IMPROVING THE PERFORMANCE OF THE PREDICTION ANALYSIS OF MICROARRAYS ALGORITHM VIA DIFFERENT THRESHOLDING METHODS AND HETEROSCEDASTIC MODELING

by

MOHAMMAD OMAR SAHTOUT

B.S., University of Jordan, 2003

M.S., University of Jordan, 2006

M.S., New Mexico State University, 2009

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the

requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Statistics

College of Arts and Sciences

KANSAS STATE UNIVERSITY

Manhattan, Kansas

2014

Abstract

This dissertation considers different methods to improve the performance of the Prediction Analysis of Microarrays (PAM). PAM is a popular algorithm for high-dimensional classification. However, it has a drawback of retaining too many features even after multiple runs of the algorithm to perform further feature selection. The average number of selected features is 2611 from the application of PAM to 10 multi-class microarray human cancer datasets. Such a large number of features make it difficult to perform follow up study. This drawback is the result of the soft thresholding method used in the PAM algorithm and the thresholding parameter estimate of PAM. In this dissertation, we extend the PAM algorithm with two other thresholding methods (hard and order thresholding) and a deep search algorithm to achieve better thresholding parameter estimate. In addition to the new proposed algorithms, we derived an approximation for the probability of misclassification for the hard thresholded algorithm under the binary case.

Beyond the aforementioned work, this dissertation considers the heteroscedastic case in which the variances for each feature are different for different classes. In the PAM algorithm the variance of the values for each predictor was assumed to be constant across different classes. We found that this homogeneity assumption is invalid for many features in most data sets, which motivates us to develop the new heteroscedastic version algorithms. The different thresholding methods were considered in these algorithms.

All new algorithms proposed in this dissertation are extensively tested and compared based on real data or Monte Carlo simulation studies. The new proposed algorithms, in general, not only achieved better cancer status prediction accuracy, but also resulted in more parsimonious models with significantly smaller number of genes.

IMPROVING THE PERFORMANCE OF THE PREDICTION ANALYSIS OF MICROARRAYS ALGORITHM VIA DIFFERENT THRESHOLDING METHODS AND HETEROSCEDASTIC MODELING

by

Mohammad Omar Sahtout

B.S., University of Jordan, 2003

M.S., University of Jordan, 2006

M.S., New Mexico State University, 2009

A DISSERTATION

submitted in partial fulfillment of the

requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Statistics

College of Arts and Sciences

KANSAS STATE UNIVERSITY

Manhattan, Kansas

2014

Approved by:

Major Professor

Haiyan Wang

Copyright

Abstract

This dissertation considers different methods to improve the performance of the Prediction Analysis of Microarrays (PAM). PAM is a popular algorithm for high-dimensional classification. However, it has a drawback of retaining too many features even after multiple runs of the algorithm to perform further feature selection. The average number of selected features is 2611 from the application of PAM to 10 multi-class microarray human cancer datasets. Such a large number of features make it difficult to perform follow up study. This drawback is the result of the soft thresholding method used in the PAM algorithm and the thresholding parameter estimate of PAM. In this dissertation, we extend the PAM algorithm with two other thresholding methods (hard and order thresholding) and a deep search algorithm to achieve better thresholding parameter estimate. In addition to the new proposed algorithms, we derived an approximation for the probability of misclassification for the hard thresholded algorithm under the binary case.

Beyond the aforementioned work, this dissertation considers the heteroscedastic case in which the variances for each feature are different for different classes. In the PAM algorithm the variance of the values for each predictor was assumed to be constant across different classes. We found that this homogeneity assumption is invalid for many features in most data sets, which motivates us to develop the new heteroscedastic version algorithms. The different thresholding methods were considered in these algorithms.

All new algorithms proposed in this dissertation are extensively tested and compared based on real data or Monte Carlo simulation studies. The new proposed algorithms, in general, not only achieved better cancer status prediction accuracy, but also resulted in more parsimonious models with significantly smaller number of genes.

Table of Contents

Table of Contents vi

List of Figures viii

List of Tables xiv

Acknowledgements xviii

Dedication xix

1 Introduction 1

2 Literature Review 7

2.1 Thresholding methods 7

2.2 Classification in high-dimensional setting 10

2.3 Nearest shrunken centroids 13

3 Improving the Original PAM Algorithm by Using Different Thresholding Methods and Deep Search Algorithm 16

3.1 Method 18

3.1.1 Nearest shrunken centroids classification with different thresholding methods 18

3.1.2 Deep search algorithm for thresholding parameter estimate 21

3.2 Data analysis 25

3.2.1 Performance of STh, OTh, and HTh 28

3.2.1.1 Detailed comparison 28

3.2.1.2 Overall comparison based on all ten data sets 40

3.2.2 Performance of STh2, OTh2, and HTh2 44

4 The Optimal Thresholding Parameter Estimate and The Probability of Misclassification 49

4.1 The choice of the optimal thresholding parameter estimate 49

4.2 Probability of misclassification 54
4.2.1 Probability of misclassification using the exact discriminant function . 56
4.2.2 Probability of misclassification using PAM discriminant function with hard thresholding 57

5 Feature Selection and Classification Based on Heteroscedastic Models 75
5.1 Introduction 75
5.2 Method 80
5.2.1 Heteroscedastic case test statistic and discriminant function 80
5.2.2 Thresholding the test statistics in the heteroscedastic case 83
5.3 Numerical comparisons 85
5.3.1 Simulation study 86
5.3.2 Real data analysis 100

6 Summary and Future Research 102
6.1 Summary 102
6.2 Future research 104

Bibliography 111

A Heatmaps for heterogeneity among different classes 112

List of Figures

3.1 SRBCT analysis: Scatter plot of the STh test error vs HTh and OTh test errors from 100 runs. The plotting symbol H (in red) is for HTh and O (in black) is for OTh. The numbers used in the plot are the frequencies of test errors out of 100 runs. The table gives a summary of the percentage of test errors. One sample out of the 20 test samples was misclassified for all three methods in all 100 runs, except for one run in which OTh misclassified 5 samples. Average number of genes used in OTh is about 1/3 of that by STh. "NaN" in the plot means that STh and HTh have similar numbers of misclassified samples in all 100 runs. 29

3.2 Breast cancer data analysis: Scatter plot of the STh test error vs HTh and OTh test errors from 100 runs. Count is used in the plot. The plotting symbol H (in red) is for HTh and O (in black) is for OTh. The numbers to the right of the plotting symbols report how many runs out of 100 has the plotted test error value. The table gives a summary of the percentage of misclassification in the 30 test samples. The probabilities given in the plot represents the proportion of times out of 100 runs that the test error for one thresholding method is smaller than another. The table reports a summary of the percent of misclassification error for test samples and average number of informative genes based on 100 runs. 30

3.3 Cancers analysis: Scatter plot of the STh test error vs HTh and OTh test errors from 100 runs. Count is used in the plot. The plotting symbol H (in red) is for HTh and O (in black) is for OTh. The numbers to the right of the plotting symbols report how many runs out of 100 has the plotted test error value. The table gives a summary of the percentage of misclassification in the 74 test samples. The probabilities given in the plot represents the proportion of times out of 100 runs that the test error for one thresholding method is smaller than another. The table reports a summary of the percent of misclassification error for test samples and average number of informative genes based on 100 runs. 31

3.4 DLBCL analysis: Scatter plot of the STh test error vs HTh and OTh test errors from 100 runs. Count is used in the plot. The plotting symbol H (in red) is for HTh and O (in black) is for OTh. The numbers to the right of the plotting symbols report how many runs out of 100 has the plotted test error value. The table gives a summary of the percentage of misclassification in the 30 test samples. The probabilities given in the plot represents the proportion of times out of 100 runs that the test error for one thresholding method is smaller than another. The table reports a summary of the percent of misclassification error for test samples and average number of informative genes based on 100 runs. 32

3.5 GCM data analysis: Scatter plot of the STh test error vs HTh and OTh test errors from 100 runs. Count is used in the plot. The plotting symbol H (in red) is for HTh and O (in black) is for OTh. The numbers to the right of the plotting symbols report how many runs out of 100 has the plotted test error value. The table gives a summary of the percentage of misclassification in the 46 test samples. The probabilities given in the plot represents the proportion of times out of 100 runs that the test error for one thresholding method is smaller than another. The table reports a summary of the percent of misclassification error for test samples and average number of informative genes based on 100 runs. 33

3.6 Leukemia1 cancer data analysis: Scatter plot of the STh test error vs HTh and OTh test errors from 100 runs. Count is used in the plot. The plotting symbol H (in red) is for HTh and O (in black) is for OTh. The numbers to the right of the plotting symbols report how many runs out of 100 has the plotted test error value. The table gives a summary of the percentage of misclassification in the 34 test samples. The probabilities given in the plot represents the proportion of times out of 100 runs that the test error for one thresholding method is smaller than another. The table reports a summary of the percent of misclassification error for test samples and average number of informative genes based on 100 runs. 35

3.7 Leukemia2 cancer data analysis: Scatter plot of the STh test error vs HTh and OTh test errors from 100 runs. Count is used in the plot. The plotting symbol H (in red) is for HTh and O (in black) is for OTh. The numbers to the right of the plotting symbols report how many runs out of 100 has the plotted test error value. The table gives a summary of the percentage of misclassification in the 15 test samples. The probabilities given in the plot represents the proportion of times out of 100 runs that the test error for one thresholding method is smaller than another. The table reports a summary of the percent of misclassification error for test samples and average number of informative genes based on 100 runs. 36

3.8 Leukemia3 cancer data analysis: Scatter plot of the STh test error vs HTh and OTh test errors from 100 runs. Count is used in the plot. The plotting symbol H (in red) is for HTh and O (in black) is for OTh. The numbers to the right of the plotting symbols report how many runs out of 100 has the plotted test error value. The table gives a summary of the percentage of misclassification in the 112 test samples. The probabilities given in the plot represents the proportion of times out of 100 runs that the test error for one thresholding method is smaller than another. The table reports a summary of the percent of misclassification error for test samples and average number of informative genes based on 100 runs. 37

3.9 Lung1 cancer data analysis: Scatter plot of the STh test error vs HTh and OTh test errors from 100 runs. Count is used in the plot. The plotting symbol H (in red) is for HTh and O (in black) is for OTh. The numbers to the right of the plotting symbols report how many runs out of 100 has the plotted test error value. The table gives a summary of the percentage of misclassification in the 32 test samples. The probabilities given in the plot represents the proportion of times out of 100 runs that the test error for one thresholding method is smaller than another. The table reports a summary of the percent of misclassification error for test samples and average number of informative genes based on 100 runs. 38

3.10 Lung2 cancer data analysis: Scatter plot of the STh test error vs HTh and OTh test errors from 100 runs. Count is used in the plot. The plotting symbol H (in red) is for HTh and O (in black) is for OTh. The numbers to the right of the plotting symbols report how many runs out of 100 has the plotted test error value. The table gives a summary of the percentage of misclassification in the 67 test samples. The probabilities given in the plot represents the proportion of times out of 100 runs that the test error for one thresholding method is smaller than another. The table reports a summary of the percent of misclassification error for test samples and average number of informative genes based on 100 runs. 39
3.11 The SRD comparison of mean test errors for the three thresholding methods. The x-axis and the left-side y-axis are the scaled SRD values. The right-side y-axis is the relative frequencies in percentages for the SRD theoretical distribution function. 42
3.12 The SRD comparison for the number of informative genes for the three thresholding methods. The x-axis and the left-side y-axis are the scaled SRD values. The right-side y-axis is the relative frequencies in percentages for the SRD theoretical distribution function. 43
3.13 The SRD comparison of mean test errors for the three algorithms STh2, OTh2, and HTh2. The x-axis and the left-side y-axis are the scaled SRD values. The right-side y-axis is the relative frequencies in percentages for the SRD theoretical distribution function. 45
3.14 The SRD comparison for the number of informative genes for the three algorithms STh2, OTh2, and HTh2. The x-axis and the left-side y-axis are the scaled SRD values. The right-side y-axis is the relative frequencies in percentages for the SRD theoretical distribution function. 45
3.15 The SRD comparison of mean test errors for all six algorithms STh, OTh, HTh, STh2, OTh2, and HTh2. The x-axis and the left-side y-axis are the scaled SRD values. The right-side y-axis is the relative frequencies in percentages for the SRD theoretical distribution function. 46
3.16 The SRD comparison for the number of informative genes for all six algorithms STh, OTh, HTh, STh2, OTh2, and HTh2. The x-axis and the left-side y-axis are the scaled SRD values. The right-side y-axis is the relative frequencies in percentages for the SRD theoretical distribution function. . . . 47

5.1 Heatmap of the sample standard deviation for 50 genes from the DLBCL cancer dataset. These are for the top 50 genes that have the highest range of the standard deviations for different classes. For most genes, the cell_lines class has the highest standard deviation which is 3 times more than standard deviation in other classes for some cases. 77
5.2 Heatmap of the sample standard deviation for 50 genes from the Leukemia2 cancer dataset. These are for the top 50 genes that have the highest range of the standard deviations for different classes. For most genes, the AML class has the highest standard deviation. The range of the standard deviations among different classes is more than 10,000 for a lot of genes. 78
5.3 Heatmap of the sample standard deviation for 50 genes from the Lung2 cancer dataset. These are for the top 50 genes that have the highest range of the standard deviations for different classes. The range of the standard deviations among different classes is up to 3,000 for some genes. For most genes, the NORMAL class has the lowest standard deviation. 79
5.4 The distributions of the different variables in each of the two classes in (5.3.1). The red curve corresponds to the pdf of the variables 21-10,000 in class 2 . . 88
5.5 The distributions of the different variables in each of the two classes in (5.3.2). The red curve corresponds to the pdf of the variables 21-10,000 in class 2 . . 88
5.6 The distributions of the different variables in each of the three classes in (5.3.3). The red curve corresponds to the pdf of the variables 21-10,000 in class 2 92
5.7 The distributions of the different variables in each of the three classes in (5.3.4). The red curve corresponds to the pdf of the variables 21-10,000 in class 2 93

A.1 Heatmap of the sample standard deviation for 50 genes from the SRBCT cancer dataset. These are for the top 50 genes that have the highest range of the standard deviations for different classes. 112
A.2 Heatmap of the sample standard deviation for 50 genes from the Breast cancer dataset. These are for the top 50 genes that have the highest range of the standard deviations for different classes. 113
A.3 Heatmap of the sample standard deviation for 50 genes from the Cancers dataset. These are for the top 50 genes that have the highest range of the standard deviations for different classes. 113

A.4 Heatmap of the sample standard deviation for 50 genes from the GCM cancer dataset. These are for the top 50 genes that have the highest range of the standard deviations for different classes. 114
A.5 Heatmap of the sample standard deviation for 50 genes from the Leukemia1 cancer dataset. These are for the top 50 genes that have the highest range of the standard deviations for different classes. 114
A.6 Heatmap of the sample standard deviation for 50 genes from the Lung1 cancer dataset. These are for the top 50 genes that have the highest range of the standard deviations for different classes. 115

List of Tables

3.1 Illustration of potential problem of thresholding parameter estimate in PAM. This is obtained for Leukemia2 data using pamr.cv with the seed of random number generation set to set.seed=100 in R 2.15.0. The number of genes survived soft thresholding corresponding to the smallest cv error could be drastically different from that corresponding to the second smallest cv error. 17

3.2 Summary of data sets used in this dissertation. 26

3.3 The SRD of mean test errors for the three thresholding methods. 41

3.4 Average number of informative genes based on 100 runs for each thresholding method. The value in parenthesis is the standard error. 43

