1, -1\}$. We wish to find a classification rule from the training data, so that when given a new input x , we can assign a class label from $\{+1, -1\}$ to it. Meanwhile, we wish to understand more on the relationship between the binary response and each gene variable. The relationship is defined by the probability of y conditional on x_j with logistic regression model. A statistic based on the conditional probability is accompanying a tissues sample and is used as our cut-off value for classification.

3.1.1 Conditional Logistic Regression Formulation

Usually it is assumed that the training data are an independently and identically distributed sample from an unknown probability distribution. To estimate the curves of the conditional probability, we think of the parametric approach and use the most prominent logistic models, which has the form

$$P(y|x_1) = \frac{1}{1 + \exp^{-yf_1(x_1)}} ,$$

$$\begin{aligned}
P(y|x_{.2}) &= \frac{1}{1 + \exp^{-y f_2(x_{.2})}} , \\
&\vdots \\
P(y|x_{.p}) &= \frac{1}{1 + \exp^{-y f_p(x_{.p})}} ,
\end{aligned}$$

where $x_{.j}$ represents the j th gene expression level of the tissues sample $x_{.}$, and $f_j(x) = b_{j0} + b_{j1}x$, $j = 1, 2, \dots, p$.

Logistic regression models are usually fit by maximum likelihood. Given the training set, the negative log-likelihood is

$$-\sum_{i=1}^n \sum_{j=1}^p \log P(y = y_i | x_{ij}) = \sum_{i=1}^n \sum_{j=1}^p \log(1 + \exp^{-y_i f_j(x_{ij})}) .$$

We hope Two similar gene expression value measured on a same gene variable would lead to very close conditional probability. Therefore, the coefficients b_{j1} for each gene variable are better to be reasonably small. Additionally, to avoid the special case in which some optimal b_{j1} 's will be infinite, we consider the negative log-likelihood with the L-2 penalty term, which is

$$\min_{b_{j0}, b_{j1}} \left[C \sum_{i=1}^n \sum_{j=1}^p \log(1 + \exp^{-y_i f_j(x_{ij})}) + \frac{1}{2} \sum_{j=1}^p b_{j1}^2 \right] . \quad (3.1)$$

With the gene expression arrays, it is typically that $p \gg n$. Notice we have $2p$ parameters in the optimization. Since p is often in thousands, and motivated by the logistic assumption, we come up with SMO algorithm.

3.1.2 Algorithm

The spirit of the conditional logistic regression SMO algorithm follows the two class SMO algorithm for penalized logistic regression by Keerthi *et al.* (2002). We extend the algorithm to our case.

To minimize (3.1), we rewrite it as

$$\min_{b_{j0}, b_{j1}} \left[\frac{1}{2} \sum_{j=1}^p b_{j1}^2 + C \sum_{i=1}^n \sum_{j=1}^p g(\xi_{ij}) \right] , \quad (3.2)$$

subject to the conditions

$$\xi_{ij} = -y_i(b_{j0} + b_{j1}x_{ij}) , \quad (3.3)$$

$$g(\xi_{ij}) = \log(1 + e^{\xi_{ij}}), \quad \forall i, j, \quad (3.4)$$

where C is the regularization parameter. The Lagrangian dual form for the problem is

$$L = \frac{1}{2} \sum_{j=1}^p b_{j1}^2 + C \sum_{i=1}^n \sum_{j=1}^p g(\xi_{ij}) + \sum_{i=1}^n \sum_{j=1}^p \alpha_{ij} [-\xi_{ij} - y_i(b_{j0} + b_{j1}x_{ij})],$$

where the α_{ij} 's are the Lagrangian dual multipliers. By the KKT optimality conditions, we have

$$\begin{aligned} \frac{\partial L}{\partial b_{j1}} &= b_{j1} + \sum_{i=1}^n \alpha_{ij}(-y_i)x_{ij} = 0, \quad \forall j, \\ \frac{\partial L}{\partial b_{j0}} &= \sum_{i=1}^n \alpha_{ij}(-y_i) = 0, \quad \forall j, \\ \frac{\partial L}{\partial \xi_{ij}} &= Cg'(\xi_{ij}) - \alpha_{ij} = 0, \quad \forall i, j, \end{aligned}$$

so that b_{j1} and ξ_{ij} can be expressed as function of the α_{ij} 's, which is

$$b_{j1} = \sum_{i=1}^n \alpha_{ij} y_i x_{ij}, \quad \xi_{ij} = (g')^{-1}\left(\frac{\alpha_{ij}}{C}\right).$$

Define $\delta = \frac{\alpha_{ij}}{C}$ and $G(\delta) = \delta \xi_{ij} - g(\xi_{ij})$, so obviously

$$G(\delta) = \delta \log(\delta) + (1 - \delta) \log(1 - \delta).$$

Therefore,

$$\frac{\partial G}{\partial \delta} = \delta \frac{\partial \xi_{ij}}{\partial \delta} + \xi_{ij} - g'(\xi_{ij}) \frac{\partial \xi_{ij}}{\partial \delta} = (g')^{-1}(\delta).$$

So the dual form of optimization problem (3.2), (3.3) and (3.4) becomes

$$\max_{\alpha_{ij}} \left[-\frac{1}{2} \sum_{j=1}^p b_{j1}^2 - C \sum_{i=1}^n \sum_{j=1}^p G\left(\frac{\alpha_{ij}}{C}\right) \right], \quad (3.5)$$

subject to the condition

$$\sum_{i=1}^n \alpha_{ij} y_i = 0, \quad \forall j = 1, 2, \dots, p, \quad (3.6)$$

where $b_{j1} = \sum_{i=1}^n \alpha_{ij} y_i x_{ij}$, $\forall j = 1, 2, \dots, p$. Rewrite the above in the dual form, which is

$$\max_{\phi_j} \min_{\alpha_{ij}} \bar{L} = \frac{1}{2} \sum_{j=1}^p b_{j1}^2 + C \sum_{i=1}^n \sum_{j=1}^p G\left(\frac{\alpha_{ij}}{C}\right) - \sum_{j=1}^p \phi_j \sum_{i=1}^n \alpha_{ij} y_i.$$

Denote

$$F_{ij} = b_{j1}x_{ij} = x_{ij} \sum_{i=1}^n \alpha_{ij} y_i x_{ij}, \quad H_{ij} = F_{ij} + y_i G'(\frac{\alpha_{ij}}{C}),$$

by KKT conditions, we have

$$\begin{aligned} \frac{\partial \bar{L}}{\partial \alpha_{ij}} &= y_i F_{ij} + G'(\frac{\alpha_{ij}}{C}) - \phi_j y_i \\ &= (H_{ij} - \phi_j) y_i \\ &= 0, \quad \forall i, j. \end{aligned}$$

So the optimal solution vector $\alpha = (\alpha_{11}, \dots, \alpha_{n1}, \dots, \alpha_{1p}, \dots, \alpha_{np})'$ has to satisfy $H_{ij} = \phi_j$, $\forall i = 1, 2, \dots, n$, and for each $j = 1, 2, \dots, p$. That is to say, if we denote

$$i_{up}(j) = \arg \max_i H_{ij}, \quad i_{low}(j) = \arg \min_i H_{ij}, \quad \forall j,$$

we must have

$$H_{i_{up}(j)j} = H_{i_{low}(j)j}, \quad \forall j. \quad (3.7)$$

For some $j \in 1, 2, \dots, p$, suppose (i_1, i_2) satisfies $H_{i_1j} \neq H_{i_2j}$, we define

$$\tilde{\alpha}_{i_1j} = \alpha_{i_1j} + \frac{t_j}{y_{i_1}}, \quad \tilde{\alpha}_{i_2j} = \alpha_{i_2j} - \frac{t_j}{y_{i_2}}, \quad \tilde{\alpha}_{ij} = \alpha_{ij}, \quad \forall i \neq i_1, i_2, \forall j, \quad (3.8)$$

and let

$$\psi(t_j) = h_j(\tilde{\alpha}),$$

where $h_j = \frac{1}{2}b_{j1}^2 + C \sum_{i=1}^n G(\frac{\alpha_{ij}}{C})$. Therefore,

$$\begin{aligned} \psi'(t_j) &= F_{i_1j} + \frac{1}{y_{i_1}} G'(\frac{\alpha_{i_1j}}{C}) - F_{i_2j} - \frac{1}{y_{i_2}} G'(\frac{\alpha_{i_2j}}{C}) \\ &= H_{i_1j} - H_{i_2j}, \end{aligned}$$

where H_{i_1j} and H_{i_2j} are evaluated at $\tilde{\alpha}$. Since $H_{i_1j} - H_{i_2j} \neq 0$ at $t_j = 0$ for some j , a decrease in ψ is possible by choosing t_j suitably away from 0.

Through the procedure, we can tell that $b_{j0} = -\phi_j$, $\forall j = 1, 2, \dots, p$. Therefore, at the optimal solution α , we have $-b_{j0} = H_{i_{up}(j)j} = H_{i_{low}(j)j}$, $\forall j = 1, 2, \dots, p$. Since, in numerical solution, it is usually not possible to achieve optimality exactly, there is a need to

define approximate optimality conditions. We denote τ as our upper bound for how much we can put up with. The exact KKT condition can be replace by

$$H_{i_{up}(j)j} - H_{i_{low}(j)j} \leq \tau .$$

As a consequence, we may define $-b_{j0} = \frac{H_{i_{up}(j)j} + H_{i_{low}(j)j}}{2}$.

The SMO algorithm can now be described as following:

1. Choose $\alpha^0 = (\alpha_{.1}, \alpha_{.2}, \dots, \alpha_{.p})'$ satisfying conditions (3.6). One possible initial vector is

$$\begin{aligned} \alpha_{ij} &= \frac{C}{m_1}, \text{ i with } y_i = +1, \forall j, \\ \alpha_{ij} &= \frac{C}{m_2}, \text{ i with } y_i = -1, \forall j, \end{aligned}$$

where m_1 and m_2 denote the number of training examples in $\{+1\}$ and $\{-1\}$, respectively.