3.5 The percent of mean misclassification error for test samples and average number of informative genes based on 100 runs for each thresholding method with and without the deep search algorithm. 48

4.1 Results based on our own code (Chapter 3) versus those based on modified pamr package (Chapter 4). The given results are the average for 100 runs of each algorithm with random partition of the training data in cross-validation. 51

4.2 Percent of misclassification error for test samples (test error) and number of selected genes (selec. genes) for the soft thresholding algorithm (STh) and the hard thresholding algorithm (HTh). The thresholding parameters were estimated by using cross-validation (CV) based on 100 runs, the universal thresholding (Uni.) $(2\log(\eta))^{-1/2}$, or Fan (1996) modified universal thresholding (M.Uni.) $[2\log\left(\eta\log^{-2}\eta\right)]^{-1/2}$. 52

4.3 Comparison of different thresholding parameter estimates for the order thresholding algorithm (OTh) based on percent of misclassification error for test samples (test error) and the number of selected genes (selec. genes). The thresholding parameters were estimated either using cross-validation based on 100 runs or Kim and Akritas's formula $[\log\eta]^{3/2}$. 53

5.1 Example 1 simulation results: Percent of mean misclassification error for test samples (test error), the average number of selected variables (selected variables), and the average of how many selected variables are from the first 20 double-signal variables. All these results are based on 100 runs for each algorithm. STh, HTh, and OTh are the homoscedastic version algorithms. STh3, HTh3, and OTh3 are the heteroscedastic version algorithms. 89
5.2 Example 2 simulation result: Percent of mean misclassification error for test samples (test error) and average number of selected variables (selected variables) based on 100 runs for each algorithm. The value in parenthesis is the median absolute deviation. STh, HTh, and OTh are the homoscedastic version algorithms. STh3, HTh3, and OTh3 are the heteroscedastic version algorithms. 91
5.3 Settings of real data for simulation settings. 94
5.4 Number of training samples (tr) and test samples (te) in each class of the data sets used in this dissertation. 95
5.5 Scenario 1 simulation result: Percent of mean misclassification error for test samples (test error) and average number of selected variables (selec. var.) based on 100 runs for each algorithms. STh, HTh, and OTh are the homoscedastic version algorithms. STh3, HTh3, and OTh3 are the heteroscedastic version algorithms. In all data sets the variables were generated according to (5.3.5) with 20 mean signals. 96
5.6 Scenario 2 simulation result: Percent of mean misclassification error for test samples (test error) and average number of selected variables (selec. var.) based on 100 runs for each algorithms. STh, HTh, and OTh are the homoscedastic version algorithms. STh3, HTh3, and OTh3 are the heteroscedastic version algorithms. In all data sets the variables were generated according to (5.3.6) with 20 mean signals for each class. 97
5.7 Scenario 3 simulation result: Percent of mean misclassification error for test samples (test error) and average number of selected variables (selec. var.) based on 100 runs for each algorithms. STh, HTh, and OTh are the homoscedastic version algorithms. STh3, HTh3, and OTh3 are the heteroscedastic version algorithms. In all data sets the variables were generated according to (5.3.7) without mean signals. 98

5.8 Comparing Scenario 1 simulation result to Scenario 3 simulation result for the soft thresholding method algorithms STh and STh3. 99
5.9 Real Data Analysis: Percent of mean misclassification error for test samples (test error) and average number of selected genes (selec. genes) based on 100 runs for each algorithms. STh, HTh, and OTh are the homoscedastic version algorithms. STh3, HTh3, and OTh3 are the heteroscedastic version algorithms. 101

Acknowledgments

First and foremost, I would like to thank Almighty Allah for instilling in me the ability and courage to continue my education. Allah has given me the strength and has been with me throughout this journey.

A special acknowledgment is given to my mother and father who unselfishly and generously gave of themselves to see me through my life successfully. They have always believed in the importance of learning and pushed me to pursue and excel in my education. I would not be where I am today in my life without their constant support and prayers.

My deepest gratitude goes to my beloved wife whom patiently listened during the rough times and encouraged me to push through the obstacles with my head held high. She filled my place when I could not be there for my children and for that I am eternally grateful.

I would like to express my sincere thanks and appreciation to my major advisor Dr. Haiyan Wang for her continuous encouragement, generous advice and drive to make me become a better statistician and researcher. I am indebted to her for the time, support and life lessons she has given me. She has helped me mold this work into what it is today.

Another recognition belongs to the committee members: Dr. Paul Nelson, Dr. Weixin Yao, Dr. Ming-Shun Chen, and Dr. Virginia Naibo. I appreciate their involvement in making my work the best it can be. I am grateful to Dr. Paul Nelson who always provided me with honest lessons that I will carry with me the rest of my life. The relationships Dr. Weixin Yao has built with his students will continue to influence my daily interactions with my own students.

I would also like to thank all faculty members of the Statistics Department for the quality coursework and support during these last five years at Kansas State University. Additionally, I would like to thank the faculty members of the Statistics Department at New Mexico State University and the Math Department at the University of Jordan for shaping my education. A special thanks to Dr. Mofid Azzam at the University of Jordan. He has been my role model, for not only his vast knowledge, but also his caring attitude towards his students.

A heartfelt gratitude also goes to my sisters and parents-in-law for their care and love during this time. They have always shown me the importance of family.

Finally, I would like to take this opportunity to recognize friends and colleagues I knew prior to starting my degree and met along the way, whom through their endless support, became family to me.

Dedication

This work is dedicated to my parents and family for their never-ending love and support.

Chapter 1

Introduction

Enormous amount of high dimensional data have been created by modern sciences and technologies. Some well known fields that generate such data are genomics, satellite imaging, document classification, web browsing, and online consumer transactions. Consider the cancer genomics using microarray data as an example. There are tens of thousands of genes that are available and coded by specific genetic sequences. During cancer progression, genes in some cells are overly expressed leading to excessive growth and promotion of tumor cell division. The expression value of a gene, measured as scanned intensity in microarray experiments, reflects the activity of the gene. In genomics study, the expression values of tens of thousands of genes can be measured simultaneously. Abundant gene expression data from cancer microarrays are now publicly available. A common characteristic of these data is that the number of samples (i.e. sample size) is usually less than one hundred but the number of genes is much larger. Identifying biologically important genes that are highly related to cancer progression is an important step toward better understanding of the disease mechanism and development of effective therapeutic drug. Statistically, this problem can be approached with variable selection or feature selection. Closely related to the genes identification is the cancer status prediction. Classification models or methods serve as one of the tools for such purpose.

A key challenge for variable selection and classification for high dimensional data is the large number of predictors compared to the relatively small sample size. Typically, model-

based variable selection and classification require to estimate a large number of unknown parameters. This is difficult when the sample size is small. The problem is aggravated by the presence of large amount of noise contributed by ultra high dimensional noisy variables (i.e. genes) that are irrelevant to the disease status. Including a large number of irrelevant variables in a model prevents accurate parameter estimation and leads to reduced classification accuracy. In the literature, there are not many methods that use the entire set of variables to build classifiers. Among those, several have relatively better performance in terms of accuracy. They are Support Vector Machine classifiers (SVM) by Chang and Lin (2011), Naive Bayes classifier (NB), and k-Nearest Neighbor (k-NN). Since these methods use all the variables to build classifiers, they do not offer any guidance on identification of important genes. As a result, it is impossible to design further experiments for follow-up study of some genes. Recent efforts have been focused mostly on variable screening to filter out irrelevant variables or using penalization methods to shrink small parameter estimates toward zero. Thresholding is one of the techniques used for this purpose. It uses cutoff thresholds on some test statistics (such as a t-test to compare the mean values of a variable in two different classes) to determine which variables have important contribution to classification accuracy. A review of several thresholding methods and classification methods will be given in this dissertation.

One popular thresholding based method is the Nearest Shrunken Centroids classifier by Tibshirani et al. (2002). It is known as the Prediction Analysis of Microarrays (PAM) and has been widely cited by the scientific community. PAM was originated from the Naive Bayes classifier assuming that the conditional distribution of predictor values given a class label is normally distributed with predictor specific variance. The variance of the values for each predictor were assumed to be constant across different classes in PAM. If the conditional distribution of predictor values given the class label is correctly specified, the Naive Bayes classifier gives the optimal classification due to Bayes Theorem. Different from Naive Bayes classifier that uses all the variables, PAM only uses the variables that have survived a soft thresholding. Specifically, define a class centroid to be the column vector of

features consisting of mean values over all samples in the class. Similarly, define the overall centroid to be the column vector of features composed of mean values over all samples across all classes. PAM compares the class centroids to the overall centroid using a t-test in each dimension. The test statistics were compared to a thresholding parameter. Those statistics with absolute values greater than the thresholding parameter are shrunken by the amount of the thresholding parameter. On the other hand, those statistics whose absolute values are below the thresholding parameter are set to zero. As a result, the variables with small test statistic values are removed from further classification. Basically, this is the soft thresholding. The soft thresholding parameter of PAM is obtained from cross-validation. In general, PAM is a simple to use algorithm that gives good accuracy for both binary and multi-class classification problems. However, there is an undesired characteristic of PAM. Due to the random partition in cross-validation for thresholding parameter estimate, PAM is unstable in variable selection. There could be several thousands of selected variables in one run of PAM but only a few in another run. In general, it tends to select too many variables.

The aforementioned disadvantage of PAM is due to two reasons: one is the choice of the thresholding method and the other is the thresholding parameter estimate used in the PAM algorithm. Different thresholding methods could lead to different performance. In Fan and Fan (2008), they gave a classifier called Featured Annealed Independence Rules classifier (FAIR). FAIR is actually a modified version of the nearest shrunken centroid classifier in binary classification problem by using hard thresholding to replace soft thresholding. Hard thresholding sets all test statistic values that are less than the thresholding parameter to zero and uses the rest for the classification process. In addition, FAIR selects the thresholding parameter by minimizing their derived upper bound of the classification error. The authors reported that FAIR is able to drastically reduce the number of selected variables tested on three data sets. However, FAIR is only applicable to two classes. More general cases with multi-class data using different thresholding methods remain to be investigated. In our exploration, we also find that the thresholding parameter makes critical difference for

the performance of the algorithm.

In this dissertation the author considers two other thresholding methods, hard and order thresholdings, to improve the performance of the Nearest Shrunken Centroids classifier. Order thresholding uses only a certain number of variables whose test statistic values are among the highest in absolute values. Different thresholding methods lead to different resulting classifiers. In addition to the different thresholding methods, an algorithm for better thresholding parameter estimate will be introduced in this dissertation. This algorithm will be referred as the deep search algorithm. To assess the performance of these new classifiers compared to the original PAM classifier, a study of 10 multi-class human cancer gene expression data sets was conducted. Each dataset contains samples from different human cancer types, including bladder, breast, central nervous system, colorectal, leukemia, lung, lymphoma, melanoma, mesothelioma, ovary, pancreas, prostate, renal, and uterus. For classification studies, these samples are divided into two parts, training samples part, and testing samples part. The training samples are used to train the classifiers. Then the resulted models are used to predict the class label (cancer type) of each sample from the testing samples. Further detailed information about these data sets will be given later in this dissertation. The nonparametric approach using the Sum of Ranking Difference (SRD) by Héberger (2010) was used to compare the overall performance of the different classifiers based on misclassification error for independent test samples obtained for all ten data sets. Our data analysis shows that the hard and order thresholding methods resulted in much smaller average number of genes and in general better classification accuracy as well.

Another important thing that affects the performance of the classifier is the proper choice of the thresholding parameter estimate. PAM algorithm uses cross-validation as a data-driven rule to select the thresholding parameter estimate. The estimated parameter minimizes the 10-fold cross-validation error over the training samples. In this dissertation we examine three different thresholding parameter estimates that were suggested in literature and compare them with the thresholding parameters obtained from cross-validation. The first two estimates are the universal thresholding parameter suggested by Donoho and

Johnstone (1994) and its modified version by Fan (1996), for both soft and hard thresholding. While the third estimate is for order thresholding recommended by Kim and Akritas (2010). A theoretical aspect of a classifier is its probability of misclassification. The theoretical result for the probability of misclassification for a classifier could also be used in estimating the optimal thresholding parameter. The optimal thresholding parameter is the parameter which minimizes the probability of misclassification. This dissertation provides an approximation for the probability of misclassification for the homoscedastic version of the hard thresholded algorithm under the binary case.

The other undesired characteristic of PAM is that it assumes homogeneity over different classes and hence uses the pooled within class standard deviation in its calculations. In this dissertation we show that the heterogeneity among class variances exist for many predictors in most cases. Therefore, the assumption of constant variance across different classes is not reasonable for most cases. For this reason, in this work we propose a variable selection and classification algorithms for the heteroscedastic case. Accordingly in developing the new heteroscedastic algorithms we considered the three thresholding methods. In this dissertation simulation studies in multiple scenarios are used to compare our new heteroscedastic case algorithms to their counter part homoscedastic algorithms including the original PAM. Our simulation result shows that the proposed heteroscedastic algorithms are superior to the homoscedastic algorithm in the presence of heterogeneity. The proposed heteroscedastic algorithms resulted in smaller test error and are better in identifying important variables than their counter part homoscedastic algorithms.

The organization of the rest of this dissertation is as follows. In Chapter 2, a literature review of the thresholding methods and the high-dimensional classification methods will be presented in addition to a detailed description of the Prediction Analysis of Microarrays (PAM) algorithm. In Chapter 3, a presentation of the idea of improving the PAM algorithm by considering two different thresholding methods, hard and order thresholding, are given. Moreover, this chapter presents the deep search algorithm. Then data analysis of 10 real data sets are discussed to compare the three methods and to asses the performance of

all proposed algorithms. In Chapter 4, we compare the thresholding parameter estimates obtained from cross-validation to different theoretically driven thresholding parameter estimates that were suggested in literature. Moreover, we present an approximation for the probability of misclassication for the hard thresholding algorithm in the two classes problem. Given the fact that the assumption of homogeneity is not reasonable in many data sets, in Chapter 5 we introduce feature selection and classication algorithms for the heteroscedastic case. Simulation and real data studies are used to validate the new algorithms in the case of heteroscedasticity. In Chapter 6 we give a summary for the research done in this dissertation and highlight its contributions to the field of high dimensional data classification. Finally, we list a few future research of interest.

Chapter 2

Literature Review

2.1 Thresholding methods

Thresholding has been introduced under several statistical topics; such as model selection, data mining, estimation, hypothesis testing, and image processing. Even though those topics are completely different from each other, the thresholding concept is the same. The idea is to use a subset of the data instead of the whole data hoping this will reduce the dimensionality and the noise-to-signal ratio. The main challenge, in selecting the best subset that can capture the desired features we are interested in, lies in minimizing the noise accumulation and at the same time keeping the important signals.

The start of the thresholding goes back to Neyman (1937) who proposed the truncation idea by using only the first m-dimensional subproblem for his smooth test for goodness of fit. Then Bickel (1983) introduced the soft thresholding in his work on multivariate normal decision theory. The more classical thresholding, hard thresholding, was defined in Donoho and Johnstone (1994) and compared with the soft threshoding when used for wavelet shrinkage in estimation process for nonparametric functions. The most recent order statistics based thresholding method, the order thresholding, was proposed in Kim and Akritas (2010) and Kim and Akritas (2012). They compared it to the other two thresholding methods in the context of testing against the high-dimensional alternative and goodness of fit testing. These three thresholding methods can be defined in general as follows.

Soft Thresholding:

$$\Lambda_S(x) = sgn(x)(|x| - \Delta_S)_+, \tag{2.1.1}$$

where Δ_S is the soft thresholding parameter and $+$ means positive part ($k_+ = kI\{k > 0\}$).

Hard Thresholding:

$$\Lambda_H(x) = xI\{|x| > \Delta_H\}, \tag{2.1.2}$$

where Δ_H is the hard thresholding parameter.

Order Thresholding:

$$\Lambda_O(x) = \begin{cases} x & if \quad rank(|x|) > n - \Delta_O \\ 0 & otherwise \end{cases}, \tag{2.1.3}$$

where Δ_O is the order thresholding parameter and n is the sample size before the thresholding.

The benefit of using thresholding techniques and the comparison between thresholding methods were the core topics of many literature pertaining to different applications. According to some articles in the literature (e.g. Bickel (1983), Donoho and Johnstone (1994), Fan (1996), Johnstone and Silverman (2004), Tibshirani et al. (2002), and Hall et al. (2008)) the thresholding methods perform better than their non-thresholded counterparts, provided that the thresholding parameter is chosen appropriately. One of the most interesting comments that had been said about the thresholding in literature is that when thresholding is used with the appropriate statistics it can be a good tool to find needles in haystack. Donoho and Johnstone (1994) applied soft and hard thresholding to wavelet coefficients in the context of nonparametric function estimation with Gaussian white noise model. By finding the asymptotic risk for both hard and soft thresholding estimators, they showed that the hard thresholding estimator exhibits the same asymptotic performance as the soft

thresholding estimator. Tibshirani (1996) proposed the Least Absolute Shrinkage and Selection Operator (LASSO) method for linear models estimation. The LASSO is a shrinkage and selection method that retains the good features of both subset selection and ridge regression. Interestingly, under the orthonormal design matrix case $X^T = X^{-1}$, Tibshirani found that the LASSO estimate has the same form as the soft thresholding (2.1.1). Fan (1996) proposed the adaptive Neyman test and the wavelet thresholding tests based on soft and hard thresholding as more powerful alternatives for the traditional distributional-based or linear rank-based test statistics. He suggested using the thresholding procedures over the adaptive Neyman test when there is no prior information about the signal concentration. The result of his simulation study shows that the hard thresholding outperforms both soft thresholding and adaptive Neyman test when testing multivariate normal mean in high dimension (i.e. $H_0 : \theta = 0$ vs $\theta \neq 0$, where $\mathbf{X} \sim N(\theta, I_n)$). Later in Fan and Fan (2008) it is shown that using most linear discriminant analysis for high-dimensional classification can perform as poorly as the random guessing due to noise accumulation resulted from using all the features in the data. Hence, they proposed the Features Annealed Independence Rule (FAIR) that uses the hard thresholding to select a subset of important features in the high-dimensional data. Kim and Akritas (2010) found that the order thresholding techniques improved the power of Pearson's chi-square test significantly more than the hard and soft thresholding in testing against the high-dimensional alternative under the Gaussian distribution. In addition, using the asymptotic theory they showed that the choice of the order thresholding parameter can be very flexible.

The appropriate choice of the thresholding parameter is critical for the good performance of the thresholding method. Donoho and Johnstone (1994) studied the minimax optimal thresholding parameter for the soft thresholding nonlinearity and they offer computer programs for the calculations. Since the implementation of the optimal thresholds for any other case would require a huge computational effort they suggest to use a universal thresholding parameter $(2 \log n)^{1/2}$ for both soft and hard thresholding. This universal parameter is asymptotically optimal and dose not involve computation burden. Fan (1996)

suggested taking the thresholding parameter to be $(2\log(na_n))^{1/2}$, with $a_n = c(\log n)^{-d}$ for some positive constants c and d. This choice of the thresholding parameter is meant to remove most of the noise (because $(2\log n)^{1/2}$ is the maximum of n independent Gaussian white noises) and avoid filtering out all the important coefficients in the test statistics. On the other hand, cross-validation is used in Tibshirani et al. (2002) as a data-driven rule to select the soft thresholding parameter for their nearest shrunken centroids classifier. The optimal order thresholding parameter, defined as the number of false null hypothesis, was recommended to be estimated by $(\log n)^{3/2}$ by Kim and Akritas (2010) in their simulations.

There are other thresholding methods. For example, the Stein's unbiased risk estimate in Donoho and Johnstone (1995), the false discovery rate in Benjamini and Hochberg (1995), the block thresholding methods of Cai and Silverman (2001) and Cai (2002), and the empirical Bayes thresholding in Johnstone and Silverman (2004). In this dissertation, we only consider the three methods in our previous discussion, i.e. soft, hard, and order thresholdings. Classification in high-dimensional data is one of the most popular areas where thresholding is implemented. The next section will be a review of classification methods in literature.

2.2 Classification in high-dimensional setting

In case of categorical response, classification is the process of obtaining a function from the training samples to predict the response category (class label) for any new observation (sample). Logistic regression, Fisher discriminant analysis, and Bayes classifier are some examples of classification methods in low-dimension setting, where the number of predictors (features) p is less than the sample size n.

Under high-dimensional settings ($p > n$), the aforementioned traditional classifiers will breakdown due to insufficient sample size to estimate the large number of unknown parameters. This problem is termed as the "curse of dimensionality" in literature, cf. Donoho (2000) and Fan et al. (2006). The other problem with high-dimensional data is the existence of large amount of noise that contribute negatively toward both parameter estimation and

classification errors.

To overcome these problems many ideas and new methods have been introduced in literature. Mainly, there are four directions of developing classification methods. In the first direction researchers concentrate on the classification accuracy to construct new and better classifiers using the same entire set of variables (features). The Naive Bayes classifier (NB), also known as the independence rule, is one example of this direction. Other examples of this direction are some versions of the Support Vector Machine classifiers (SVM) by Chang and Lin (2011), k-Nearest Neighbor (k-NN), and the projection methods such as using the Principal Component Analysis for dimension reduction (Bair et al. 2006). The use of the entire set of features in these methods will result in high noise accumulation and therefore less classification accuracy.

In the second direction, available classifiers such as SVM or Linear Discriminant Analysis (LDA) are used but the focus is on finding the best features. Example classifiers in this direction are GEMS-SVM by Statnikov et al. (2005), SCAD-SVM by Zhang et al. (2006), and the Features Annealed Independence Rule (FAIR) by Fan and Fan (2008). Such methods usually start with ranking the variable according to some univariate measurements, then selecting a subset of the highest ranked variables to use in the classification process. Even though those ranking methods reduce the dimensionality of the problem, they measure the importance of each variable individually and do not consider the correlation or interaction among variables.

The third direction is assembly of different classifiers or same classifiers on different subspaces. Methods of this direction are called the ensemble methods. An ensemble method could be a combination of tree classifiers where each trained on random subspace. Then the majority of votes will be used as decision rule resulting in method called Random Forest (Breiman 2001). Bagging (Breiman 1996) and Boosting (Freund and Schapire 1997) are other examples of ensemble classifiers. Difficulties of these classifiers include; the unclarity on how to decide the number of subclassifiers that should be included in the ensemble, and the large memory size required to run all those subclassifiers.

Finally, the last direction has dual goals of both feature selection and giving a better classifier. Examples of this direction are the wrapper methods (Kohavi and John 1997), the Prediction Analysis of Microarrays (PAM) (Tibshirani et al. 2002), and the Binary Matrix Shuffling Filter (BMSF) (Zhang et al. 2012), among others. Classifiers of this direction often link the feature selection with the classification accuracy. They overcome some drawbacks of classifiers of previously mentioned directions. This makes these classifiers widely adapted for high-dimensional classification.

Another way of grouping classifiers is according to the type of classification problem they handle. There are two types of high-dimensional classification problems, either binary classification or multi-class classification. This depends on the number of categories of the response variable. For binary classification, the response variable has two levels such as in the problem of identifying cancer from no cancer samples. Some examples of classifiers originally designed for binary classification are SVM, FAIR, BMSF, and Top Scoring Pairs family (TSP-family by Tan et al. (2005)). On the other hand, the multi-class classification deals with response variables that can take more than two levels such as the problem of distinguishing among different types of cancers or the problem of identifying subtypes of cancer. NB, PAM, and k-NN are multi-class classifiers. Moreover, many binary classifiers were extended to handle multi-class problems. See for example some versions of the Multicategory Support Vector Machine (MSVM) proposed in Weston and Watkins (1999), Crammer and Singer (2001), and Lee et al. (2004). Researchers also used different techniques for dividing the multi-class problem into multiple binary classification problem and then use binary classifiers. The three most popular techniques are one-vs-one (Hastie and Tibshirani (1998), Knerr et al. (1990)), one-vs-others (Hsu and Lin (2002), Rifkin and Klautau (2004), Vapnik (1998)), and hierarchical classification (Dumais and Chen 2000).

The PAM can handle multi-class classification problems directly and performs feature selection using soft thresholding with the goal of reaching better classification accuracy. It is widely adapted by researchers in many fields and is simple to implement. The next section will be a detailed review of the PAM classifier.

2.3 Nearest shrunken centroids

Nearest Shrunken Centroids is known by the Prediction Analysis of Microarray-s (PAM). Given a set of n training samples from K different classes and each is a vector with p genes, the single entry x_{ij} represents the gene expression for gene i of sample j and y_j represents the class label for sample j. Without loss of generality we can assume the classes are labeled 1 through K, such that $y_j \in \{1, 2, \ldots, K\}$. Let n_k represent the number of samples from class k and C_k be the set of indices for those samples.

The centroid for class k is the column vector with the mean gene expression values, $\overline{x}_i^{(k)} = \sum_{j \in C_k} x_{ij}/n_k$. The overall centroid is the column vector of mean gene expression values over all classes, $\overline{x}_i = \sum_{j=1}^{n} x_{ij}/n$. The goal of the PAM is to shrink each class centroid to the overall centroid. The algorithm starts by standardizing the centroids by the within-class gene expression standard deviation. Genes with stable expression values for samples within the same class should gain more weight according to this standardization. This process will result in a t statistic for comparing class k to the overall centroid

$$d_{ik} = \frac{\overline{x}_i^{(k)} - \overline{x}_i}{m_k(s_i + s_0)}, \tag{2.3.1}$$

where

$$s_i^2 = \frac{\sum_k \sum_{j \in C_k} (x_{ij} - \overline{x}_i^{(k)})^2}{n - K}, \tag{2.3.2}$$

and $m_k = \sqrt{1/n_k - 1/n}$. The s_0, set to be the median of the s_i values, is a constant to guard against large d_{ik} values caused by the possibility of small gene expression values. Then, the centroid of the k^{th} class can be written as a function of the t statistic in (2.3.1)

$$\overline{x}_i^{(k)} = \overline{x}_i + m_k(s_i + s_0)d_{ik}. \tag{2.3.3}$$

After performing a soft thresholding on the d_{ik} values using

$$d'_{ik} = sgn(d_{ik})(|d_{ik}| - \Delta_S)_+, \tag{2.3.4}$$

where Δ_S is the thresholding parameter and $+$ means positive part (i.e. $b_+ = bI\{b > 0\}$), the new shrunken centroid is defined as

$$\overline{x}_i'^{(k)} = \overline{x}_i + m_k(s_i + s_0)d'_{ik}. \tag{2.3.5}$$

In Tibshirani et al. (2002), a 10-fold cross-validation were used to choose the thresholding parameter or the amount of shrinkage that minimizes the classification error. The resulting shrunken centroids in (2.3.5) will then be used for classifying any new sample, say $x^* = (x_1^*, x_2^*, \ldots, x_p^*)$, by first computing the discriminant score for each class using

$$\delta_k(x^*) = \sum_{i=1}^{p} \frac{(x_i^* - \overline{x}_i^{\backprime(k)})^2}{(s_i + s_0)^2} - 2\log \pi_k, \tag{2.3.6}$$

where π_k is the class prior probability satisfying $\sum_{k=1}^{K} \pi_k = 1$. This prior probability is used to account for the frequency of class k in the population and it can be estimated from the training-set by $\hat{\pi_k} = n_k/n$. Then, x^* will be classified as a sample coming from that class with the smallest discriminant score (i.e. $\underset{k}{argmin}\ \delta_k(x^*)$).

PAM is powerful because of two reasons: (1) Its use of thresholding to reduce noise signals. (2) The original classifier before using the shrunken centroids is the Naive Bayes classifier assuming that the conditional distribution of gene expression given a class is normally distributed with gene specific variance. The variance of gene expressions from the same gene were assumed to be constant across different classes. If the conditional distribution of gene expressions given the class label is correctly specified, the Naive classifier gives the optimal classification due to Bayes Theorem.

It is worth mentioning that the FAIR classifier by Fan and Fan (2008) is also a shrunken centroid classifier. It uses hard thresholding (instead of soft thresholding) on t-statistics to shrink the class centroids toward the overall centroid. Different from PAM that is applicable to both binary and multi-class problems, FAIR is only for binary classification. In addition, FAIR selects the thresholding parameter by minimizing their derived upper bound of the classification error. The decision boundary between the two classes is given by

$$\delta_{FAIR}(x^*) = \sum_{i=1}^{p} \frac{(\overline{x}_i^{(1)} - \overline{x}_i^{(2)})(x_i^* - \overline{x}_i)}{\hat{\sigma}_i^2} I\{\sqrt{\frac{n}{n_1 n_2}}|T_i| > b\}, \tag{2.3.7}$$

where T_i is the t-statistic for the i^{th} feature and b is the thresholding parameter. They chose b in a way such that there are m features with $\sqrt{\frac{n}{n_1 n_2}}|T_i| > b$. The optimal m is defined as

$$m_{opt} = \underset{1 \le m \le p}{argmax} \left[\frac{1}{\hat{\lambda}_{max}^m} \cdot \frac{n[\sum_{i=1}^{m} T_i^2 + m(n_1 - n_2)/n]^2}{m n_1 n_2 + n_1 n_2 \sum_{i=1}^{m} T_i^2} \right], \tag{2.3.8}$$

where $\hat{\lambda}^m_{max}$ is the largest eigenvalue of the correlation matrix $\mathbf{R}^m$.

Chapter 3

Improving the Original PAM Algorithm by Using Different Thresholding Methods and Deep Search Algorithm

Our first motivation to improve the popular Prediction Analysis of Microarrays (PAM) algorithm by Tibshirani et al. (2002) is that the PAM seems to select too many features. The average number of selected genes by the PAM is 2611 based on 10 multi-class microarray human cancer data sets considered in this study. For the case of cancer diagnostics studies, such a large number of genes are difficult to perform follow up experiments. One of the reasons for this drawback of the PAM algorithm is the soft thresholding method used in the PAM algorithm.

Another motivation of this study is the phenomenon we observed when examining the details of how PAM chose the thresholding parameter for Leukemia2 dataset (Armstrong et al. 2002). This phenomenon is that the number of genes survived soft thresholding corresponding to the smallest cross-validation error could be drastically different from that corresponding to the second smallest cross-validation error. On the other hand, the smallest and 2nd smallest cross-validation errors only differ by one misclassified sample. This could be a potential problem of the thresholding parameter estimate in PAM. Illustration

of this potential problem using Leukemia2 dataset is presented in Table 3.1 in which the number of genes and thresholding parameter estimate based on smallest and second smallest cross-validation errors are reported. In these cases, the number of genes corresponding to the smallest cross-validation error could be several thousand, while the number of genes corresponding to the 2nd smallest cross-validation error may be less than 100. In all cases, the smallest and second smallest errors differ only by misclassification of one more sample.