Set $r = 1$.

2. If $\alpha_{.j}^r$ satisfies (3.7) for some $j \in \{1, 2, \dots, p\}$, stop. If (3.7) does not hold for any $j \in \{1, 2, \dots, p\}$, let

$$t_j^* = \arg \min \psi(t_j), \alpha_{i_{low}(j)j}^{r+1} = \alpha_{i_{low}(j)j}^r - \frac{t_j^*}{y_{i_{low}}}, \alpha_{i_{up}(j)j}^{r+1} = \alpha_{i_{up}(j)j}^r + \frac{t_j^*}{y_{i_{up}}} . \quad (3.9)$$

3. Update $\alpha_{.j}^{r+1}$ according to the above formulation. If necessary, go back to step 2.

As stated in Keerthi (2002), the value of C should not be large. For our simulated microarray data sets, when $C > 0.5$, the initial value of α is out of the domain of log function. Recall that $b_{j0} = \frac{-H_{i_{low}(j)j} - H_{i_{up}(j)j}}{2}$ and $b_{j1} = \sum_{i=1}^n \alpha_{ij} y_i x_{ij}$, so the probability path of class y conditional on the j th gene expression level $x_{.j}$ is $P(y|x_{.j}) = \frac{1}{1 + \exp^{-y(b_{j0} + b_{j1}x_{.j})}}$.

3.1.3 Feature Selection

Besides predicting the correct cancer class for a new tumor sample, another challenge in microarray cancer diagnosis is to identify the relevant genes which contribute the most to the classification. In our study, we used the ratio of between-group to within-group sum of squares (BSS/WSS) (Dudoit *et al.* (2002)) and recursive feature elimination (RFE) to determine the initial gene order.

About the BSS/WSS, intuitively, genes with relatively large variation between classes and relatively small variation within classes are the most likely candidates for relevant genes.

BSS/WSS is a univariate gene selection method in which genes with large BSS/WSS ratios are good candidate relevant genes. For a gene j , let D_{ij} denote the expression level of gene j under sample i , \bar{D}_{kj} denote the average expression level of gene j over samples in class $k \in \{+1, -1\}$, and $\bar{D}_{.j}$ denote the average expression level of gene j over all samples. The BSS/WSS ratio for gene j is defined as

$$\frac{BSS(j)}{WSS(j)} = \frac{\sum_i \sum_k I(y_i = k) (\bar{D}_{kj} - \bar{D}_{.j})^2}{\sum_i \sum_k I(y_i = k) (D_{ij} - \bar{D}_{kj})^2}. \quad (3.10)$$

We compute the BSS/WSS ratio for each of the p genes and order the genes in descending order of the BSS/WSS ratio.

The gene selection method RFE depends on the classification method. We rank the genes in descending order of the absolute value of b_{j1} . The coefficient b_{j1} reflects how the change on gene j expression level will affect the conditional probability. The bigger the $|b_{j1}|$, the more effective the gene j is to influence the assignment of class. Therefore, gene j is more possible to be a relevant gene for the classification.

Aiming at gene selection, we apply BSS/WSS and RFE. As another choice, we propose a new univariate gene ranking method in terms of their classification performance, called modified univariate ranking (MUR). The criterion is defined as:

$$\rho_j = \frac{\beta |\bar{x}_j^+ - \bar{x}_j^-| + (1 - \beta) |\sigma_j^+ - \sigma_j^-|}{\sigma_j^+ + \sigma_j^-}, \quad (3.11)$$

where $\beta \in [0, 1]$, \bar{x}_j^k and σ_j^k indicate the average and standard deviation of the gene expression levels of gene j for all samples of class $k \in \{+1, -1\}$. Our ranking statistics is motivated by the univariate ranking statistics

$$s_j = \frac{|\bar{x}_j^+ - \bar{x}_j^-|}{\sigma_j^+ + \sigma_j^-}.$$

It was introduced by Dudoit *et al.* (2002). The s_j mainly takes care of the location difference between the two class for any gene j and can not detect any difference when the mean of the two classes are equal but the standard deviations are totally different. However, s_j considers both the location and scale difference even when one difference is hard to detect. Through our simulation study, we conclude that ρ_j fits some data, including Leukemia data, better than UR and BSS/WSS in selecting groups of genes for classification.

3.1.4 Cut-off statistics a_p

The probability path of class y conditional on the j th gene expression value $x_{.j}$ is

$$P(y|x_{.j}) = \frac{1}{1 + \exp^{-y(b_{j0} + b_{j1}x_{.j})}} .$$

Outline of a_p with iterative RFE for gene selection:

- Input: training set D with p genes and n samples
- Pre-processing step: use all genes for classification and rank p genes by the value of $|b_{j1}|$.
- Determine a_m : at j th iteration step, let q_j denote the ordered list consisting of $m = p \times 90\%^{j-1}$ top ranked genes. We compute the statistics

$$\eta_i = \frac{\prod_{j \in q_j} P(y = +1|x_{ij})}{\prod_{j \in q_j} P(y = -1|x_{ij})}$$

for each tissue sample in the training set, $i = 1, 2, \dots, n$.

Our classification rule is based on the comparison of the η_i value and a cut-off value a_m : assign a new sample $x_{.}$ with gene expression values $x_{.j}, j = 1, 2, \dots, p$ to class $\{+1\}$ if $\eta_i > a_m$ and assign $x_{.}$ to $\{-1\}$ otherwise. Consider two groups of indices,

$$G_1 = \{i \in 1, \dots, n | y_i = +1\}, \quad G_2 = \{i \in 1, \dots, n | y_i = -1\} .$$

Ideally, we wish $\max_{i \in G_2} \eta_i < \min_{i \in G_1} \eta_i$ so that the cut-off value a_m which can classify all training samples to the right class exists. In practice, we can find a value a_m such that there will be least misclassification on the training set. The classifier $\eta_i > a_m$ is applied on the test samples and the performance of the selected m genes is evaluated by the test error.

- Iterate the the previous steps until $m = 1$.

Output: the probability distribution of the class y conditional on each gene variable, selected genes at each iteration (q_m) associated with the cut-off value a_m , training error(cv-error if cross-validation is done) and test error.

When apply the BSS/WSS to select a_m , we rank all p genes by the $BSS(j)/WSS(j)$, so we change the gene set q_m at each iteration. Everything else is exactly the same as the steps for RFE.

Table 3.1. Comparison of classification methods for breast cancer prognosis data.

method	training error	Test error	m
van't Veer et al	15/76	2/19	70
BAM	16/76	3/19	6
CLR MUR($\beta = 0.8$)	16/76	3/19	1
CLR UR	16/76	3/19	1
CLR BSS/WSS	16/76	3/19	1
CLR RFE	16/76	3/19	1

Table 3.2. Output of CLR on breast cancer data.

	b_{j0}	b_{j1}
AL080059	0.013	-1.293

3.2 Results

3.2.1 Breast cancer prognosis data

Refer to the description of breast cancer prognosis data in section 2.2.3. Using all 4918 genes and $C = 0.3$, our CLR algorithm provides the the probability distribution of the class y conditional on each individual gene variable. For the breast cancer diagnosis data, we use all 76 training samples and get the ratio statistic corresponding to each sample. Our a_m to determine the classifier for different set of selected gene is the one which separates the training samples with the lowest error rate. If in the case there are several such a_m values, we choose their median for the classifier.

Our iterative CLR algorithm combined with BSS/WSS or RFE produced 3 classification errors out of 19 test samples with 1 selected gene, AL080059. van't Veer *et al.* (2002) reported 2 classification errors on the test set using 70 relevant genes. Yeung *et al.* (2005) reported 3 classification errors out of 19 test samples using 6 selected genes. Furthermore, our selected gene, AL080059, is the only common gene in the 70 selected gene set by van't Veer and 6 selected gene set by Yeung. By applying our CLR method for classification, the single AL080059 performs similarly to those two sets of genes. The gene AL080059 is ranked top 1 by BSS/WSS and it has the biggest absolute value of the coefficient b_{j1} . To some degree, the result confirms the consistency of CLR in the sense that top rank BSS/WSS gene also has a top rank coefficient as we wish.

The result in Table 3.1 and Table 3.2 is from $C = 0.3$, $a_1 = 1.29$. The selected gene by CLR with MUR, UR, BSS/WSS or RFE is the gene identified as AL080059 which is also

Table 3.3. Comparison of CLR on leukemia data.

Method	training error	Test error	No.of genes
CLR MUR	0/38	1/34	6
CLR UR	0/38	2/34	8
CLR BSS/WSS	0/38	1/34	9
CLR RFE	0/38	1/38	9

Table 3.4. Comparison of leukemia classification methods.

Method	CV-error	Test error	No. of genes
Golub <i>et al.</i>	3/38	4/34	50
Tibshirani <i>et al.</i>	2/38	2/34	21
SVM RFE	2/38	1/34	31
PLR RFE	2/38	1/34	26

one of the 70 selected genes by van't Veer *et al.* (2002) and one of the 6 chosen genes by Yeung *et al.* (2005).

3.2.2 Leukemia data

This data set consists of 38 training samples and 34 test samples of the two types of acute leukemia, acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL) (Golub *et al.* (1999)). Each sample is a vector of 7129 genes. The C value is limited by 0.2 due to the domain of log function. The performance of the classifier is evaluated on the test tissue samples.

$C = 0.1$ is used in the simulation for results in Table 3.3.

The results of Golub *et al.* (1999), Tibshirani *et al.* (2002), SVM with RFE and PLR with RFE are summarized in Table 3.4.

The CV-error in Table 3.4 is for 10-fold cross-validation error.

The result in Table 3.5 is with $C=0.1$, $a_m=0.37 \times 36$.

The result in Table 3.6 is for $C=0.1$, $a_m = 0.34 \times (3^9)$.