Table 3.1: *Illustration of potential problem of thresholding parameter estimate in PAM. This is obtained for Leukemia2 data using pamr.cv with the seed of random number generation set to set.seed=100 in R 2.15.0. The number of genes survived soft thresholding corresponding to the smallest cv error could be drastically different from that corresponding to the second smallest cv error.*

	parameter with smallest CV error			parameter with 2nd smallest CV error		
	threshold	n.genes	CV error	threshold	n.genes	CV error
run 1	0.418878	10283	5	7.539809	26	6
run 2	1.256635	6127	4	7.539809	26	6
run 3	0.418878	10283	4	7.12093	30	5
run 4	0.837757	7959	3	6.283174	77	4
run 5	6.283174	77	5	6.702052	52	6
run 6	1.675513	4735	5	6.702052	52	6
run 7	1.256635	6127	4	7.539809	26	5
run 8	7.539809	26	6	7.958687	18	7
run 9	6.702052	52	4	7.958687	18	5
run 10	1.256635	6127	5	0.418878	10283	6

Considering these two issues of the PAM algorithm, in this chapter we propose two ways of improving the PAM algorithm. One way is to replace the soft thresholding used in the PAM algorithm by either hard or order thresholding. The second way is to give a better estimate of the thresholding parameter. We will provide an algorithm that performs a deep search for selecting the optimal thresholding parameter.

3.1 Method

3.1.1 Nearest shrunken centroids classification with different thresholding methods

In this section, we present an improved version of the PAM algorithm. The order and hard thresholding will be used in the PAM algorithm instead of soft thresholding. To illustrate how the algorithm of PAM can be modified by using different thresholding method, we will consider first the case of replacing the soft thresholding by the hard thresholding. Assume that we have a set of n training samples and p genes. This will give a $n \times p$ matrix with each entry x_{ij} represents the gene expression for gene i of the training sample j, where $i = 1, ..., p$ and $j = 1, ..., n$. If those n training samples are from K different classes, let y_j denote the class label for sample j. The class labels are the different cancer types or subtypes. Without loss of generality we will give them the codes $\{1, 2, \ldots, K\}$. Let n_k denote the number of samples from the same class k and C_k be the set of indices for those samples. The idea of the PAM is to shrink each class centroid $\overline{x}_i^{(k)} = \sum_{j \in C_k} x_{ij}/n_k$ toward the overall centroid $\overline{x}_i = \sum_{j=1}^{n} x_{ij}/n$ by using soft thresholding.

The shrunken centroid for class k is written as

$$\overline{x}_i^{\backprime(k)} = \overline{x}_i + m_k(s_i + s_0)\hat{d}_{ik}, \tag{3.1.1}$$

where $m_k = \sqrt{1/n_k - 1/n}$ and $s_i^2 = \sum_k \sum_{j \in C_k} (x_{ij} - \overline{x}_i^{(k)})^2/(n - K)$. The s_0 is a constant that is set to be the median of the s_i values.

In the original PAM algorithm, the $\hat{d}_{ik}$ is the thresholded value of the test statistic

$$d_{ik} = \frac{\overline{x}_i^{(k)} - \overline{x}_i}{m_k(s_i + s_0)}, \tag{3.1.2}$$

using soft thresholding

$$d'_{ik} = sgn(d_{ik})(|d_{ik}| - \Delta_S)_+, \tag{3.1.3}$$

where + means positive part (i.e. $b_+ = bI\{b > 0\}$). The soft thresholding parameter Δ_S is chosen to be the thresholding value that minimizes the misclassification error in a 10-fold

cross-validation of the training samples.

To improve the PAM algorithm, we replace the soft thresholding in (3.1.3) by the hard thresholding

$$d_{ik}^{\prime\prime\prime} = d_{ik} I\{|d_{ik}| > \Delta_H\}, \tag{3.1.4}$$

where Δ_H is the hard thresholding parameter. The optimal Δ_H can be determined by using a 10-fold cross-validation over the training-set. It is selected to be the one that provides the amount of shrinkage that minimizes the cross-validation misclassification error.
Then the shrunken centroid for class k can be written as

$$\overline{x}_i^{\prime\prime\prime(k)} = \overline{x}_i + m_k(s_i + s_0)d_{ik}^{\prime\prime\prime}. \tag{3.1.5}$$

The new shrunken centroids for all K classes will be used to classify any new sample $z = (z_1, z_2, \ldots, z_p)$. This is done by computing the discriminant score for each class using

$$\delta_k^{\prime\prime\prime}(z) = \sum_{i=1}^{p} \frac{(z_i - \overline{x}_i^{\prime\prime\prime(k)})^2}{(s_i + s_0)^2} - 2\log \pi_k, \tag{3.1.6}$$

where π_k is the prior probability for class k satisfying $\sum_{k=1}^{K} \pi_k = 1$. Using the training-set, π_k can be estimated by $\hat{\pi}_k = n_k/n$. Then the decision is to classify the new sample z as coming from the class c if $\delta_c^{\prime\prime\prime}(z) = \min\limits_{1 \le k \le K} \delta_k^{\prime\prime\prime}(z)$, where $c \in \{1, 2, \ldots, K\}$.
In the binary case, this is the FAIR classifier except that the optimal thresholding value in FAIR is obtained with a fixed formula (2.3.8) instead of estimation with cross-validation. Fan and Fan (2008) found in their application to 3 binary cancer data sets that FAIR selects smaller number of genes than PAM.

In the case of improving the PAM algorithm using order thresholding, we replace (3.1.3) by

$$d_{ik}^{\prime\prime\prime} = \begin{cases} d_{ik} & if \quad rank(|d_{ik}|) > n - \Delta_O \\ 0 & otherwise \end{cases} \tag{3.1.7}$$

where Δ_O is the order thresholding parameter, which can be determined by using a 10-fold cross-validation over the training-set.

Then the shrunken centroid for class k in this case can be written as

$$\overline{x}_i'''^{(k)} = \overline{x}_i + m_k(s_i + s_0)d_{ik}''', \tag{3.1.8}$$

and it will be used to compute the value of the new sample z discriminant score using

$$\delta_k'''(z) = \sum_{i=1}^{p} \frac{(z_i - \overline{x}_i'''^{(k)})^2}{(s_i + s_0)^2} - 2\log \pi_k. \tag{3.1.9}$$

Similar to the hard thresholding case, the decision is to classify the new sample z as coming from the class c if $\delta_c'''(z) = \min_{1\leq k\leq K} \delta_k'''(z)$, where $c \in \{1, 2, \ldots, K\}$.

In all three methods described above, the informative genes are those genes that survived thresholding. The number of informative genes resulting from a specific thresholding method is the count of all the genes that have at least one non-zero thresholded d_{ik}. For example under the soft thresholding method, gene i can be counted as an informative gene if at least one d'_{ik} is a non-zero for any $k = 1, 2, \ldots, K$.

Estimation of the thresholding parameter The original PAM suggested to consider a grid search and the optimal thresholding parameter is the parameter value with the smallest cross-validation error. Assume we start with a set of m thresholding parameter values $\Theta_0 = \{\theta_{01}, \ldots, \theta_{0m}\}$. These values are typically taken to be evenly spaced in the range of the thresholding parameter. Note that the actual thresholding parameter for the soft thresholding lies in a continuous space. Hence the finite set Θ_0 may not contain the optimal parameter value, which could be between two values in Θ_0. Without loss of generality, we assume that $\theta_{01}, \ldots, \theta_{0m}$ are arranged in an increasing order. For the original PAM algorithm whose optimal parameter is defined as the value that has the smallest cross-validation error, we will repeatedly shrink the search range to find the value of the best thresholding parameter. Specifically, let $Err(\theta_{0i})$ represent the cross-validation error of the algorithm when θ_{0i} is the thresholding parameter, where $i = 1, \ldots, m$. Define $\tau_1 = \underset{1\leq i\leq m}{argmin}\{Err(\theta_{0i}), i = 1, \ldots, m\}$ to be the index of the thresholding parameter value whose corresponding cross-validation error $Err(\theta_{0\tau_1})$ is the smallest among all parameter

values in set Θ_0. That is, inequality $Err(\theta_{0k}) \geq Err(\theta_{0\tau_1})$ holds for all $k \neq \tau_1$. Then the optimal parameter of the original PAM algorithm is in the interval $(\theta_{0,\tau_1-1}, \theta_{0,\tau_1+1})$. We then consider a second set of thresholding parameter values $\Theta_1 = \{\theta_{11}, \ldots, \theta_{1m}\}$ evenly spaced in the interval $(\theta_{0,\tau_1-1}, \theta_{0,\tau_1+1})$. The parameter value in Θ_1 that has the smallest cross-validation error is then identified. Denote it as θ_{1,τ_2}. This leads to a even smaller interval $(\theta_{1,\tau_2-1}, \theta_{1,\tau_2+1})$ for further search. The process is repeated and a sequence of intervals $(\theta_{i-1,\tau_i-1}, \theta_{i-1,\tau_i+1})$ is obtained for $i = 1, 2, \ldots$. The search will be terminated when the number of variables surviving the thresholding remains unchanged for all parameters in an interval.

In practice, it is not necessary to always use m parameters in each interval when the range of the number of variables that survived thresholding is below m. In particular, denote $g_{ij}, j = 1, \ldots m$, to be the number of variables that survived thresholding in the PAM algorithm using parameter θ_{ij} in the i^{th} round of search. Then the further search only need to consider the number of different thresholding parameter values to be $\min\{m, \max_{1\leq j\leq m}(g_{ij}) - \min_{1\leq j\leq m}(g_{ij})\}$. This was used in the actual search algorithm. At the end of the search, there is only one parameter in the final interval and that is the optimal thresholding parameter for the original PAM algorithm. Theoretically speaking, the sequence of intervals $(\theta_{i-1,\tau_i-1}, \theta_{i-1,\tau_i+1})$ will converge to the optimal parameter if the cross-validation errors are good estimates of the true errors. Unfortunately, the random partition in a cross-validation may split the samples in a way such that the separation of the classes can be easily achieved even with very high dimensional noisy data. If this happens, the parameter obtained based on the original parameter estimation principle of PAM will be trapped to intervals far away from the optimal parameter value.

3.1.2 Deep search algorithm for thresholding parameter estimate

As mentioned in the beginning of this chapter and previous section, a problematic situation could occur with the original thresholding parameter estimate of PAM. This happens when the thresholding parameter corresponding to the smallest cross-validation error leads

to several thousands of genes surviving the thresholding while the number of genes corresponding to the 2nd smallest cross-validation error may be less than 100. Meanwhile, the smallest and 2nd smallest cross-validation errors only differ by one misclassified sample. One way to alleviate this problem is to employ multiple runs of cross-validation with different random partitions for each thresholding parameter. In the end, the optimal parameter is selected to be the value that minimizes the maximum (or average) cross-validation errors from multiple runs. This idea critically depends on the speed of the algorithm to perform one cross-validation run. We will explore this option in later studies.

An alternative idea to approach the problem is to take into consideration of whether the obtained smallest cross-validation is largely by chance due to a random partition that makes the class samples supper well-separated. When the samples are well separated, even a simplest method could classify the samples well. If this happens by chance due to a special partition, then the misclassification error for this thresholding parameter value would be very different from those for the nearby thresholding parameter values being considered. Specifically, suppose the algorithm starts with a set of m thresholding parameter values $\Theta_0 = \{\theta_{01}, \ldots, \theta_{0m}\}$ and the smallest cross-validation error is achieved by $\theta_{0\tau_1} \in \Theta_0$. If the true misclassification error curve is minimized at θ_0, a value far away from $\theta_{0\tau_1}$, then their will be other parameter values in Θ_0 that have misclassification errors close to that for $\theta_{0\tau_1}$. In such case, $\theta_{0\tau_1}$ is by chance to get the smallest cross-validation error due to a special partition. Suppose $\theta_{0\nu}$ is the parameter value in Θ_0 that is closest to the optimal parameter θ_0. Then the misclassification error for $\theta_{0\nu}$ is close to that of $\theta_{0\tau_1}$. Unfortunately, the magnitude of the optimal thresholding parameter is unknown to us and it is difficult to decide the sufficient distance between $\theta_{0\nu}$ and $\theta_{0\tau_1}$ to tell that the cross-validation error at $\theta_{0\tau_1}$ is minimized by chance among all values in Θ_0. On the other hand, the number of genes that survived thresholding with parameter $\theta_{0\tau_1}$ will be very different from the number of genes that survived thresholding with parameter $\theta_{0\nu}$. So we will judge whether $\theta_{0\tau_1}$ is a by chance minimizer by comparing the number of genes. Below is the algorithm to perform this task.

Deep search algorithm

Input: Training-set

Output: Optimal thresholding value

Assume we start with a set of m thresholding parameter values $\Theta_0 = \{\theta_{01}, \ldots, \theta_{0m}\}$. These values are typically taken to be evenly spaced in the range of the thresholding parameter. Without loss of generality, we assume that $\theta_{01}, \ldots, \theta_{0m}$ are arranged in an increasing order. Let $Err(\theta_{0i})$ represent the cross-validation error of the algorithm in terms of the number of misclassifications when θ_{0i} is the thresholding parameter, where $i = 1, \ldots, m$. Define $\tau = \underset{1 \leq i \leq m}{argmin}\{Err(\theta_{0i}), i = 1, \ldots, m\}$ to be the index of the thresholding parameter value whose corresponding cross-validation error $Err(\theta_{0\tau})$ is the smallest among all parameter values in set Θ_0. That is, inequality $Err(\theta_{0k}) \geq Err(\theta_{0\tau})$ holds for all $k \neq \tau$. Then suppose $\theta_{0\nu}$ is the parameter value in Θ_0 that has the 2nd smallest cross-validation error. That is, $Err(\theta_{0k}) \geq Err(\theta_{0\nu}) \geq Err(\theta_{0\tau})$ holds for all $k \neq \tau$, ν. Let $g_{ij}, j = 1, \ldots m$, denote the number of variables that survived thresholding using parameter θ_{ij} in the i^{th} round of search. The algorithm proceeds as follows,

1. Start by searching within the m thresholding values (m=30 default) to find the thresholding values corresponding to the smallest and 2nd smallest cross-validation (CV) error (i.e. $\theta_{0\tau}$ and $\theta_{0\nu}$).

- In case of more than one thresholding values with the same CV error, chose the one with the smallest number of selected genes.
- Set the temporary further search location as $\theta_{temp} = \theta_{0\tau}$.

2. The thresholding value corresponding to the 2nd smallest CV error ($\theta_{0\nu}$) can be assigned to θ_{temp} in our algorithm if both conditions in 2a and 2b are satisfied.

2a. The difference between the smallest and the 2nd smallest CV error does not differ by more than one misclassified sample (i.e. $Err(\theta_{0\nu}) - Err(\theta_{0\tau}) \leq 1$).

2b. The number of genes survived thresholding corresponding to the second smallest CV error ($g_{0\nu}$) is either

- less than half of that for the thresholding value with the smallest CV error (i.e. $2g_{0\nu} < g_{0\tau}$),

  or

- 2,000 less than that for the thresholding value with the smallest CV error (i.e. $g_{0\tau} - g_{0\nu} > 2000$).

After locating this initial thresholding value (θ_{temp}), the next process will be to deeply search the neighborhood of θ_{temp} for another possible thresholding value with smaller CV error. Record the index ℓ in Θ_0 such that $\theta_{temp} = \theta_{0\ell}$.

3. To identify the neighboring interval that will be investigated, consider both sides of the selected thresholding value (θ_{temp}). That is, both intervals ($\theta_{0,\ell-1}$, $\theta_{0\ell}$) and ($\theta_{0\ell}$, $\theta_{0,\ell+1}$).