Table 3.5. Parameters for the chosen genes in the final classifier from CLR MUR.

	1834	1882	2288	2642	4847	5772
b_{j0}	0.96	0.95	0.96	0.97	0.97	0.99
b_{j1}	-0.72	-0.67	-0.69	0.56	-0.78	0.68

Table 3.6. Parameters for the selected genes in the final classifier from CLR RFE.

	461	1745	1834	2020	3320	3847	4196	4847	5039
b_{j0}	0.99	0.97	0.96	1.00	0.99	0.99	0.98	0.97	0.97
b_{j1}	-0.73	-0.73	-0.72	-0.79	-0.79	-0.73	-0.71	-0.78	-0.75

3.3 Conclusion

We have proposed a CLR for the 2-class microarray cancer diagnosis classification problem. Besides doing classification, CLR can offer an estimate of the conditional density $p_i(x)$. The simulation results on the real leukemia data and breast cancer prognosis data show that when using the same set of genes, CLR identifies less significant genes with smaller error rate than SVM and PLR. When using the modified univariate ranking method to select relevant genes for leukemia classification, CLR provides 0 10-fold CV-error and 1 test error. Moreover, it removes more genes than SVM and PLR, which makes the set of chosen genes more manageable. Therefore, we think CLR is a good classification method for two class microarray cancer data and it can estimate the underlying distribution.

CHAPTER 4

Asymptotics and Variable Screening for Microarrays with Fixed Small Sample Size and Large Dimension

DNA microarray is a new and promising biotechnology which allows the monitoring of expression levels for thousands of genes simultaneously. Microarray is being applied more and more often in biological and medical research to address a wide range of problems, including identifying a set of candidate genes that are most likely related to the outcome in the experiment. Statistical considerations are frequently to the fore in the analysis of microarray data, as researchers shift through massive amounts of data and adjust various sources of variability in order to identify the most important genes among the many which are measured. However, there are many more candidate genes in microarrays than the number of available samples in almost all studies, which leads to the improper application of traditional statistical technology. Some existing statistical methods deal with a single gene at a time (see Chen *et al.* (2003)), or deal with the case in which the sample size $n \rightarrow \infty$ (see Shao (2006)). The present chapter describes insight for the analysis of microarray data with the more realistic case that sample size is fixed and the dimension of variables $p \rightarrow \infty$.

By applying any type of shrinkage estimation to a linear model, we have more clues in interpreting the effects of the predictors. For example, the best subset selection of size k method, which shrinks the coefficients by setting some coefficients to be exactly zero and

makes it much easier to interpret the data. In another words, it identifies the important variables for the outcome. This process is discrete since the genes are either retained or dropped, so that the obtained result after variable selection might be extremely unstable. For another type of shrinkage estimations, lasso estimation (see Zhu (2003 and 2004) and Tibshirani (1999)), could be a very good choice. Fu and Knight proved that the Lasso estimation is consistent when the regularization parameter over the sample size tends to 0 as sample size increases to infinity and the number of predictors p is fixed. But the condition doesn't hold in real microarray data, and it is hardly appropriate to consider asymptotic methods for $n \rightarrow \infty$ when in reality the sample size n is fixed.

In practice, it is often true that a given outcome of interest is affected significantly by only a few genes among the large number of candidate genes, and the rest of the genes are approximately irrelevant to the outcome. Under such an idea, our task is to identify these important genes based on the sampled data. This is actually a variable selection problem. In the current statistical literature, however, there is no established variable selection procedure that can deal with a variable selection problem with number of variables $p \rightarrow \infty$ while sample size n is fixed. Zheng and Loh (1997) considered linear model selection with high-dimensional covariates, but they assumed that the dimension of the covariates over the sample size tends to 0 as the sample size increases to infinity.

In this chapter, we use ridge regression estimation (see Hoerl and Kennard (1970), which shrinks the coefficients but does not set any of those to be exactly 0. The problem considered concerns the asymptotic properties of microarray data with fixed sample size n and very large dimension of gene variables. Furthermore, we propose a variable screening method to eliminate the insignificant candidate genes and prove its consistency. The shrinkage procedure of ridge regression is continuous, therefore it is quite stable. After screening out the genes, if necessary, we may apply an established variable selection method, such as AIC and BIC, to select a final set of genes and fit a linear model which interprets the relationship between the predictors and the outcome in a concise way.

Mean squared error (MSE) consistency of the ridge regression

estimators, as dimension $p \rightarrow \infty$ and sample size is fixed, is proved in section 4.1. The variable screening method is described in section 2. Finally, a simulation study is presented

in section 4.3 to investigate the performance of the variable screening method.

4.1 Mean Squared Error Consistency

Consider the model $Y = X\beta + \varepsilon$, where $E(\varepsilon) = 0$ and $\text{Var}(\varepsilon) = \sigma_p^2 I_n$, X is a $n \times p$ matrix and β is a $p \times 1$ vector. Apply ridge regression method to obtain the estimator of β , i.e.

$$\min_{\beta} (Y - X\beta)'(Y - X\beta), \quad (4.1)$$

subject to the condition

$$\sum_{j=1}^p \beta_j^2 \leq t_p.$$

According to Fu (2004), this is equivalent to the L_2 -norm SVM model

$$\min_{\beta} (Y - X\beta)'(Y - X\beta) + h_p \sum_{j=1}^p \beta_j^2, \quad (4.2)$$

where h_p is a regularization parameter. The equivalence is in the sense that for any positive constant t_p , there exists a positive constant h_p such that model (4.1) and the model (4.2) share the same optimal solution $\hat{\beta}$, so we can stay focused on the model (4.2). The convex objective function makes it feasible to get the optimal solution at

$$\hat{\beta} = (X'X + h_p I_p)^{-1} X'Y. \quad (4.3)$$

Throughout the chapter, X is the gene expression data and Y is the outcome vector for the n samples. In this section, we will discuss the MSE consistency of $\hat{\beta}$ when n is fixed while $p \rightarrow \infty$. Due to the fact p is involved in the dimension of X , we will state the conditions on components of X as dimension $p \rightarrow \infty$.

Assumption A. There exists a constant $0 \leq \delta < 0.5$ such that each component of X is $O(h_p^\delta)$. $\sigma_p = o(1)$, $\frac{1}{h_p} = o(1)$ and $\frac{h_p^{1-2\delta}}{p} = o(1)$ as $p \rightarrow \infty$.

Theorem 1. Under Assumption A, we claim that $\max_{1 \leq i \leq p} \text{var}(\hat{\beta}_i) \rightarrow 0$, as $p \rightarrow \infty$.

Proof.

$$\begin{aligned}
\text{var}(\hat{\beta}) &= (X'X + h_p I_p)^{-1} X' \sigma_p^2 X (X'X + h_p I_p)^{-1} \\
&= \frac{\sigma_p^2}{h_p} \left(\frac{X'X}{h_p} + I_p \right)^{-1} \frac{X'X}{h_p} \left(\frac{X'X}{h_p} + I_p \right)^{-1} \\
&= \frac{\sigma_p^2}{h_p} \left[\left(\frac{X'X}{h_p} + I_p \right)^{-1} - \left(\frac{X'X}{h_p} + I_p \right)^{-1} \left(\frac{X'X}{h_p} + I_p \right)^{-1} \right] \\
&= \frac{\sigma_p^2}{h_p} \left[\left(\frac{X'X}{h_p} + I_p \right)^{-1} - \left(\left(\frac{X'X}{h_p} + I_p \right)^2 \right)^{-1} \right] \\
&= \frac{\sigma_p^2}{h_p} \left[\left(\frac{X'X}{h_p} + I_p \right)^{-1} - \left(\frac{X'X}{h_p} \frac{X'X}{h_p} + I_p + 2 \frac{X'X}{h_p} \right)^{-1} \right].
\end{aligned}$$

So we have

$$\frac{\text{var}(\hat{\beta})}{\frac{\sigma_p^2}{h_p}} = A^{-1} - B^{-1},$$

where $A = \frac{X'X}{h_p} + I_p$ and $B = \frac{X'X}{h_p} \frac{X'X}{h_p} + I_p + 2 \frac{X'X}{h_p}$. Because each component of $\frac{X'X}{h_p}$ is $o(1)$ and $\frac{h_p^{1-2\delta}}{p} = o(1)$ (which implies $(\frac{X'X}{h_p} \frac{X'X}{h_p} + I_p + 2 \frac{X'X}{h_p})^{-1} = o(1)_{p \times p}$) under assumption A, we have

$$\frac{\text{var}(\hat{\beta}_i)}{\frac{\sigma_p^2}{h_p}} \rightarrow 1$$

as $p \rightarrow \infty$. Further,

$$\begin{aligned}
\left\| \frac{\text{var}(\hat{\beta})}{\frac{\sigma_p^2}{h_p}} - I_p \right\| &= \sup_{\|a\|=1} |a' A^{-1} a - a' a - a' B^{-1} a| \\
&= \sup_{\|a\|=1} |a' (A^{-1} - I_p) a - a' B^{-1} a| \\
&\rightarrow 0
\end{aligned}$$

So $\max_{1 \leq i \leq p} \text{var}(\hat{\beta}_i) \rightarrow 0$, as $p \rightarrow \infty$.

For the analysis of bias vector, we need the below two assumptions.

Assumption B. There are only p_0 components of β which are nonzero (p_0 doesn't depend on p). Furthermore, β is in the linear space generated by the rows of $X'X$ for sufficiently large p , i.e., there is a vector $b_{p \times 1}$ such that $\beta = X'X b$ when p is large enough.

As $p \rightarrow \infty$, n is fixed throughout the chapter. $X'X$ has at most n positive eigenvalues. Let λ_{ip} be the i th nonzero eigenvalue of $X'X$, $i = 1, 2, \dots, n$. Without loss of generality, we

assume $\lambda_{ip} > 0$.

Assumption C. Assume there is a sequence of positive numbers $\xi_p \rightarrow \infty$ such that $h_p = o(\xi_p)$. Moreover, there exists a finite positive constant c such that $\lambda_{ip} \geq c\xi_p$ for all $i = 1, 2, \dots, n$. (Normally we can set the $\xi_p = p$). The constant c does not depend on p .

Theorem 2. Under the assumptions B and C, we have $\max_{1 \leq i \leq p} \text{bias}(\hat{\beta}_i) \rightarrow 0$, as $p \rightarrow \infty$.

Proof. When p is large enough, let Γ be an orthogonal matrix such that

$$\Gamma' X' X \Gamma = \begin{bmatrix} \Lambda_{n \times n} & O_{n \times (p-n)} \\ O_{(p-n) \times n} & O_{(p-n) \times (p-n)} \end{bmatrix}_{p \times p},$$

where $\Lambda_{n \times n}$ is a diagonal matrix with elements $\lambda_{ip}, i = 1, 2, \dots, n$, then it follows that

$$\begin{aligned} \text{bias}(\hat{\beta}) &= E(\hat{\beta}) - \beta \\ &= (X'X + h_p I_p)^{-1} X'X \beta - \beta \\ &= -\left(\frac{X'X}{h_p} + I_p\right)^{-1} \beta \\ &= -\left[\Gamma\left(\frac{\Gamma' X' X \Gamma}{h_p} + I_p\right)\Gamma'\right]^{-1} \beta \\ &= -\Gamma\left(\frac{\Gamma' X' X \Gamma}{h_p} + I_p\right)^{-1} \Gamma' \beta \\ &\doteq -\Gamma C \Gamma' \beta, \end{aligned}$$

where $C = \left(\frac{\Gamma' X' X \Gamma}{h_p} + I_p\right)^{-1}$ is a diagonal matrix with first n diagonal elements $\frac{h_p}{h_p + \lambda_{ip}}$, $i = 1, 2, \dots, n$, and the rest $p - n$ diagonal elements all equal to 1. Under assumption B,

$$\begin{aligned} \Gamma' \beta &= \Gamma' X' X \Gamma \Gamma' b \\ &= \begin{bmatrix} \Lambda_{n \times n} & O_{n \times (p-n)} \\ O_{(p-n) \times n} & O_{(p-n) \times (p-n)} \end{bmatrix} \Gamma' b. \end{aligned}$$

Notice $\Gamma' \beta$ is a $p \times 1$ vector and the only possible nonzero components are some of its first n components. By the assumption that only p_0 components of β are nonzero, we know the first n components of $\Gamma' \beta$ are finite. Since $\lambda_{ip} \geq c\xi_p$ for all $i = 1, 2, \dots, n$, and we obtain that $\max_{1 \leq i \leq n} h_p / (h_p + \lambda_{ip}) = O(h_p / \xi_p) = o(1)$. So $\max_{1 \leq i \leq p} \text{bias}(\hat{\beta}_i) \rightarrow 0$.

From the above fact the following result is also of interest.

Lemma 1. Under conditions A, B and C, we have the estimation $\hat{\beta}_i$ is MSE consistent for

β_i for all $i = 1, 2, \dots, p$, as $p \rightarrow \infty$.

Proof. $\text{MSE}(\hat{\beta}_i) = E(\hat{\beta}_i - \beta_i)^2 = \text{var}(\hat{\beta}_i) + \text{bias}^2(\hat{\beta}_i) \rightarrow 0$ when both terms approach 0 as $p \rightarrow \infty$.

4.2 Variable Screening Procedure

Let a_p be a sequence of positive numbers satisfying $a_p \rightarrow 0$ as $p \rightarrow \infty$. For each p value, we screen out the i th gene if and only if $|\hat{\beta}_i| \leq a_p$. Therefore, after screening out procedure, only genes associated with $|\hat{\beta}_i| > a_p$ are remained in the model as predictors. The sequence a_p acts as a filter in the process and eliminates genes with relative small coefficients. We will prove the consistency of the procedure when there are only finite components of β are nonzero.

Theorem 3. Suppose that assumptions A, B and C hold. There are only finite nonzero gene expression measurements in each sample as p increases to ∞ . We also assume $E|\varepsilon_i|^{2k} < \infty$ for an integer k such that $\frac{p}{(h_p^{1-\delta} a_p)^{2k}} = p(\frac{h_p}{\xi_p a_p})^{2k} = o(1)$, for all $i = 1, 2, \dots, n$. (One example of a choice is $\delta = 0.4$, $k = 20$, $\xi_p = p$, $h_p = c_1 p^{1/4}$, $a_p = c_2 p^{-1/9}$, where c_i 's are positive constants.) Then the variable screening method is consistent in the sense that

$$\begin{aligned} \lim_{p \rightarrow \infty} P(\text{All genes related to } Y \text{ are remained}) &= 1, \\ \lim_{p \rightarrow \infty} P(\text{all genes unrelated to } Y \text{ are screened out}) &= 1. \end{aligned} \quad (4.4)$$

Proof. Statement (4.4) is equivalent to

$$\lim_{p \rightarrow \infty} P(|\hat{\beta}_i| > a_p \text{ for all } i \text{ with } \beta_i \neq 0) = 1, \quad (4.5)$$

$$\lim_{p \rightarrow \infty} P(|\hat{\beta}_i| \leq a_p \text{ for all } i \text{ with } \beta_i = 0) = 1. \quad (4.6)$$

Assumptions A, B and C guarantee that $\hat{\beta}_i$ is MSE consistent for β_i , which implies for any $\epsilon > 0$,

$$\lim_{p \rightarrow \infty} P(|\hat{\beta}_i - \beta_i| > \epsilon) = 0.$$

Since $\beta_i \neq 0$ and $a_p = o(1)$, we have

$$\lim_{p \rightarrow \infty} P(|\hat{\beta}_i| > a_p) = 1.$$

That is equivalent to say

$$\lim_{p \rightarrow \infty} P(|\hat{\beta}_i| \leq a_p) = 0.$$

Therefore,

$$\begin{aligned} & P(|\hat{\beta}_i| > a_p \text{ for all } i \text{ with } \beta_i \neq 0) \\ &= 1 - P(\text{there exists at least one } i \text{ satisfying } |\hat{\beta}_i| \leq a_p \text{ among all } i \text{ with } \beta_i \neq 0) \\ &\geq 1 - \sum_{\{i: \text{all } i \text{ with } \beta_i \neq 0\}} P(|\hat{\beta}_i| \leq a_p). \end{aligned}$$

There are only finite components of β are nonzero, and this finishes the proof for (4.5).

If $\beta_i = 0$,

$$\begin{aligned} \hat{\beta} - E(\hat{\beta}) &= (X'X + h_p I_p)^{-1} X'(X\beta + \varepsilon) - (X'X + h_p I_p)^{-1} X'X\beta \\ &= (X'X + h_p I_p)^{-1} X'\varepsilon \\ &= B_{p \times n} \varepsilon_{n \times 1}, \end{aligned}$$

where

$$\begin{aligned} B &= (X'X + h_p I_p)^{-1} X' \\ &= \frac{1}{h_p^{1-\delta}} \times \left(\frac{X'X}{h_p} + I_p \right)^{-1} \times \frac{X'}{h_p^\delta} \\ &= (O(\frac{1}{h_p^{1-\delta}}))_{p \times n}. \end{aligned}$$

Let η_i be the i th component of $(X'X + h_p I_p)^{-1} X'\varepsilon$, then $\hat{\beta}_i = \text{bias}(\hat{\beta}_i) + \eta_i$. Therefore

$$\begin{aligned} P(|\hat{\beta}_i| > a_p) &\leq \frac{E|\hat{\beta}_i|^{2k}}{a_p^{2k}} \\ &= \frac{E|\text{bias}(\hat{\beta}_i) + \eta_i|^{2k}}{a_p^{2k}} \\ &\leq 2^{2k-1} \frac{|\text{bias}(\hat{\beta}_i)|^{2k} + E|\eta_i|^{2k}}{a_p^{2k}}. \end{aligned}$$

From the discussion in the previous section: $\text{bias}(\hat{\beta}_i) = O(\frac{h_p}{\xi_p})$.

From the analysis of matrix $B \times \varepsilon$: $E|\eta_i|^{2k} = O(\frac{1}{h_p^{2k(1-\delta)}})$.

Consequently,

$$\begin{aligned}
P(|\hat{\beta}_i| > a_p \text{ for at least one } i \leq p \text{ with } \beta_i = 0) &\leq \sum_{\{i: \text{all } i \text{ with } \beta_i=0\}} P(|\hat{\beta}_i| > a_p) \\
&\leq pP(|\hat{\beta}_i| > a_p) \\
&= O(p(\frac{h_p}{\xi_p a_p})^{2k}) + O(\frac{p}{(h_p^{1-\delta} a_p)^{2k}}).
\end{aligned}$$

For the given choice of (h_p, ξ_p, a_p) in the Theorem 3, both $O(\cdot)$ terms in the above expression are $o(1)$, which establishes result (4.6) and completes the proof.

The next result establishes the consistency of the variable screening method replacing the moment condition on ε , constraint on X and h_p by the normality assumption on the residual.

Theorem 4. Suppose the assumptions A-C hold. ε is normally distributed. Additionally, a_p, h_p and ξ_p are chosen in the way that $a_p = o(1)$, $\frac{h_p}{\xi_p} = o(a_p)$ and $\frac{h_p a_p^2 / \sigma_p^2}{\log \xi_p} \rightarrow \infty$. Then the result (4.4) holds.

Proof. Again, (4.4) is equivalent to (4.5) and (4.6). The proof of (4.5) is exactly the same as that in the proof of Theorem 3. It remains to show (4.6).