- In case the selected thresholding value in step 2 is a boundary value (i.e. $\ell = 1$ or $\ell = m$).
  - If $\ell = 1$, just consider the right side of the selected thresholding value (i.e. interval ($\theta_{0\ell}$, $\theta_{0\ell+1}$)).
  - If $\ell = m$, just consider the left side of the selected thresholding value (i.e. interval ($\theta_{0,\ell-1}$, $\theta_{0,\ell}$)).
- The following two conditions specify which interval to perform the deep search:
  - Only perform the deep search on the interval ($\theta_{0\ell}$, $\theta_{0,\ell+1}$) if the difference in number of selected genes is more than one gene (i.e. $g_{0\ell} - g_{0,\ell+1} > 1$).
  - Only perform the deep search on the interval ($\theta_{0\ell-1}$, $\theta_{0,\ell}$) if the difference in number of selected genes is less than m (i.e. $g_{0,\ell-1} - g_{0\ell} < m$).

– If both $g_{0\ell} - g_{0,\ell+1} > 1$ and $g_{0,\ell-1} - g_{0\ell} < m$ conditions are satisfied, perform the deep search in $(\theta_{0\ell-1},\ \theta_{0,\ell+1})$.

After deciding on which interval to refine the search $(\theta_{0,\ell-1},\ \theta_{0,\ell+1})$, $(\theta_{0,\ell-1},\ \theta_{0\ell})$, or $(\theta_{0\ell},\ \theta_{0,\ell+1})$:

4. Now consider a second set of thresholding parameter values $\Theta_1 = \{\theta_{11}, \ldots, \theta_{1k}\}$ evenly spaced in the selected interval from the previous step.

- The number of thresholding values k is the minimum between m and the difference between the number of genes that correspond to the lower and upper bounds of the interval. For example, if the selected interval is $(\theta_{0\ell},\ \theta_{0,\ell+1})$, then the number of thresholding values to be considered is $k = \min(m, g_{0\ell} - g_{0,\ell+1})$.

5. Run cross-validation to obtain the CV errors for the set of selected thresholding values from the previous step.

6. If $k > 0$, repeat steps 1 to 5 with the parameter values in Θ_1. Otherwise, report the optimal thresholding value as the most recently obtained θ_{temp}.

3.2 Data analysis

Data sets Ten multi-class gene expression data sets for human cancers were investigated in this study. These ten data sets are listed in Table 3.2. We obtained those data sets from the first author of Tan et al. (2005). The number of classes in those data sets ranges from 3 to 14 and the number of genes ranges from 2308 to 16063. Each dataset contains two parts, training samples part (training-set), and testing samples part (test-set). The training samples are used to train the classifiers by calculating the class shrunken centroids. Then those shrunken centroids are used to predict the class label of each sample in the test-set.

Table 3.2: *Summary of data sets used in this dissertation.*

Dataset abbreviation	Platform	No of classes	No of genes	No of samples Training	Testing	Reference
SRBCT	cDNA	4	2308	63	20	Khan et al. (2001)
Breast	Affy	5	9216	54	30	Perou et al. (2000)
Cancers	Affy	11	12533	100	74	Su et al. (2001)
DLBCL	cDNA	6	4026	58	30	Alizadeh et al. (2000)
GCM	Affy	14	16063	144	46	Ramaswamy et al. (2001)
Leukemia1	Affy	3	7129	38	34	Golub et al. (1999)
Leukemia2	Affy	3	12582	57	15	Armstrong et al. (2002)
Leukemia3	Affy	7	12558	215	112	Yeoh et al. (2002)
Lung1	Affy	3	7129	64	32	Beer et al. (2002)
Lung2	Affy	5	12600	136	67	Bhattacharjee et al. (2001)

Methods to be compared The PAM algorithm in Tibshirani et al. (2002) uses the soft thresholding (3.1.3) to shrink the class centroids to the over all centroid. In this study, we compare the original PAM with the other two thresholding methods, hard thresholding (3.1.4) and order thresholding (3.1.7). For easier discussion, from now on we will refer to the hard thresholded PAM algorithm by HTh, the order thresholded PAM algorithm by OTh, and the soft thresholded PAM algorithm (the original PAM) by STh.

The R software, version 2.15.0, was used for programming of those three PAM algorithms. In our code for the STh, we mainly used functions from the pamr package that was developed by the authors of Tibshirani et al. (2002). The pamr.cv function is used to perform cross-validation for selecting the soft thresholding parameter. This function uses 30 thresholding values by default. These values evenly split the whole range of the test statistic values (3.1.2). The default number of thresholding values 30 is small compared to the range of the test statistic values. So one application of pamr.cv with 30 values may not find the optimal thresholding value. In our code we refine the neighborhood of the thresholding value with the smallest cross-validation error following the search procedure described at the end of Section 3.1.1. Specifically, we first identify a shorter interval and evenly re-split this interval

into 30 values. Then we calculate their cross-validation error for each value. This process will continue until we reach the thresholding value with the smallest cross-validation error. After determining the shrinkage parameter using cross-validation, the pamr.train function is used to build the classifier with the informative genes that survived the thresholding. Then the model is used to classify the class label of each test sample by applying the method of nearest centroid classification using the pamr.predict function.

For the HTh and OTh algorithms, we wrote our own functions to calculate the class centroids, to perform cross-validation, and to predict the class label for the test samples. The refining process is also implemented in our code for these two algorithms. In all three algorithms the number of folds for the cross-validation is set to be 10 unless some class sample size is less than 10. In the later case, the fold is set to be the smallest class size.

STh, HTh, and OTh use the smallest cross-validation error for the thresholding parameter estimate. The deep search algorithm in Section 3.1.2 results in possibly different parameter estimate. We refer to these algorithms using soft, hard, and order thresholding along with deep search algorithm for parameter estimate as STh2, HTh2, and OTh2, respectively.

Comparison metric For fair comparison, we train each classifier with the training samples and predict the class label for the test samples. The data used in this dissertation were already divided into training and test samples by earlier authors Tan et al (2005). We will adopt the same partition in our study. In binary classification problems, multiple metrics (such as proportion of correctly classified samples, Mathew's Correlation coefficient, sensitivity, specificity, area under the receiver operating characteristic (ROC) curve) may be used for comparison. In the case of at least 3 classes, proportion of correctly classified samples or (misclassified samples) is typically used in the literature as the comparison metric. When discussion is within the same dataset, the number of misclassified test samples by different methods can also be used. We will use the test error in our comparison. It is defined as the percent of misclassification error, which is equal to the number of misclassified test

samples divided by the total number of test samples. We will also compare the number of informative genes used in each method. It is widely accepted that the better method is one that used less genes to achieve the same accuracy as other methods using more genes.

3.2.1 Performance of STh, OTh, and HTh

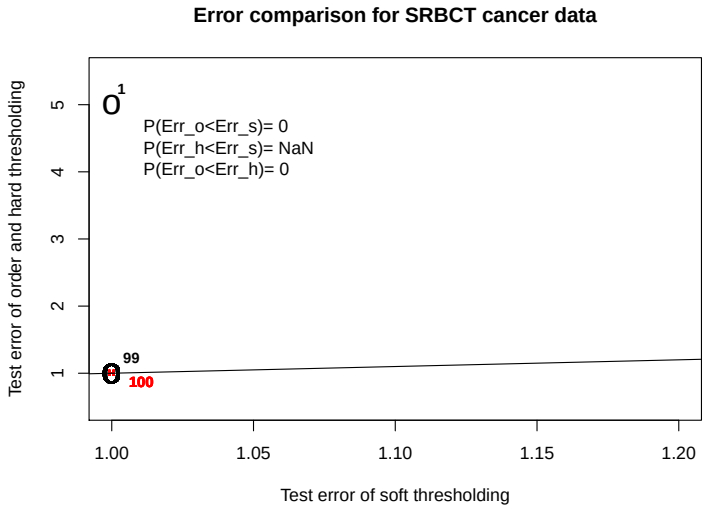
In this section, we discuss the performance of the three PAM algorithms using the 10 multi-class human cancers data sets. In all that follows, our reported misclassification error refers to the percentage of misclassified test samples. We repeated this process 100 times for each dataset. The random partition of the training data in cross-validation could lead to different estimated thresholding parameter and hence possibly a different test error.

3.2.1.1 Detailed comparison

We will start by discussing the results of each dataset individually. For each dataset, we will compare the performance of the three PAM algorithms; the STh (the original PAM) that uses soft thresholding, the HTh that uses hard thresholding, and the OTh that uses order thresholding. For better visualization of our comparison, in Figures 3.1 to 3.10 we plotted the test errors of the STh against the test errors of both OTh and HTh. We computed the number of times out of 100 runs that the OTh has less test error than the STh and this proportion is given in the plots as $P(Err_o < Err_s)$. Similarly, $P(Err_h < Err_s)$ and $P(Err_o < Err_h)$ are given in the plots with their meaning accordingly defined. Below those plots we reported the mean, median, and standard error of the different algorithms based on the 100 runs. The average number of informative genes for each algorithm is also reported.

Starting with the small round blue cell tumors (SRBCT) dataset analysis in Figure 3.1, we can see that only one sample out of the 20 test samples was misclassified for all three methods in all 100 runs, except for one run for the OTh that has misclassified 5 samples. So for this dataset the three methods are almost equivalent with a 5% test error. However, the average number of informative genes used by the OTh is equal to one third of the number used in the STh. The total number of genes in this dataset is 2308 genes. The average

number of informative genes used by the OTh is 1.4% of the total number of genes. HTh used 4 more genes on average than the OTh.

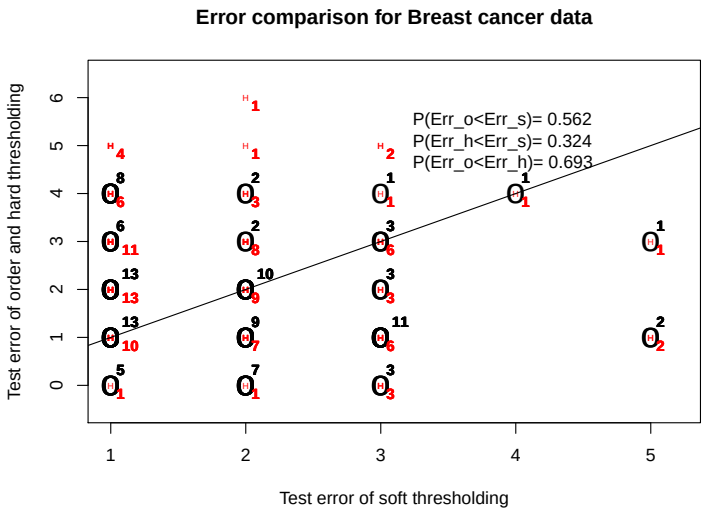


	Order	Hard	Soft
Mean test error	5.2	5	5
Median test error	5	5	5
Standard error of mean test error	0.2	0	0
Average number of active genes	32	36	94

Figure 3.1: *SRBCT analysis: Scatter plot of the STh test error vs HTh and OTh test errors from 100 runs. The plotting symbol H (in red) is for HTh and O (in black) is for OTh. The numbers used in the plot are the frequencies of test errors out of 100 runs. The table gives a summary of the percentage of test errors. One sample out of the 20 test samples was misclassified for all three methods in all 100 runs, except for one run in which OTh misclassified 5 samples. Average number of genes used in OTh is about 1/3 of that by STh. "NaN" in the plot means that STh and HTh have similar numbers of misclassified samples in all 100 runs.*

Figure 3.2 displays the result for the Breast cancer dataset analysis. The OTh has the smallest test error and has the smallest average number of informative genes. The HTh has the highest mean test error, but similar median test error to the STh. The STh selected the highest number of genes again. It used more than 3300 (36% of the total number of genes), while the OTh only used 7.4% of total number of genes to achieve even better performance. The STh has the smallest standard error of the mean test error. Then the OTh has the

second smallest standard error and the HTh has the largest standard error.

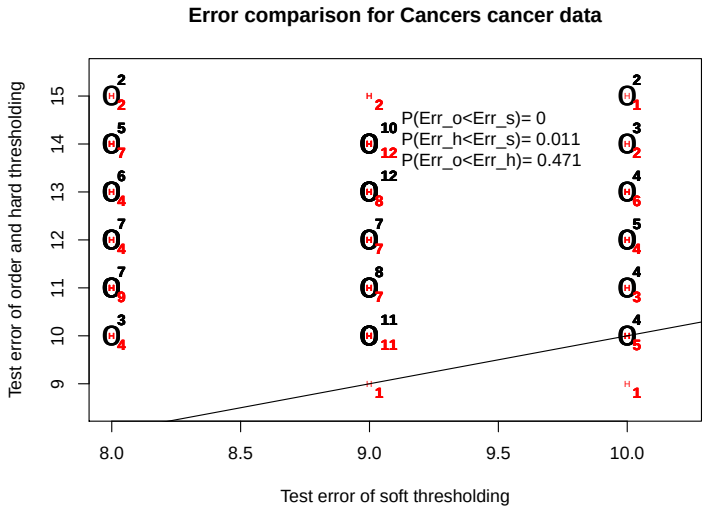


	Order	Hard	Soft
Mean test error	5.7	7.93	6.23
Median test error	5	6.67	6.67
Standard error of mean test error	0.41	0.45	0.33
Average number of active genes	679	1494	3317

Figure 3.2: *Breast cancer data analysis: Scatter plot of the STh test error vs HTh and OTh test errors from 100 runs. Count is used in the plot. The plotting symbol H (in red) is for HTh and O (in black) is for OTh. The numbers to the right of the plotting symbols report how many runs out of 100 has the plotted test error value. The table gives a summary of the percentage of misclassification in the 30 test samples. The probabilities given in the plot represents the proportion of times out of 100 runs that the test error for one thresholding method is smaller than another. The table reports a summary of the percent of misclassification error for test samples and average number of informative genes based on 100 runs.*

Figure 3.3 presents the result from the Cancers dataset analysis. This dataset contains different types of cancer samples; prostate, breast, lung, ovary, colorectum, kidney, liver, pancreas, bladder/ureter, and gastroesophagus. For this dataset the STh has the best performance in that it has the smallest mean test error, standard error for mean test error, and average number of informative genes. In fact, almost all test errors of STh in 100 runs reached the smallest of the three methods except in one run, in which the HTh has one

less misclassified sample than STh. All three methods used more than 1000 genes, but the OTh used 81 genes less than the HTh. The OTh and the HTh have the same median test error. The percentages of identified informative genes by the three methods are 8.9% with the STh, 11.8% with the OTh, and 12.4% with the HTh method.

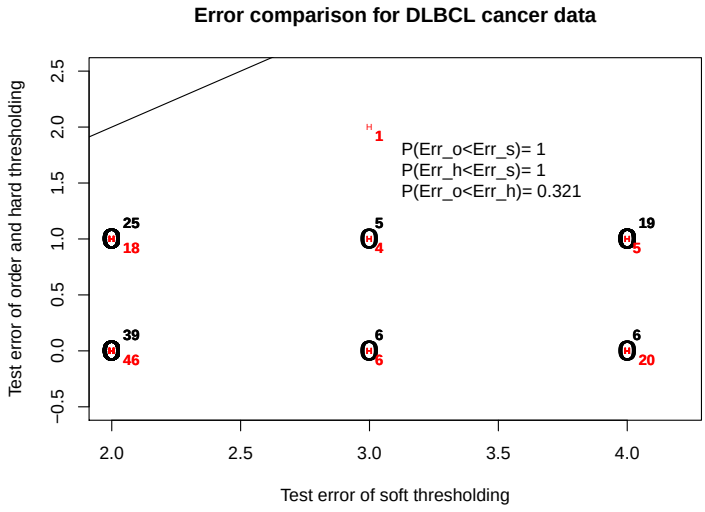


	Order	Hard	Soft
Mean test error	16.42	16.35	12.05
Median test error	16.22	16.22	12.16
Standard error of mean test error	0.2	0.22	0.1
Average number of active genes	1469	1548	1111

Figure 3.3: *Cancers analysis: Scatter plot of the STh test error vs HTh and OTh test errors from 100 runs. Count is used in the plot. The plotting symbol H (in red) is for HTh and O (in black) is for OTh. The numbers to the right of the plotting symbols report how many runs out of 100 has the plotted test error value. The table gives a summary of the percentage of misclassification in the 74 test samples. The probabilities given in the plot represents the proportion of times out of 100 runs that the test error for one thresholding method is smaller than another. The table reports a summary of the percent of misclassification error for test samples and average number of informative genes based on 100 runs.*

Moving to Figure 3.4, this analysis is for the Diffuse large B-cell lymphoma (DLBCL) dataset. In this case the OTh and the HTh always have less test errors than that for the STh in all 100 runs. Both OTh and HTh have zero median test error. Even though the HTh had a better mean test error and standard error, the OTh had the smallest average number

of selected genes. There is a very big difference in the number of selected genes between the OTh and the STh methods, as OTh selected 360 genes while STH selected 3483 genes.

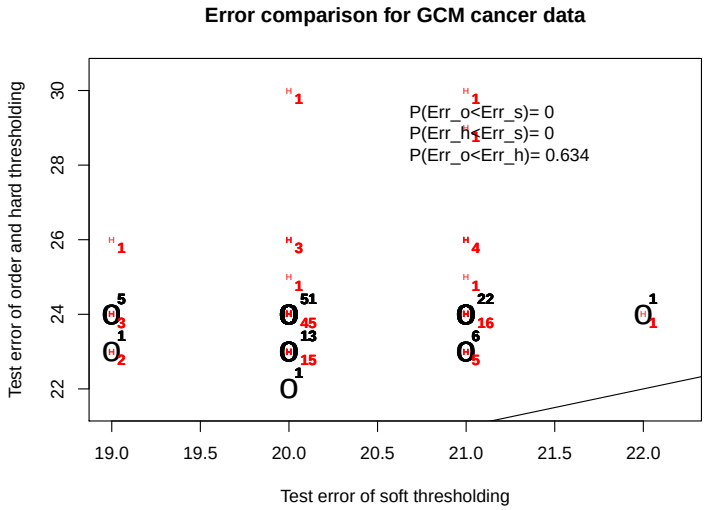


	Order	Hard	Soft
Mean test error	1.63	0.97	8.7
Median test error	0	0	6.67
Standard error of mean test error	0.17	0.16	0.29
Average number of active genes	360	716	3483

Figure 3.4: *DLBCL analysis: Scatter plot of the STh test error vs HTh and OTh test errors from 100 runs. Count is used in the plot. The plotting symbol H (in red) is for HTh and O (in black) is for OTh. The numbers to the right of the plotting symbols report how many runs out of 100 has the plotted test error value. The table gives a summary of the percentage of misclassification in the 30 test samples. The probabilities given in the plot represents the proportion of times out of 100 runs that the test error for one thresholding method is smaller than another. The table reports a summary of the percent of misclassification error for test samples and average number of informative genes based on 100 runs.*

On the other hand, we see the opposite result with the GCM dataset analysis given in Figure 3.5. The OTh and the HTh always have larger test errors than that for the STh in all 100 runs. This dataset is a collection of samples from 14 common human tumor types and it has the largest number of genes. In this analysis all three algorithms had the worst test error rate among all data sets in this study. In addition, all three methods used more than 2000 genes. The median test error was 43.48% for the STh and 52.17% for both OTh

and HTh. The OTh has the smallest standard error (0.1) while the HTh has the largest (0.27) standard error. The average number of selected genes ranged from 2010 for the STh to 3716 genes for the HTh method.



	Order	Hard	Soft
Mean test error	51.7	52.46	44
Median test error	52.17	52.17	43.48
Standard error of mean test error	0.1	0.27	0.12
Average number of active genes	2931	3716	2010

Figure 3.5: *GCM data analysis: Scatter plot of the STh test error vs HTh and OTh test errors from 100 runs. Count is used in the plot. The plotting symbol H (in red) is for HTh and O (in black) is for OTh. The numbers to the right of the plotting symbols report how many runs out of 100 has the plotted test error value. The table gives a summary of the percentage of misclassification in the 46 test samples. The probabilities given in the plot represents the proportion of times out of 100 runs that the test error for one thresholding method is smaller than another. The table reports a summary of the percent of misclassification error for test samples and average number of informative genes based on 100 runs.*

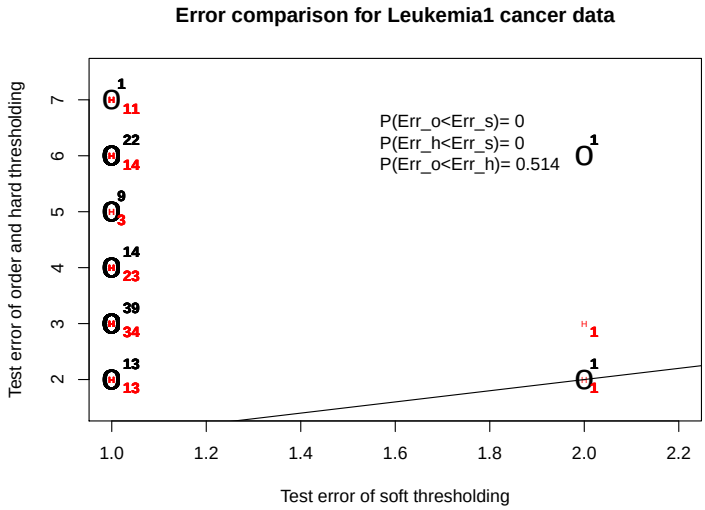
Figures 3.6 – 3.8 are for Leukemia cancer data sets. Even though all of them are for the same cancer, the results based on the three data sets are very different. This might be due to the following reasons: (1) The number of classes in these three data sets are different. There are 3 classes in Leukemia1 and Leukemia2 but there are 7 classes in Leukemia3 data. (2) The

training sample sizes are different (38, 57, 215 for Leukemia1, Leukemia2, and Leukemia3 respectively.) (3) The genes and the number of genes in the three data sets are different. Leukemia1 data used a much earlier version of Affymetrix GeneChip array that has 7129 genes. Leukemia2 and Leukemia3 used later versions of Affymetrix GeneChip array(s), one with 12582 genes and the other one with 12558 genes. In terms of accuracy, STh appears to be the best method out of the three for two of the data sets but has the worst performance in the remaining dataset. In terms of the average number of informative genes, however, the STh has the worst performance in two out of the three data sets. It is interesting to see that the number of genes that survived thresholding with the STh method show a clear association with the version of Affymetrix GeneChip array. In the earlier version (i.e. Leukemia1 data) STh has 111 genes survived while in the later version(s) more than 5300 genes survived thresholding.

For the Leukemia1 dataset, Figure 3.6 summarizes the result of this analysis. Here the STh has 3% mean test error and 111 average number of selected genes. Both values are less than those for either OTh or HTh. In all 100 runs, STh has the smallest test error among all 3 methods. The HTh and OTh have comparable performance in test errors but the OTh used less number of informative genes.

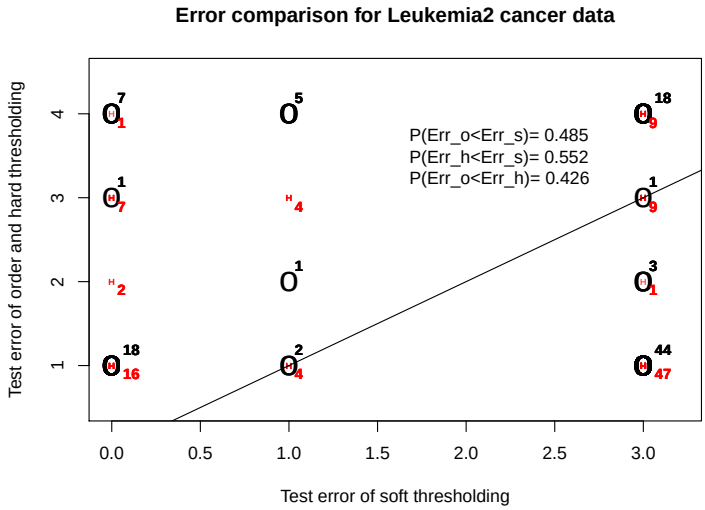
Figure 3.7 for Leukemia2 dataset shows STh has the worst performance among the three methods in that it not only has the largest average and median test errors but also has trouble in informative genes selection. The final model of STh kept on average 5389 genes, which is 16 times more than that used by OTh. The OTh has the smallest average number of selected genes (327). There is also a big difference in the number of selected genes for the HTh (1492) and STh (5389). The HTh has similar median test error of 6.67% to that for the OTh but smaller standard error. The median test error for the STH is 20%.

The analysis for the third Leukemia cancer dataset (Figure 3.8) shows that the STh has the least mean test error (1.11%) but with a very large average number of selected genes, 8637. This number of selected genes is the highest among all 10 data sets. OTh has an average of 5.01% test error with an average number of genes being 1156. HTh has mean



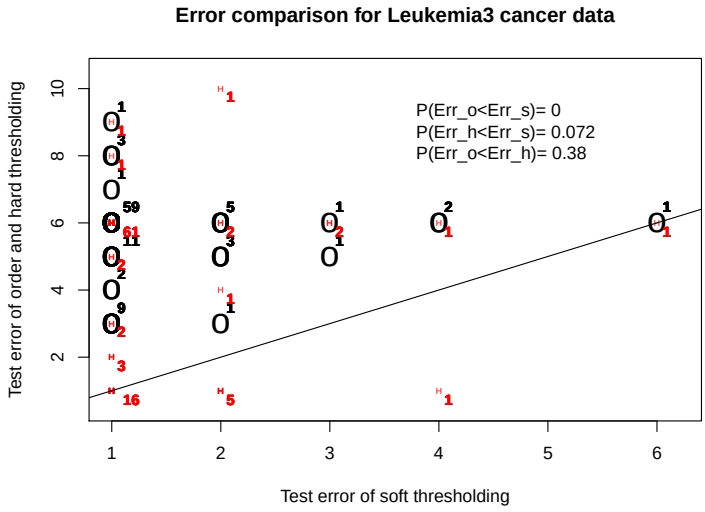
	Order	Hard	Soft
Mean test error	11.5	11.79	3
Median test error	8.82	11.76	2.94
Standard error of mean test error	0.42	0.47	0.04
Average number of active genes	139	149	111

Figure 3.6: *Leukemia1 cancer data analysis: Scatter plot of the STh test error vs HTh and OTh test errors from 100 runs. Count is used in the plot. The plotting symbol H (in red) is for HTh and O (in black) is for OTh. The numbers to the right of the plotting symbols report how many runs out of 100 has the plotted test error value. The table gives a summary of the percentage of misclassification in the 34 test samples. The probabilities given in the plot represents the proportion of times out of 100 runs that the test error for one thresholding method is smaller than another. The table reports a summary of the percent of misclassification error for test samples and average number of informative genes based on 100 runs.*



	Order	Hard	Soft
Mean test error	13.2	11.53	13.73
Median test error	6.67	6.67	20
Standard error of mean test error	0.91	0.73	0.89
Average number of active genes	327	1492	5389

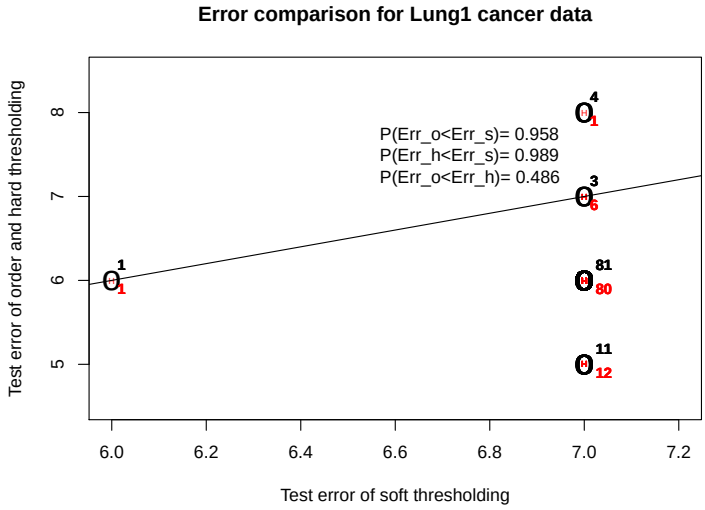
Figure 3.7: *Leukemia2 cancer data analysis: Scatter plot of the STh test error vs HTh and OTh test errors from 100 runs. Count is used in the plot. The plotting symbol H (in red) is for HTh and O (in black) is for OTh. The numbers to the right of the plotting symbols report how many runs out of 100 has the plotted test error value. The table gives a summary of the percentage of misclassification in the 15 test samples. The probabilities given in the plot represents the proportion of times out of 100 runs that the test error for one thresholding method is smaller than another. The table reports a summary of the percent of misclassification error for test samples and average number of informative genes based on 100 runs.*



	Order	Hard	Soft
Mean test error	5.01	4.26	1.11
Median test error	5.36	5.36	0.89
Standard error of mean test error	0.1	0.2	0.07
Average number of active genes	1156	2073	8637

Figure 3.8: *Leukemia3 cancer data analysis: Scatter plot of the STh test error vs HTh and OTh test errors from 100 runs. Count is used in the plot. The plotting symbol H (in red) is for HTh and O (in black) is for OTh. The numbers to the right of the plotting symbols report how many runs out of 100 has the plotted test error value. The table gives a summary of the percentage of misclassification in the 112 test samples. The probabilities given in the plot represents the proportion of times out of 100 runs that the test error for one thresholding method is smaller than another. The table reports a summary of the percent of misclassification error for test samples and average number of informative genes based on 100 runs.*

test error of 4.46% with average number of informative genes being 2073. On average the HTh has 0.75% less mean test error than the OTh with the price of using on average 917 more genes.

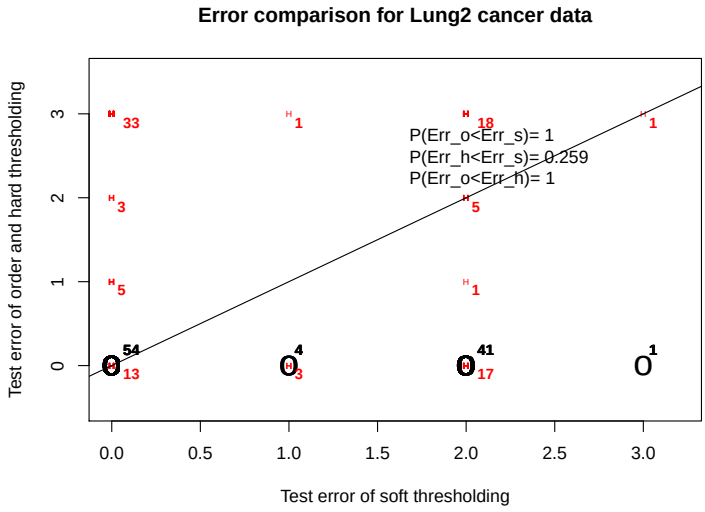


	Order	Hard	Soft
Mean test error	18.75	18.62	21.84
Median test error	18.75	18.75	21.88
Standard error of mean test error	0.17	0.15	0.03
Average number of active genes	87	134	50

Figure 3.9: *Lung1 cancer data analysis: Scatter plot of the STh test error vs HTh and OTh test errors from 100 runs. Count is used in the plot. The plotting symbol H (in red) is for HTh and O (in black) is for OTh. The numbers to the right of the plotting symbols report how many runs out of 100 has the plotted test error value. The table gives a summary of the percentage of misclassification in the 32 test samples. The probabilities given in the plot represents the proportion of times out of 100 runs that the test error for one thresholding method is smaller than another. The table reports a summary of the percent of misclassification error for test samples and average number of informative genes based on 100 runs.*

The last two data sets are for Lung cancer. The analysis of Lung1 dataset analysis in Figure 3.9 shows that OTh and HTh have equivalent performance in terms of the test error but OTh used less genes. The STh has the smallest average number of selected genes in this case. Figure 3.10 presents the analysis for the Lung2 dataset with best performance

achieved by OTh followed by the STh. In this case the OTh classified all test samples correctly in all 100 runs. The STh had the smallest average number of informative genes, 1911. The OTh used 2106 genes and the HTh used the highest number of genes (3910).