If $\beta_i = 0$, under the normality assumption on ε , $\hat{\beta}_i$ is normally distributed for each i . So

$$\begin{aligned}
P(|\hat{\beta}_i| > a_p) &= P(\hat{\beta}_i > a_p) + P(\hat{\beta}_i < -a_p). \\
&= \Phi\left(\frac{\text{bias}(\hat{\beta}_i) - a_p}{\text{sd}(\hat{\beta}_i)}\right) + \Phi\left(\frac{-\text{bias}(\hat{\beta}_i) - a_p}{\text{sd}(\hat{\beta}_i)}\right),
\end{aligned}$$

where Φ is the standard normal distribution function. In theorem 2, we proved that $\text{bias}(\hat{\beta}_i) = O(h_p/\xi_p)$ for all $i = 1, 2, \dots, p$. From the condition that $\frac{h_p}{\xi_p} = o(a_p)$, the bias part is $o(a_p)$. In Theorem 1, we have $\frac{\text{sd}(\hat{\beta}_i)}{\frac{\sigma_p}{\sqrt{h_p}}} \rightarrow 1$ for all $i = 1, 2, \dots, p$, which tells us

$$\frac{\pm \text{bias}(\hat{\beta}_i) - a_p}{\frac{\text{sd}(\hat{\beta}_i)}{\frac{\sigma_p}{\sqrt{h_p}}}} \rightarrow -1.$$

For a large enough p value, there exists a positive constant $t \in (0, 1)$ so that

$$P(|\hat{\beta}_i| > a_p) \leq 2\Phi(-t\sqrt{h_p a_p / \sigma_p}).$$

By assumption $\frac{h_p a_p^2 / \sigma_p^2}{\log \xi_p} \rightarrow \infty$, we have $t\sqrt{h_p a_p / \sigma_p} \geq \sqrt{2q \log(\xi_p)}$ for a large enough p value, where q is a constant such that $p/\xi_p^q = o(1)$.

Now we apply the inequality $2\Phi(-x) \leq e^{-x^2/2}$ for any $x \geq 1$. The probability of interest becomes

$$P(|\hat{\beta}_i| > a_p) \leq 2\Phi(-\sqrt{2q \log \xi_p}) \leq e^{-q \log(\xi_p)} = \xi_p^{-q}.$$

It follows that

$$\begin{aligned} P(|\hat{\beta}_i| > a_p \text{ for at least one } i \text{ with } \beta_i = 0) &\leq \sum_{\text{all } i \text{ with } \beta_i = 0} P(|\hat{\beta}_i| > a_p) \\ &\leq p/\xi_p^q \rightarrow 0. \\ \lim_{p \rightarrow \infty} P(|\hat{\beta}_i| \leq a_p \text{ for all } i \text{ with } \beta_i = 0) &= 1. \end{aligned}$$

That finishes the proof for (4.6).

4.3 Simulation Result

A simulation study was carried out to investigate the performance of the proposed variable screening method with fixed n and increasing p to study the asymptotic effect. We consider two sets of p : $p = 360$ and $p = 600$ with same $n = 60$ to evaluate the MSE consistency. For the variable selection part, we consider two sets of p : $p = 360$ and $p = 1200$. Since the convergent rate can also be influenced by the variability of the residual ε as in Theorem 1, we consider two values of the standard deviation of error : $\sigma = 1$ and $\sigma = p^{-1/9}$ combined with each p for simulation. We assume only the first 5 among all candidate genes are related to Y ($p_0 = 5$).

A flowchart for generating microarray data X is shown as the figure 4.1. Our n samples are from multi normal $N(p^{0.2}I_n, I_n)$.

In each of 1000 simulation runs, y_i was generated according to $y_i = x_i\beta + \varepsilon_i$, $i = 1, 2, \dots, n$, where ε_i 's were independently generated from $N(0, \sigma^2)$.

In table 3.1, we report MSE for the first 10 genes. The convergent tendency is quite clear. For distinguished nonzero β_i values, the MSE convergent rate is slower than that of the β_i 's which are zero. In the simulation, the variability of the error doesn't significantly affect the MSE convergent rate.

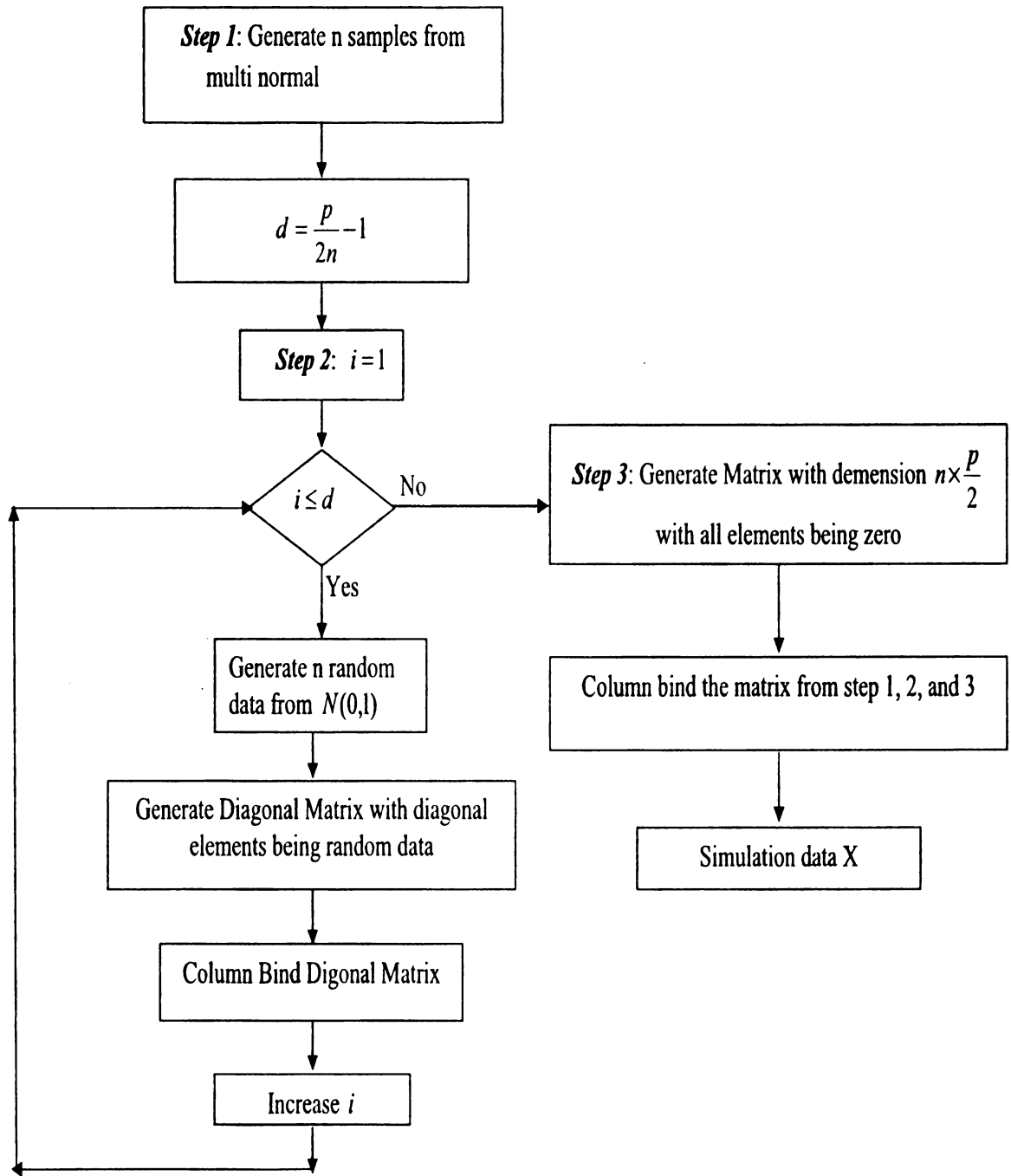


Figure 4.1. Flowchart of data generation



In Table 4.2, each simulation was applied to the generated data X and the true β vector with nonzero components are quite close to 0. We carry out the simulation with $h_p = p^{0.25}$, $a_p = p^{-1/11}$. Those two parameters are chosen to satisfy the conditions in Theorem 4. The results in Table 4.2 for small coefficients indicate that the proposed variable screening method can just leave out the unrelated gene variables gradually. The consistency of the screening method is strongly supported through the simulation study.

To show that the screening method is stable no matter how big or how small the nonzero components of β are, we provide the simulation result for big nonzero β_i in Table 4.3, where each simulation was applied to the generated data X with $h_p = p^{0.25}$, $a_p = p^{-1/11} \times 11$. The results in Table 4.3 are unbelievably perfect even for small p and large variability of the residual. The reason lies in the MSE consistency of the estimations.

Table 4.2 and 4.3 report the performance of the variable screening method in terms of two measures with results for f1, f2, g1, g2, h1 and h2. Note it is possible that after screening, some of the 5 variables related to Y are not selected although the number of remaining variables is 5 or larger.

Table 4.1. Simulation results for MSE consistence based on 1,000 runs (x fixed), $h_p = p^{0.25}$, $n = 60$, X is from the flowchart

σ	p		1	2	3	4	5	6	7	8	9	10
1	360	m1	0.281	0.747	0.129	0.317	0.204	0.022	0.019	0.103	0.025	0.084
	600		0.095	0.263	0.116	0.168	0.149	0.048	0.027	0.016	0.017	0.024
$p^{-1/9}$	360	m1	0.242	0.466	0.354	0.265	0.084	0.006	0.032	0.017	0.076	0.008
	600		0.146	0.272	0.251	0.181	0.019	0.022	0.008	0.016	0.015	0.003
1	360	m2	59.04	43.49	85.82	31.96	97.82	0.419	0.164	23.58	0.032	0.144
	600		17.11	27.75	40.77	12.91	34.21	0.013	0.424	5.763	0.097	3.638
$p^{-1/9}$	360	m2	38.89	84.19	63.94	81.19	40.76	15.23	1.782	2.832	1.046	2.802
	600		24.11	65.20	38.55	53.41	27.65	11.90	0.003	0.890	0.376	1.642

m1 = MSE of the first 10 gene variables with $\beta = (1.4, -2.5, 1.8, -1.7, 1.2, 0, 0, \dots, 0)'$

m2 = MSE of the first 10 gene variables with $\beta = (21.4, -22.5, 21.8, -21.7, 21.2, 0, 0, \dots, 0)'$

Table 4.2. Simulation results for close-to-0 nonzero β_i values based on 1,000 runs (x fixed)

σ	n	p		≤ 3	4	5	6	7	8	9	≥ 10
$p^{-1/9}$	60	360	f1	0	14	833	137	16	0	0	0
			f2	0	14	833	137	16	0	0	0
			g1	0	15	881	97	7	0	0	0
			g2	0	15	880	97	7	0	0	0
			h1	0	15	921	62	2	0	0	0
			h2	0	15	920	62	2	0	0	0
		1200	f1	0	73	891	36	0	0	0	0
			f2	0	73	887	36	0	0	0	0
			g1	0	75	896	29	0	0	0	0
			g2	0	75	896	29	0	0	0	0
			h1	0	76	904	20	0	0	0	0
			h2	0	76	904	20	0	0	0	0
	360	360	f1	0	3	835	162	0	0	0	0
			f2	0	3	833	162	0	0	0	0
			g1	0	4	944	52	0	0	0	0
			g2	0	4	944	52	0	0	0	0
			h1	0	5	982	13	0	0	0	0
			h2	0	5	982	13	0	0	0	0
		1200	f1	0	0	1000	0	0	0	0	0
			f2	0	0	1000	0	0	0	0	0
			g1	0	0	1000	0	0	0	0	0
			g2	0	0	1000	0	0	0	0	0
			h1	0	0	1000	0	0	0	0	0
			h2	0	0	1000	0	0	0	0	0

Table 4.2 is simulated with $\beta = (1.4, -2.5, 1.8, -1.7, 1.2, 0, 0, \dots, 0)'$, $h_p = p^{0.25}$, $a_p = p^{-1/11}$, $p_0 = 5$.

f1=frequencies of the number of remaining variables after screening

f2=frequencies of including all 5 relevant variables after screening

g1=frequencies of the number of remaining variables after screening and AIC

g2=frequencies of including all 5 relevant variables after screening and AIC

h1= frequencies of the number of remaining variables after screening and BIC

h2=frequencies of including all 5 relevant variables after screening and BIC

f1, f2, g1, g2, h1 and h2 are frequencies of including only variables related to Y when the remaining number of variables is less than 5

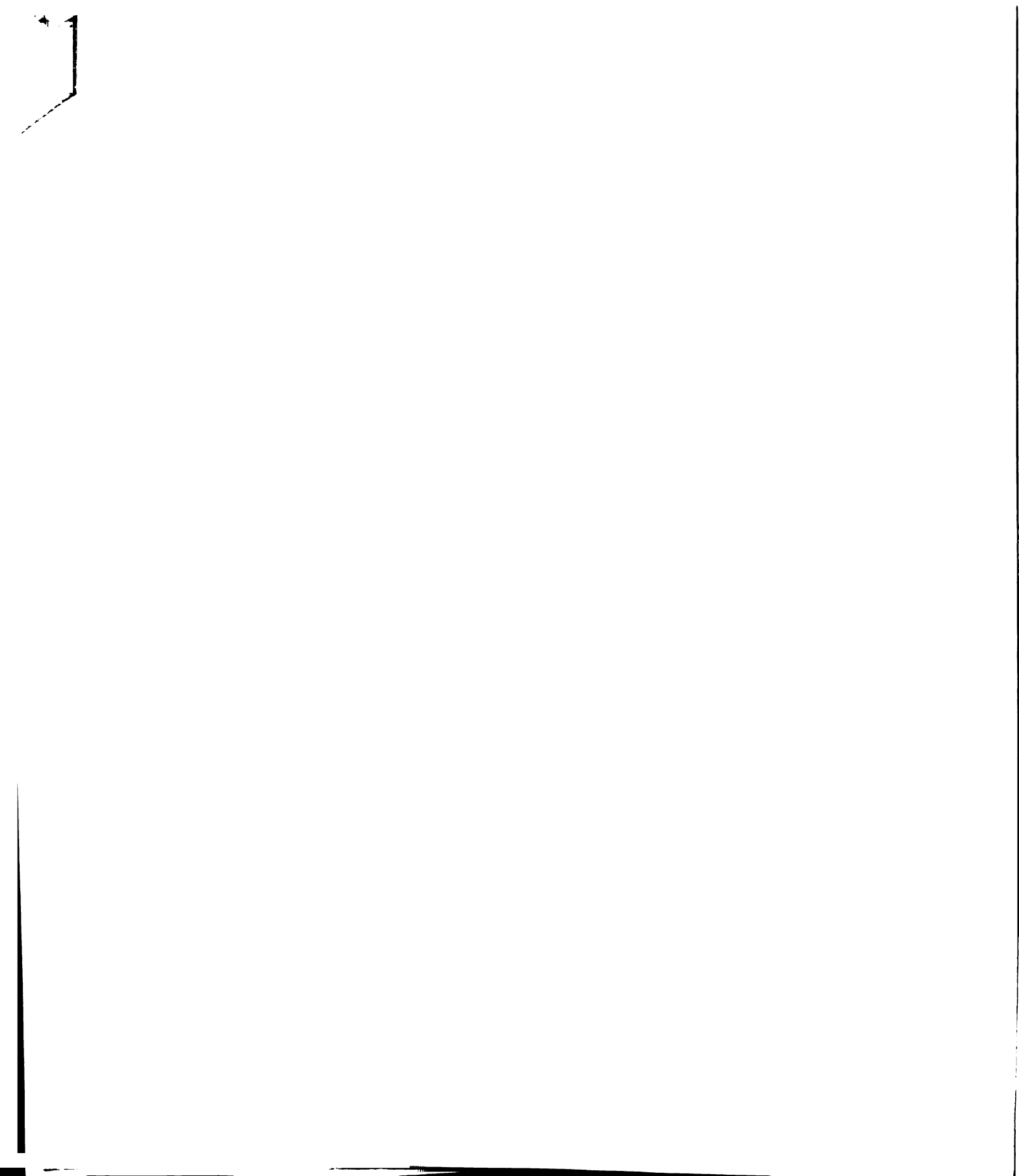


Table 4.3. Simulation results for distinguished nonzero β_i values based on 1,000 runs (x fixed)

σ	n	p		≤ 3	4	5	6	7	8	9	≥ 10
$p^{-1/9}$	60	360	f1	0	0	1000	0	0	0	0	0
			f2	0	0	1000	0	0	0	0	0
			g1	0	0	1000	0	0	0	0	0
			g2	0	0	1000	0	0	0	0	0
			h1	0	0	1000	0	0	0	0	0
			h2	0	0	1000	0	0	0	0	0
		1200	f1	0	0	1000	0	0	0	0	0
			f2	0	0	1000	0	0	0	0	0
			g1	0	0	1000	0	0	0	0	0
			g2	0	0	1000	0	0	0	0	0
			h1	0	0	1000	0	0	0	0	0
			h2	0	0	1000	0	0	0	0	0
		360	f1	0	0	1000	0	0	0	0	0
			f2	0	0	1000	0	0	0	0	0
			g1	0	0	1000	0	0	0	0	0
			g2	0	0	1000	0	0	0	0	0
			h1	0	0	1000	0	0	0	0	0
			h2	0	0	1000	0	0	0	0	0
		1200	f1	0	0	1000	0	0	0	0	0
			f2	0	0	1000	0	0	0	0	0
			g1	0	0	1000	0	0	0	0	0
			g2	0	0	1000	0	0	0	0	0
			h1	0	0	1000	0	0	0	0	0
			h2	0	0	1000	0	0	0	0	0

The above table is obtained with $h_p = p^{0.25}$, $a_p = 11 \times p^{-1/11}$, $p_0 = 5$, $\beta = (21.4, -22.5, 21.8, -21.7, 21.2, 0, 0, \dots, 0)'$. f1, f2, g1, g2, h1 and h2 are the same as those in Table 4.2.

4.4 Conclusions and Discussions

The established asymptotic result shows that for a fixed sample size n , the proposed variable screening method has some good properties when the dimension p is sufficiently large. It has perfect performance when the genes which are related to the outcome have significant influence on the outcome. For applications, we need to carry out some sensitivity analysis on the gene expression data set X to check those regularity conditions, and to determine a_p and h_p in the variable screening procedure. More research in how to reduce the requirements on X is called for.

The use of ridge regression is a crucial part for the estimations being MSE consistent of the true parameters and for the proposed variable screening method. Since ridge regression is a type of shrinkage estimation, the use of other SVM estimations might also produce a

good screening method. For example, the L_1 norm SVM estimation (or lasso estimation). However, the path of lasso estimation doesn't have a clear mathematical expression (Tibshirani (1996)). While ridge regression is much more easier in practice to get the properties of the estimations because of the known solution (Saunders and Gammernan (1998)).

Penalized linear models are considered for Y and X in this chapter. In general, the regression function between Y and X may be not strictly linear. When there exists a kernel transformation function $k(\cdot)$ and it is linear for Y and $K(X)$ in the feature space, our results for the MSE consistency and the consistency of the screening method can both be extended to the feature space for new variables. In the present chapter, our proofs and simulations are for fixed design matrix. More theoretical and numerical work for random design matrix is necessary.

CHAPTER 5

Gene Selection Methods for Microarray Data

It is important to know which genes are most relevant to the binary classification task and select these genes for a variety of reasons: removing noisy or irrelevant genes might improve the performance of the classifier, a candidate list of important genes can be used to further understand the biology of the disease and design further experiments, and clinical device recording on the order of tens of genes is much more economical and practical than one requiring thousands of genes.