	Order	Hard	Soft
Mean test error	0	2.7	1.33
Median test error	0	4.48	0
Standard error of mean test error	0	0.21	0.15
Average number of active genes	2106	3610	1911

Figure 3.10: *Lung2 cancer data analysis: Scatter plot of the STh test error vs HTh and OTh test errors from 100 runs. Count is used in the plot. The plotting symbol H (in red) is for HTh and O (in black) is for OTh. The numbers to the right of the plotting symbols report how many runs out of 100 has the plotted test error value. The table gives a summary of the percentage of misclassification in the 67 test samples. The probabilities given in the plot represents the proportion of times out of 100 runs that the test error for one thresholding method is smaller than another. The table reports a summary of the percent of misclassification error for test samples and average number of informative genes based on 100 runs.*

3.2.1.2 Overall comparison based on all ten data sets

It can be seen from the previous section that none of the three algorithms (STh, OTh, and HTh) is absolutely the best across all ten data sets. In this section, we consider to combine the results from different data sets and provide an overall comparison. Specifically, we have the average percentage of misclassification errors for each method based on 100 runs for each cancer dataset. We also have the average number of informative genes from the 100 runs per method and dataset combination. Since the percent of misclassification errors are mostly small while the numbers of informative genes from different methods have drastically different ranges, a nonparametric approach without the assumption of constant variance and normality is more meaningful than a parametric method.

We will use a recent nonparametric approach proposed in Héberger (2010) and Héberger and Kollár-Hunek (2011) to do the comparison. This methods is called the Sum of Ranking Difference (SRD). It assumes that there is a golden standard. In our setting, the golden standard can be set to be the best performance out of all methods being compared. For example, if comparing methods based on their accuracy values then the maximum accuracy from all methods on each dataset is the best and hence the maximum is the golden standard. After computing the maximum accuracy among all methods for each experiment (or dataset), ranks of the maximum values is called the ideal ranking. The accuracy values of each method on different data sets are also ranked. Then the absolute values of the differences between the ideal ranking and the accuracy ranking for each method is computed. The SRD is the sum of the absolute differences. According to this approach, the method with the smaller SRD value is better than a method with bigger SRD value.

In our study, we first applied this method to the mean test errors to compare the three algorithms. We assume the golden standard to be the minimums of the mean test error across three algorithms for each dataset. Table 3.3 shows the SRD calculations based on the mean test errors from 100 runs on each method for all three methods. The minimums of the mean test error across three algorithms for each dataset, as our golden standard, are shown in the second column. The ranks of these values (i.e., ideal ranking) are on the

Table 3.3: *The SRD of mean test errors for the three thresholding methods.*

	Ideal		STh			OTh			HTh		
	Min	rnk	error	rnk1	diff1	error	rnk2	diff2	error	rnk3	diff3
Lung2	0.00	1	1.33	2	1	0	1	0	2.7	2	1
DLBCL	0.97	2	8.7	6	4	1.63	2	0	0.97	1	1
Leukemia3	1.11	3	1.11	1	2	5.01	3	0	4.26	3	0
Leukemia1	3.00	4	3	3	1	11.5	6	2	11.79	7	3
SRBCT	5.00	5	5	4	1	5.2	4	1	5	4	1
Breast	5.70	6	6.23	5	1	5.7	5	1	7.93	5	1
Leukemia2	11.53	7	13.73	8	1	13.2	7	0	11.53	6	1
Cancers	12.05	8	12.05	7	1	16.42	8	0	16.35	8	0
Lung1	18.62	9	21.84	9	0	18.75	9	0	18.62	9	0
GCM	44.00	10	44	10	0	51.7	10	0	52.46	10	0
					12			4			8

third column. The ranks of the mean test errors of each algorithm on different data sets are given in rnk1, rnk2, and rnk3 columns. The ranks for the OTH algorithm are similar to the ideal ranking except for three data sets. The absolute difference between each algorithm ranks and the ideal rank are those values in columns diff1, diff2, and diff3. The sum of those differences for each algorithm is the sum rank difference and it is given in the last row of the table. This result is presented in Figure 3.11. The OTh has the smallest sum rank difference (4); the HTh is in the middle with sum rank difference 8; and the STh has the largest sum rank difference value (12). This means that the OTh is the closest to our reference, the minimum mean test error. Therefore, according to the SRD method the OTh is the best algorithm in terms of the test error. The HTh is second and the STh is the least efficient algorithm. As our aim is to find the most efficient method generally over all data sets, and presumably for further (similar) data sets as well, the above sequence should be recommended for users of OTh, HTh and STh algorithms. When additional methods are included in comparison, the values of the golden standard will change (i.e. become of the new minimum errors across all methods). The recommendation needs to be reconsidered accordingly.

Beside the prediction accuracy of classifiers, identifying informative genes is very important for the researcher. This importance comes from the need to reduce the large number of irrelevant genes such that biologically important genes can be identified for targeted

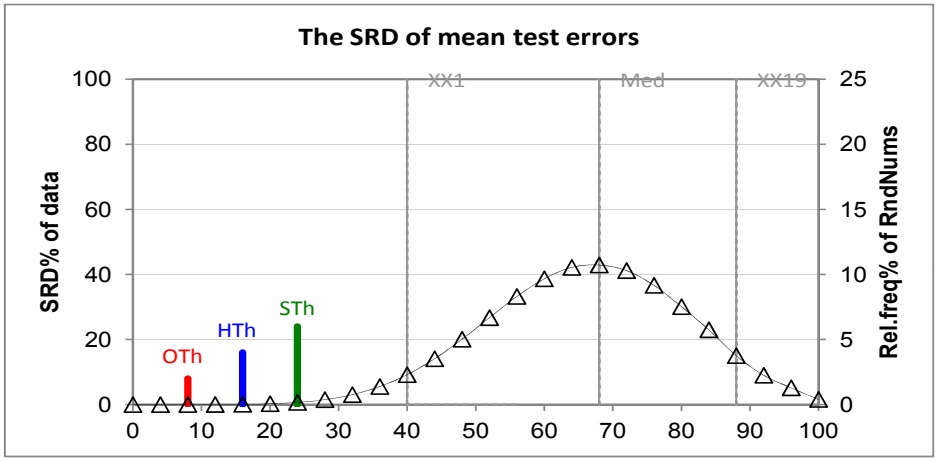


Figure 3.11: *The SRD comparison of mean test errors for the three thresholding methods. The x-axis and the left-side y-axis are the scaled SRD values. The right-side y-axis is the relative frequencies in percentages for the SRD theoretical distribution function.*

therapy. Moreover, smaller number of informative genes is convenient for followup studies. Hence, the number of informative genes identified by each thresholding method is another comparison criterion often used in literature. In Table 3.4 we listed the average number of informative genes selected over 100 runs of each algorithms for each dataset. OTh has the smallest overall average number of informative genes across all ten data sets (see the bottom row of Table 3.4). In addition, the OTh was the most consistent, in terms of the number of informative genes, compared to the other two algorithms. Its standard error of the average number of informative genes ranged from 0.9 to 47.5. While for the HTh it ranged from 1 to 258 and for the STh it ranged from 6 to 414.9. Even though the STh method identified a reasonable number of informative genes in some cases, it resulted in very large numbers in four cases Breast, DLBCL, Leukemia2, and Leukemia3. The HTh never had the minimum average number of informative genes. It had either the middle value or the largest value. In terms of overall average across all data sets as shown in the bottom row of Table 3.4, HTh is in between OTh and STh. We applied also the SRD approach to the average number of informative genes from 100 runs for each dataset to compare those three thresholding methods. As shown in Figure 3.12 the OTh and the HTh have tied SRD value that is much smaller than the SRD value for the STh.

Table 3.4: *Average number of informative genes based on 100 runs for each thresholding method. The value in parenthesis is the standard error.*

	STh	HTh	OTh
Lung1	50(6.0)	134(42.7)	87(16.5)
SRBCT	94(8.0)	36(1.0)	32(.9)
Leukemia1	111(50.0)	149(18.4)	139(12.0)
Cancers	1111(37.7)	1548(66.4)	1469(39.6)
Lung2	1911(169.3)	3610(88.1)	2106(38.0)
GCM	2010(89.9)	3716(212.9)	2931(33.9)
Breast	3317(152.3)	1494(132.5)	679(43.9)
DLBCL	3483(63.9)	716(55.4)	360(8.7)
Leukemia2	5389(414.9)	1492(258.1)	327(47.5)
Leukemia3	8637(208.9)	2073(254.5)	1156(38.6)
overall average	2611	1497	929

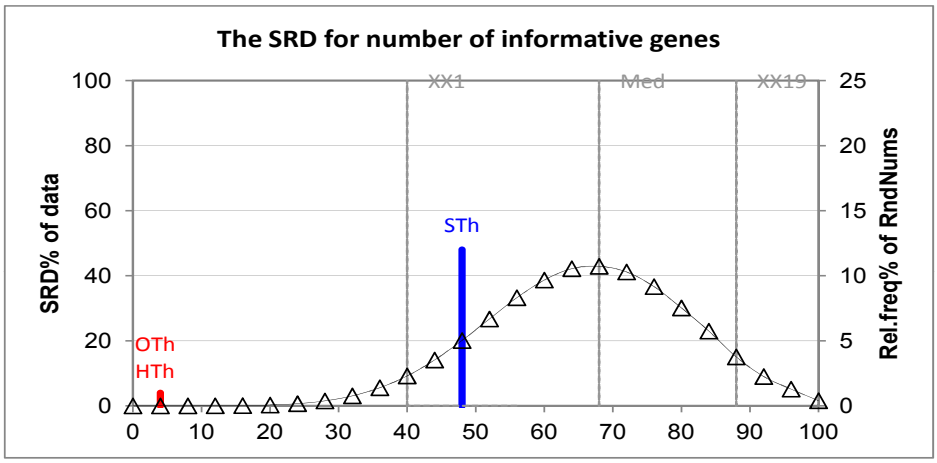


Figure 3.12: *The SRD comparison for the number of informative genes for the three thresholding methods. The x-axis and the left-side y-axis are the scaled SRD values. The right-side y-axis is the relative frequencies in percentages for the SRD theoretical distribution function.*

3.2.2 Performance of STh2, OTh2, and HTh2

STh2, OTh2, and HTh2 are the improved versions of the STh, HTh, and OTh, respectively. The deep search algorithm (3.1.2) is used for selecting the optimal thresholding parameter. It performs deep search in the neighborhood of the thresholding value for optimal thresholding parameter. The thresholding value corresponding to the second smallest cross-validation error is considered if its misclassification error differ from the smallest error by only one more sample and if it provides significant decrease in the number of selected genes.

In this section, we discuss the performance of those three algorithms: STh2, OTh2, and HTh2. Our analysis for this section is also for the 10 multi-class human cancers data sets listed in Table 3.2. In all that follows, our reported misclassification error refers to the percentage of misclassified test samples. The random partition of the training data in cross-validation could lead to different estimated thresholding parameters and hence possibly different test errors. So we repeated this process 100 times for each dataset.

The Sum of Ranking Difference (SRD) by Héberger (2010) will be used again in this section to compare the different algorithms. We will start by comparing the performance of the three algorithms that use the deep search: the STh2, HTh2, and OTh2. Figure 3.13 presents the results for the SRD of mean test errors for these three algorithms. The golden standard for the SRD method is assumed to be the minimums of the mean test error across three algorithms for each dataset. The OTh2 and HTh2 have the same sum rank difference and it is smaller than that for the STH2. This means that they are closer to the minimum mean test error than the STH2. Therefore, according to the SRD method the OTh2 and HTh2 are better algorithms in terms of the test error than the STh2. Comparing these results to those of STh, HTh, and OTh in Figure 3.11; it is clear that using the deep search algorithm reduced the SRD value for the STh2. Moreover, the differences between the SRD for the three algorithms that use the deep search are smaller.

The SRD results for the number of informative genes are presented in Figure 3.14. The golden standard for the SRD method assumed to be the minimums for the number of informative genes across three algorithms for each dataset. The HTh2 has the smallest sum

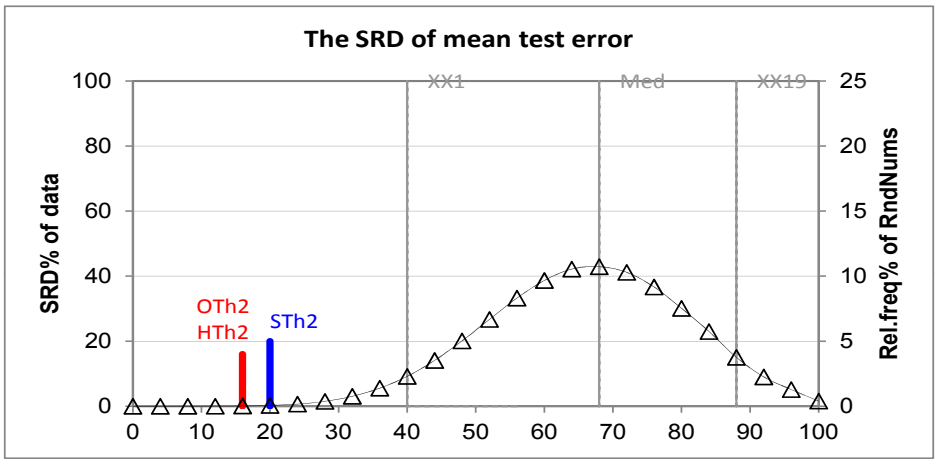


Figure 3.13: *The SRD comparison of mean test errors for the three algorithms STh2, OTh2, and HTh2. The x-axis and the left-side y-axis are the scaled SRD values. The right-side y-axis is the relative frequencies in percentages for the SRD theoretical distribution function.*

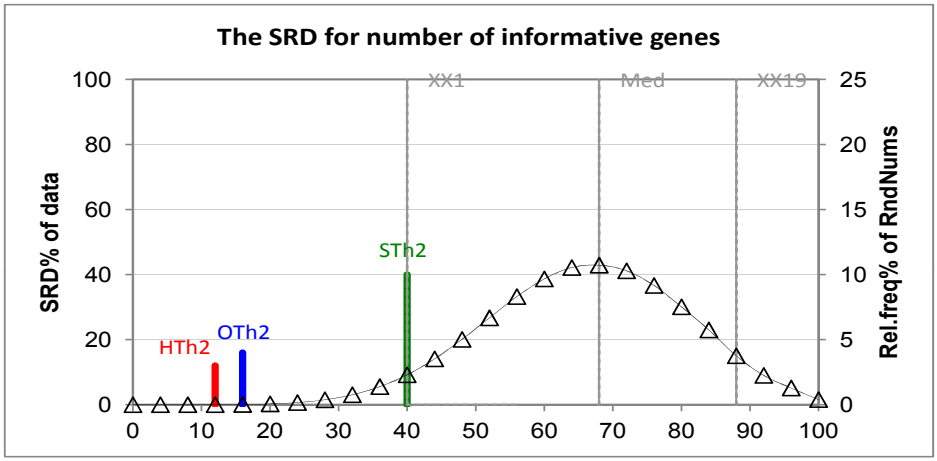


Figure 3.14: *The SRD comparison for the number of informative genes for the three algorithms STh2, OTh2, and HTh2. The x-axis and the left-side y-axis are the scaled SRD values. The right-side y-axis is the relative frequencies in percentages for the SRD theoretical distribution function.*

rank difference, which means that it is the closest to the minimum number of informative genes. Therefore, according to the SRD method the HTh2 is the best algorithm in terms of the number of informative genes. The OTh2 is in the middle and STh2 still has much larger SRD value. Comparing these results to those of STh, HTh, and OTh in Figure 3.12; we also noticed that using the deep search algorithm reduced the SRD value for the STh2 from that of STh.