The gene selection problem is an example of what is called feature selection in machine learning. Gibbs sampling (George (1993)) was used as one variable selection method. In the context of classification, feature selection methods fall into two categories filter methods and wrapper methods. Filter methods select features according to criteria that are independent of those criteria that the classifier optimizes. On the other hand, wrapper methods use the same or similar criteria as the classifier. We will discuss five feature selection approaches: univariate ranking (UR, Golub *et al.*, 2000), recursive feature elimination (RFE, Guyon *et al.*, 2002), ratio of between-group sum of squares to within-group sum of squares (BSS/WSS, reference), clustering (Sengupta (2003)), and pairwise ranking (Jonassen (2002)). They are either filter methods or wrapper methods. There are other biological methods for gene selection, see Dudoit (2000), Chow (2001), Kim (2002) and Jaeger (2003).



5.1 Univariate Ranking

Univariate ranking method is also called signal-to-noise or P-metric. It defines a statistic S_j for each gene variable j and assign a rank according to the value of statistic. S_j totally depends on the gene expression data. For each gene, we compute the following statistic:

$$S_j = \frac{\mu_+(j) - \mu_-(j)}{\sigma_+(j) + \sigma_-(j)}$$

where $\mu_+(j)$ and $\mu_-(j)$ are the means of the classes +1 and -1 for the j^{th} gene. Similarly, $\sigma_+(j)$ and $\sigma_-(j)$ are the standard deviations for the two classes for the j^{th} gene. Genes that give the most positive values are most correlated with class +1, and genes that give the most negative values are most correlated with class -1. One selects the most positive $m/2$ genes and the most negative $m/2$ genes, and then uses this reduced dataset for classification. A basic question that arises for all feature selection algorithm is how many genes the classifier should use. One approach to answer this question is using hypothesis and permutation testing (Golub *et al.* (1999)). The null hypothesis is that the UR statistic for each gene computed on the training set comes from the same distribution as that for a random data set. A random data set is the training set with its labels randomly permuted.

In detail, the permutation test procedure for the UR statistic is as follows:

- (1) Generate the statistic for all genes using the actual class label and sort the genes accordingly.
- (2) Generate 100 or more random permutations of the class labels. For each case of randomized class labels, generate the statistics for all genes and sort the genes accordingly.
- (3) Build a histogram from the randomly permuted statistics using various numbers of genes. We call this number k . For each value of k , determine different percentiles (1%, 5%, 50% etc.) of the corresponding histogram.
- (4) Compare the actual signal-to-noise scores with the different significance levels obtained for the histograms of permuted class labels for each value of k . See the figure for an illustration.

The solid curve is the UR statistic rank ordered computed on the training set. The three dashed lines are the 5th, 50th, and 95th percentiles of the same rank ordered statistic as

computed from the random data. The number of statistical genes is designated as the value of k , where the solid curve crosses the 5th percentile curve.

5.2 Recursive Feature Elimination

The method recursively removes features based upon the absolute magnitude of the hyperplane elements. We first outline the approach for linear SVMs. Given microarray data with n genes per sample, the SVM outputs the normal to the hyperplane, w , which is a vector with n components, each corresponding to the expression of a particular gene. Loosely speaking, assuming that the expression values of each gene have similar ranges, the absolute magnitude of each element in w determines its importance in classifying a sample, since the following equation holds:

$$f(x) = w \times x + b = \sum_{i=1}^n w_i x_i + b.$$

The idea behind RFE is to eliminate elements of w that have small magnitude, since they don't contribute much in the classification function. The SVM is trained with all genes; then we compute the following statistic for each gene:

$$S(j) = |w_j|$$

Where w_j is the value of the j^{th} element of w . We then sort S from largest to smallest value and we remove the genes corresponding to the indices that fall in the bottom 10% of the sorted list S . The SVM is retrained on this smaller gene expression set, and the procedure is repeated until a desired number of genes, m , is obtained. When a nonlinear SVM is used, the idea is to remove those features that affect the margin the least, since maximizing the margin is the objective of the SVM (Papageorgiou *et al.* (1998)). The nonlinear SVM has a solution of the following form:

$$f(x) = \sum_{i=1}^l c_i K(x, x_i) + b.$$

Let M denote the margin. Then we obtain Equation below:

$$\frac{1}{M} = \sum_{p,r=1}^l c_p c_r K(x_p, x_r) = \left\langle \sum_{i=1}^l c_i \phi(x_i), \sum_{j=1}^l c_j \phi(x_j) \right\rangle = \|w\|^2.$$



So for each gene j , we compute to which extent the margin changes using the following statistic:

$$S(j) = \left| \frac{\partial(1/M)}{\partial(x_j)} \right|, \quad (5.1)$$

where x_j is the j^{th} element of a vector of expression values x . We then sort S from the largest to the smallest value, and we remove the genes corresponding to the indices that fall in the bottom 10% of the sorted list S . The SVM is retrained and the procedure is repeated just as the linear case.

5.3 Ratio of Between-group sum of squares to Within-group sum of squares

The ratio of between-group to within-group sum of squares (BSS/WSS) was first introduced by Dudoit *et al.* (2002). About the BSS/WSS, intuitively, genes with relatively large variation between classes and relatively small variation within classes are likely candidates as relevant genes. BSS/WSS is a univariate gene selection method in which genes with large BSS/WSS ratios are good candidate relevant genes. For a gene j , let D_{ij} denote the expression level of gene j under sample i , \bar{D}_{kj} denote the average expression level of gene j over samples in class $k \in \{+1, -1\}$, and $\bar{D}_{.j}$ denote the average expression level of gene j over all samples. The BSS/WSS ratio for gene j is defined as

$$\frac{BSS(j)}{WSS(j)} = \frac{\sum_i \sum_k I(y_i = k) (\bar{D}_{kj} - \bar{D}_{.j})^2}{\sum_i \sum_k I(y_i = k) (D_{ij} - \bar{D}_{kj})^2} \quad (5.2)$$

We compute the BSS/WSS ratio for each of the p genes and order the genes in descending order of the BSS/WSS ratio.

5.4 Clustering method

Given a series of microarray experiments for a specific tissue under different conditions we want to find the genes most likely differentially expressed under these conditions. In another words, we want to find the genes that best explain the effects of these conditions. This task is also called feature selection, a commonly addressed problem in machine learning, where



one has class-labeled data and wants to figure out which features best discriminate among the classes. If the genes are the features describing the cell, the problem is to select the features that have the biggest impact on describing the results and to drop the features with little or no effect. These features can then be used to classify unknown data. Noisy or irrelevant attributes make the classification task more complicated, as they can contain random correlation. Therefore we want to filter out these features. Typically, informative genes are selected according to a test statistic or p-value rank according to a statistical test such as the t-test. The problem here is that we might end up with many highly correlated genes. Besides being an additional computational burden, it also can skew the results and lead to misclassifications. Additionally, if there is a limit on the number of genes to choose we might not be able to include all informative genes. Our approach is to first find similar genes, group them and then select informative genes from these groups to avoid redundancy, appeared in Gaeger (2003).

In order to increase the classification performance we propose to use more uncorrelated genes instead of just the top genes. By just using the k best ranking genes according to a test-statistic we would select highly correlated genes. Correlation can be a hint that the two genes belong to the same pathway, are co-expressed or are coming from the same chromosome. In general we expect high correlation to have a meaningful biological explanation. If, e.g., genes A and B are in the same pathway it could be that they have similar regulation and therefore similar expression profiles. If gene A has a good test score it is highly likely that gene B will, as well. Hence a typical feature selection scheme is likely to include both genes in a classifier, yet the pair of genes provides little additional information than either gene alone. Of course we could just select more genes in order to capture all relevant genes. But not only would more genes involve higher computational complexity for classification but it also can skew the result if we have a lot more genes from one pathway. Furthermore if there are several pathways involved in the perturbation but one pathway has the main influence, we will probably select all genes from this pathway. If we then have a limit for the number of genes we might end up with genes only from this pathway. If many genes are highly correlated we could describe this pathway with fewer genes and reach the same precision. Additionally, we could replace correlated genes from this pathway by genes from

other pathways and possibly increase the prediction accuracy. The same issue might be true when selecting a lot of genes as well, but it is more compelling when we have a limited budget of genes and can only select a few genes.

Our method for gene selection will therefore be to pre-filter the gene set and drop genes that are very similar. For the remaining genes we will apply a common test statistic and pull out the highest-ranking genes. One way to find correlated genes would be to calculate the correlation between all genes. Here we have two options:

(1) Select from the best genes (according to a test statistic) that have a pair-wise correlation below a certain threshold.

(2) The k -th selected gene is the gene with highest p-value among all genes whose correlation with each of the first $k - 1$ is below the specified threshold.

Option (1) and option (2) are called "correlation method". Another approach is called "clustering method". Our idea is to cluster the genes, and then select one or more representative genes from each cluster. The cluster quality is assessed by looking at the average membership probability of its elements. An element belongs to the cluster to which it has the highest membership probability. A higher cluster quality means how dispersion, and the closer the quality gets to 0, the more scattered the cluster becomes. It would be favorable to take more genes from a cluster of bad quality than from a cluster with good quality.

The drawback is that a cluster might represent a pathway that is totally unrelated to the discrimination we look for. So we mask out and exclude clusters that have an average bad test statistic p-value.

5.5 Pairwise Ranking Method

DLD and FLD are two discriminant methods for which a discriminant axis a is computed on the basis of the available training data. The prediction using axis a is to assign to class +1 if $a'(x - \frac{\mu_1 + \mu_2}{2}) > 0$, where μ_1 and μ_2 are the means of class +1 and -1, respectively. DLD axis is $a = S^{-1}(\mu_1 - \mu_2)$, where S is the diagonal variance matrix whose elements are the common variance estimate

$$\sigma_{gi}^2 = \frac{(n_1 - 1)\sigma_{1,gi}^2 + (n_2 - 1)\sigma_{2,gi}^2}{n_1 + n_2 - 2}.$$

We evaluate a gene pair by computing the projected coordinates of each experiment on the DLD axis using only these two genes. We then take the two sample t-statistic on the projected points as the pair score. In the exhaustive method, we sort the score of all pairs and select the top-ranked disjoint pairs. Assume the pair (g_i, g_j) ranks top 1, then all pairs containing g_i or g_j are removed from the list. In the greedy pairs method, we select the individual gene, g_i , with the highest t-score, and find the gene g_j that maximize the pair t-score. Then g_i, g_j are removed from the gene set and the procedure is repeated on the remaining set until we have selected the desired number of genes. See Bø (2002).

BIBLIOGRAPHY

- [1] U. Alon, N. Notterman, K. Gish, S. Ybarra, D. Mack and A. Levine, 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 Science*, 96, 6745-6750, (1999).
- [2] E. Allwein, R. Schapire and Y. Singer, Reducing multiclass to binary: a unifying approach for margin classifiers. *Journal of Machine Learning Research* 1:113-141, (2000).
- [3] P. Bradley and O. Mangasarian, Feature selection via concave minimization and support vector machines. In *J. Shawlik (eds), ICML'98*. Morgan Kaufmann, (1998).
- [4] T.H. Bø and I. Jonassen, New feature subset selection procedures for classification of expression profiles. *Genome Biology*, 3(4):research0017.1-0017.11, (2002).
- [5] M. Chow, E. Moler and I. Mian, Identifying marker genes in transcription profiling data using a mixture of feature relevance experts. *Physiol Genomics*, 5:99-111, (2001).
- [6] C. Cortes and V.N. Vapnik. Support vector networks. *Machine learning*, 20:273-297, (1995).
- [7] J. Chen and C. Chen, Microarray gene expression. In *Encyclopedia of Biopharmaceutical Statistics*, Ed. Chow, S.C., Marcel Dekker, Inc., New York, New York, 599-613, (2003).
- [8] T. Dietterich and G. Bakiri, Error-correcting output codes: A general method for improving multiclass inductive learning programs. *Proc. of the Ninth National Conference on Artificial Intelligence, AAAI Press*, 572-577, (1991).
- [9] S. Dudoit, H. Yang, M. Callow and T. Speed, Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. *Technical report 578*, University of California at Berkeley, August (2000).
- [10] S. Dudoit, J. Fridlyand and T. Speed, Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data. *JASA*, pp. 77-87(11), (2002).
- [11] M. Eisen, P. Spellman, P. Brown and D. Botstein, Cluster analysis and display of genome-wide expression patterns. *Proc. Natl Acad. Sci. USA*, 95, 14863-14868, (1998).
- [12] T. Evgeniou, M. Pontil and T. Poggio, Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13:1-50, (2000).

- [13] K. Knight and W. Fu, Asymptotics for lasso-type estimators. *The Annals of Statistics*, Vol.28, No. 5, 1356-1378, (2000).
- [14] W. Fu, Penalized regressions: the bridge versus the lasso. *Journal of Computational and Graphical Statistics*, 7 (3), 397-416, (2004).
- [15] E. George and R. McCulloch, Variable selection via Gibbs sampling. *J. Am. Stat. Assoc.*, 88, 881-889, (1993).
- [16] T. Golub *et al.*, Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286, 531-536, (1999).
- [17] G. Getz, E. Levine and E. Domany, Coupled two-way clustering analysis of gene microarray data. *Proc Natl Acad Sci USA*, 97:12097-12084, (2000).
- [18] I. Guyon, J. Weston, S. Barnhill and V. Vapnik, Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389-422, (2002).
- [19] A. Hoerl and R. Kennard, Ridge regression biased estimation for nonorthogonal problems. *Technometrics*, 12, 55-67, (1970).
- [20] T. Hastie and R. Tibshirani, Classification by pairwise coupling. In Jordan M., Kearns M., Solla S., eds. *Advances in Neural Information Processing Systems*, MIT Press 10, (1998).
- [21] H. Hishigaki, K. Nakai, T. Ono, A. Tanigami and T. Takagi, Assessment of prediction accuracy of protein function from protein-protein interaction data. *Yeast*, 18:523-531, (2001).
- [22] T. Hastie, S. Rosset, R. Tibshirani and J. Zhu, The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5, 1391-1415, (2004).
- [23] J. Jaeger, R. Sengupta and W.L. Ruzzo, Improved gene selection for classification of microarrays. *Pac Symp Biocomput*, (2003).
- [24] J. Khan, J. Wei, M. Ringner, L. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. Antinescu, C. Peterson and P. Meltzer, Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nat. Med.*, 7, 673-679, (2001).
- [25] S. Kim, E. Dougherty, J. Barrera, Y. Chen, M. Bittner and J. Trent, Strong feature sets from small samples C. *J. Comput. Biol.*, 9, 127-146, (2002).
- [26] S. Keerthi, K. Duan, S. Shevade, A. Poo, A fast dual algorithm for kernel logistic regression. *19th International Conference on Machine Learning*, (2002).
- [27] S. Li, Markov random field modeling in Computer vision. *Springer-Verlag*: Tokyo, (1995).

- [28] Y. Lee and C.K. Lee, Classification of multiple cancer types by multicategory support vector machines using gene expression data. *Technical Report 1051*, Department of Statistics, University of Wisconsin, Madison, WI 53706, (2002).
- [29] S. Mukherjee, P. Tamayo, D. Slonim, A. Verri, T. Golyb, J. Mesirov and T. Poggio, Support vector machine classification of microarray data. *Technical Report*, AI Memo 1677, MIT, (2000).
- [30] J. Platt, Sequential minimal optimization: A fast algorithm for training support vector machines. Microsoft Research, *Technical Report MSR-TR-98-14*, (1998).
- [31] J. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, MIT Press.
- [32] C. Perou, S. Jeffrey, M. van de Rijn, C. Rees, M. Eisen, D. Ross, A. Pergamenschikov, C. William, S. Zhu, J. Lee, D. Lashkari, D. Shalon, P. Brown and D. Botstein, Distinctive gene expression patterns in human mammary epithelial cells and breast cancers. *Proceedings of the National Academy of Science*, 96, 9212-9217, (1999).
- [33] B. Ripley, Pattern recognition and neural networks. *Cambridge: Cambridge University Press*, (1996).
- [34] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J. Mesirov, T. Poggio, W. Gerald, M. Loda, E. Lander and T. Golub, Multiclass cancer diagnosis using tumor gene expression signatures. *Proc. Natl. Acad. Sci. USA*, 26:15149-15154, (1998).
- [35] C. Saunders, A. Gammerman and V. Vovk, Ridge regression learning algorithm in dual variables. *15th International Conference on Machine Learning, ICML* (1998).
- [36] D. Slonim, P. Tamayo, J. Mesirov, T. Golub and E. Lander, Class prediction and discovery using gene expression data. *In Proc. of the 4th Annual International Conference on Computational Molecular Biology*, Universal Academy Press, 263-272, (2000).
- [37] G. Shieh, C. Bai, C. Lee, Identify Breast Cancer Subtypes by Gene Expression Profiles. *Journal of Data Science*, 2, 165-175, (2004).
- [38] B. Scholkopf, A.J. Smola, R.C. Williamson and D. Schuurmans, New support vector algorithms. *Neural Computation*, 12, 1207-1245, (2000).
- [39] J. Shao and SC Chow, Variable screening in predicting clinical outcome with high-dimensional microarrays. *J Multivariate Analysis*, (2006) To appear.
- [40] R. Tibshirani, Regression shrinkage and selection via the lasso. *J.R.S.S.B.*, 58, 267-288, (1996).
- [41] V. Vapnik, Statistical Learning Theory. *John Wiley & Sons*, New York, (1998).
- [42] V. Vapnik, The nature of statistical learning theory. *Springer Verlag*, New York, (1995).

- [43] L. van't Veer, H. Dai, M. van de Vijver, Y. He, A. Hart, M. Mao, H. Peterse, K. van der Kooy, M. Marton, A. Witteveen *et al*, Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415, 530-6, (2002).
- [44] M. Wang, J. Yang, G. Liu, Z. Xu, K. Chou, Weighted-support vector machines for predicting membrane protein types based on pseudo-amino acid composition. *PubMed*, 17(6):509-16, (2004).
- [45] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio and V. Vapnik, Feature selection for SVMs. *Advances in Neural Information Processing Systems 13*, MIT Press, (2001).
- [46] M. Xiong, L. Jin, W. Li and E. Boerwinkle, Computational methods for gene expression-based tumor classification. *BioTechniques*, 29:1264-1270, (2000).
- [47] Y.H. Yang, M.J. Buckley, S. Dudoit and T.P. Speed, Comparison of methods for image analysis on cDNA microarray data. *Technical report*, (2000).
- [48] Y.H. Yang, S. Dudoit, P. Luu and T.P. Speed, Normalization of cDNA microarray data. *SPIE Bios*, (2001).
- [49] K.Y. Yeung, C. Fraley, A. Murua, A.E. Raftery and W.L. Ruzzo, Model-based clustering and data transformation for gene expression data. *Bioinformatics*, 17:977-987, (2001).
- [50] K. Yeung, Bayesian Model Averaging: Development of an improved multi-class gene selection and classification tool for microarray data. *Bioinformatics*, 21, 2394-2402, (2005).
- [51] X. Zheng and W. Loh, A consistent variable selection criterion for linear models with high-dimensional covariates. *Statistica Sinica*, 7 (1997) 311-325.
- [52] J. Zhu, S. Rosset and T. Hastie, Margin maximizing loss functions. *Neural Information Processing Systems*, 16, (2003).
- [53] J. Zhu, S. Rosset, T. Hastie and R. Tibshirani, 1-norm support vector machines. *Neural Information Processing Systems 16*, (2004).
- [54] J. Zhu and T. Hastie, Classification of gene microarrays by penalized logistic regression. *Biostatistics*, 5(3), 427-443, (2004).

MICHIGAN STATE UNIVERSITY LIBRARIES



3 1293 02845 5313