Next we compare the performance of all six algorithms STh, OTh, HTh, STh2, OTh2, and HTh2. Figure 3.15 presents the results for the SRD of mean test errors for these six algorithms. Among all six algorithms, the OTh has the smallest sum rank difference, which means that it is the best algorithm in terms of the test error. The OTh2, HTh2 and HTh have tied second SRD value. The STh2 has the third largest SRD value and the largest SRD value was for the STh. Therefore, according to the SRD method the OTh is the best algorithm and the STh is the worst algorithm in terms of the test error if all six algorithms were compared.

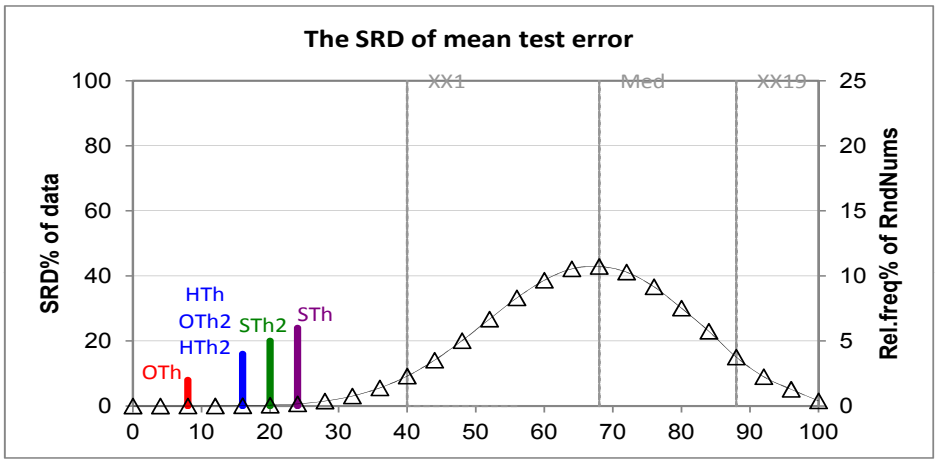


Figure 3.15: *The SRD comparison of mean test errors for all six algorithms STh, OTh, HTh, STh2, OTh2, and HTh2. The x-axis and the left-side y-axis are the scaled SRD values. The right-side y-axis is the relative frequencies in percentages for the SRD theoretical distribution function.*

The results of the SRD method for the number of informative genes for all six algorithms are presented in Figure 3.16. The interesting observation in this figure is that each one of

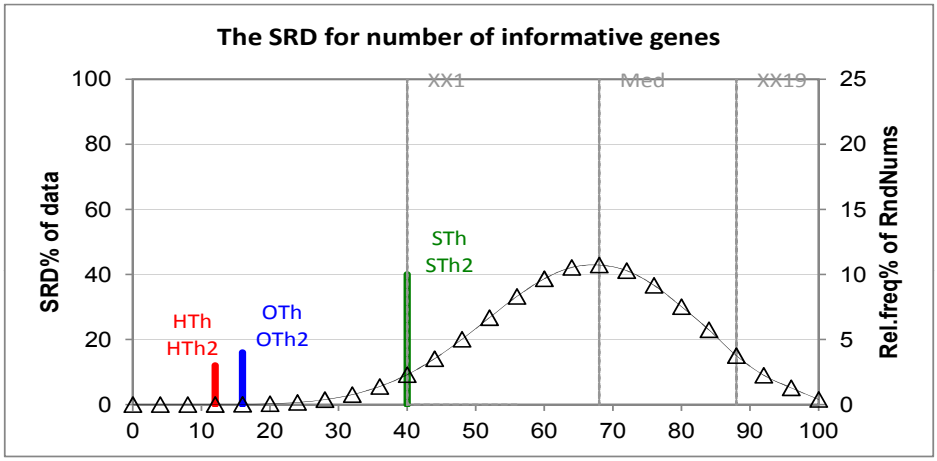


Figure 3.16: *The SRD comparison for the number of informative genes for all six algorithms STh, OTh, HTh, STh2, OTh2, and HTh2. The x-axis and the left-side y-axis are the scaled SRD values. The right-side y-axis is the relative frequencies in percentages for the SRD theoretical distribution function.*

the deep search algorithms has the same SRD value as its counterpart. The HTh2 and its counterpart HTh have the smallest sum rank difference. The OTh2 with OTh are in the middle and STh2 with STh have much larger SRD value.

For a closer view of the results of the algorithms that use the deep search and to compare them with their counterpart algorithms (STH, OTh, and HTh), Table 3.5 presents the mean test errors and the average number of informative genes based on 100 runs for each one of the six algorithms. The average number of selected genes by the STh2 were reduced from those by STh for all data sets. The mean test errors for the STh2 stayed almost the same as those for the STh except for the Leukemia2 dataset, for which STh2 has 7% less mean test error than its counterpart STh. For the Leukemia2 dataset, the average number of selected genes for the STh2 is 2236, while it was 5389 for those of STh. The average number of selected genes for the HTh2 was reduced from those of HTh for all data sets except for both Cancers and Leukemia1 data sets. The difference in mean test errors between HTh2 and its counterpart HTh is not more than 2% except for the Leukemia2 dataset (about 4%). In addition, the difference in mean test errors between OTh2 and its counterpart OTh is not more than 2% except for the Leukemia2 dataset (about 5%). Even though the

difference in average number of selected genes for the OTh2 and OTh are not as large as those between STh2 and STH or HTh2 and HTh there is still obvious reduction except for GCM, Leukemia1, and Lung1 data sets.

In conclusion, the deep search algorithm results in significant decrease in the number of selected genes for each method, while it kept the mean test errors barely changed. That is, the algorithms with deep search and their counterpart without deep search have similar test errors in that the difference in the test errors is no more than 2%.

Table 3.5: *The percent of mean misclassification error for test samples and average number of informative genes based on 100 runs for each thresholding method with and without the deep search algorithm.*

	STh		STh2		HTh		HTh2		OTh		OTh2	
	error	n.genes	error	n.genes	error	n.genes	error	n.genes	error	n.genes	error	n.genes
SRBCT	5	94	5	18	5	36	5	26	5.2	32	5	30
Breast	6.23	3317	6.33	2266	7.93	1494	5.57	549	5.7	679	6.87	371
Cancers	12.05	1111	12.31	956	16.35	1548	15.65	1631	16.42	1469	16.89	1360
DLBCL	8.7	3483	8.97	3399	0.97	716	0.83	491	1.63	360	1.83	250
GCM	44	2010	43.87	1692	52.46	3716	54.11	3709	51.7	2931	51.61	3009
Leukemia1	3	111	3.09	41	11.79	149	11.32	190	11.5	139	9.68	169
Leukemia2	13.73	5389	6.73	2236	11.53	1492	7.6	208	13.2	327	8.13	109
Leukemia3	1.11	8637	3.01	4606	4.26	2073	4.85	1943	5.01	1156	5.07	1020
Lung1	21.84	50	21.62	13	18.62	134	18.47	48	18.75	87	18.69	91
Lung2	1.33	1911	0.69	717	2.7	3610	2.06	3290	0	2106	0.01	2083

Chapter 4

The Optimal Thresholding Parameter Estimate and The Probability of Misclassification

4.1 The choice of the optimal thresholding parameter estimate

The appropriate choice of the thresholding parameter estimate is critical for the good performance of the classifiers. In Chapter 3 algorithms, cross-validation is used as a data-driven rule to select the thresholding parameter similar to the PAM algorithm. The estimated parameter minimizes the 10-fold cross-validation error over the training samples. In this section we examine three different thresholding parameter estimates that were suggested in literature. Assume that η is the number of values of interest before thresholding. One thresholding parameter estimate that we will be investigating in this section is the universal thresholding parameter $(2\log\eta)^{1/2}$, which is suggested by Donoho and Johnstone (1994) for both soft and hard thresholding. This universal thresholding parameter should asymptotically remove all the noise in case of independent Gaussian white noise sequence. Fan (1996) modified the universal thresholding to $(2\log(\eta\, a_\eta))^{1/2}$, with $a_\eta = c(\log\eta)^{-d}$ for some positive constants c and d. Upon considering the convergence rate for the test statistic, he recommended the values of $c = 1$ and $d = 2$. This choice of the thresholding parameter is

meant to remove most of the noise, but at the same time avoid filtering out all the important coefficients in the test statistics. On the other hand, the optimal order thresholding parameter was recommended to be estimated by $(\log \eta)^{3/2}$ in the simulations of Kim and Akritas (2010). The effectiveness of these three thresholding parameter estimates will be compared to the cross-validation parameter estimate through comparing the test error and the number of selected variables.

For our study in this section we will use the actual human cancers gene expression data sets listed in Table 3.2, except for Leukemia3 dataset. Leukemia3 dataset was removed from this analysis and all the following analyses in this dissertation because about 71% of its data values are zeros. All the data sets already divided by the authors of Tan et al. (2005) to training samples and test samples. Classifiers are trained with the training samples and then prediction of the class label for the test samples are conducted. The random partition of the training data in cross-validation could lead to different estimated thresholding parameter and hence possibly a different test samples prediction error. We repeated this process 100 times for each dataset in case of estimating the thresholding parameter using cross-validation.

Also, it is worth mentioning that previously in Chapter 3, we wrote our own codes for the HTh and OTh algorithms to calculate the class centroids, perform cross-validation using the training data, and to predict the class labels for the test samples. For a completely fair comparison, in the rest of the studies of this dissertation we code all the algorithms by modifying the functions from the **pamr** package, which was developed by the authors of Tibshirani et al. (2002). The 3 different thresholding methods in our study give 3 algorithms. The functions that we modified from the **pamr** package are: pamr.train, pamr.cv, pamr.predict, nsc, nsccv, and diag.disc. The soft.shrink function was replaced by the new hard.shrink function to perform hard thresholding or by the new order.shrink function to perform order thresholding. The refining process described at the end of Section 3.1.1 was also implemented in these algorithms. Specifically, this process refine the neighborhood of the thresholding value with the smallest cross-validation error to reach a better estimate of the optimal thresholding parameter. In all algorithms the number of folds for the cross-

validation with the training data is set to be 10 unless the dataset under study has some classes with sample size less than 10. In the later case, the fold is set to be the smallest class size. From now on, to refer to our codes from Chapter 3 we will add an asterisk (*) to the algorithm name. There is a little difference in some of the results of the codes of Chapter 3 and the codes from modifying the **pamr** package. This difference might be due to the different versions of R used which results in different random partition for the cross-validation. For Chapter 3 the available R version was 2.15.0 and for Chapter 4 the updated R version we used is 3.0.2. Table 4.1 list the results from the algorithms in both chapters.

Table 4.1: *Results based on our own code (Chapter 3) versus those based on modified pamr package (Chapter 4). The given results are the average for 100 runs of each algorithm with random partition of the training data in cross-validation.*

	STh*		STh		HTh*		HTh		OTh*		OTh	
	test	selec.	test	selec.	test	selec.	test	selec.	test	selec.	test	selec.
Dataset	error	genes	error	genes	error	genes	error	genes	error	genes	error	genes
SRBCT	5	94	5	110	5	36	5	40	5.2	32	5	48
Breast	6.23	3317	9.2	4312	7.93	1494	5	866	5.7	679	4.9	1233
Cancers	12.05	1111	11.97	1413	16.35	1548	12.01	1431	16.42	1469	11.84	1824
DLBCL	8.7	3483	8.2	3649	0.97	716	7.8	721	1.63	360	7.37	829
GCM	44	2010	44.17	2271	52.46	3716	54.59	4145	51.7	2931	54	3881
Leukemia1	3	111	3.24	299	11.79	149	13.29	94	11.5	139	12.06	179
Leukemia2	13.73	5389	15.13	6061	11.53	1492	11.73	1630	13.2	327	25.4	2506
Lung1	21.84	50	21.78	121	18.62	134	19.53	83	18.75	87	19.94	604
Lung2	1.33	1911	1.4	2303	2.7	3610	4.43	4275	0	2106	4.45	4419

The results for the soft (STH) and hard (HTh) thresholding algorithms are presented in Table 4.2. The thresholding parameters were estimated by either using cross-validation, the universal thresholding $(2\log(\eta))^{-1/2}$, or the modified universal thresholding $[2\log\left(\eta\log^{-2}\eta\right)]^{-1/2}$, where η in these estimates represent the number of test statistics before thresholding. Specifically, in our study η equals the number of genes multiplied by the number of classes.

In terms of the value of the estimated thresholding parameter, the modified universal

Table 4.2: *Percent of misclassification error for test samples (test error) and number of selected genes (selec. genes) for the soft thresholding algorithm (STh) and the hard thresholding algorithm (HTh). The thresholding parameters were estimated by using cross-validation (CV) based on 100 runs, the universal thresholding (Uni.)* $(2\log(\eta))^{-1/2}$*, or Fan (1996) modified universal thresholding (M.Uni.)* $[2\log\left(\eta\log^{-2}\eta\right)]^{-1/2}$.

	STh						HTh					
	CV		Uni.		M.Uni.		CV		Uni.		M.Uni.	
Dataset	test error	selec. genes	test error	selec. genes	test error	selec. genes	test error	selec. genes	test error	selec. genes	test error	selec. genes
SRBCT	5	110	5	62	5	150	5	40	5	62	0	150
Breast	9.2	4312	36.67	177	6.67	423	5	866	6.67	177	3.33	423
Cancers	11.97	1413	10.81	925	12.16	1732	12.01	1431	13.51	925	12.16	1732
DLBCL	8.2	3649	26.67	355	23.33	435	7.8	721	20	79	10	435
GCM	44.17	2271	43.48	3048	45.65	5112	54.59	4145	56.52	3048	58.7	5112
Leukemia1	3.24	299	2.94	66	2.94	191	13.29	94	11.76	66	8.82	191
Leukemia2	15.13	6061	20	410	20	1206	11.73	1630	26.67	410	26.67	1206
Lung1	21.78	121	21.88	99	21.88	210	19.53	83	18.75	99	21.88	210
Lung2	1.4	2303	0	1279	2.99	2312	4.43	4275	0	1270	1.49	2312
average	13.34		18.61		15.62		14.82		17.65		15.89	

thresholding is smaller than the universal thresholding to avoid filtering some of the important genes. This is clear in the results as the number of selected genes with the modified universal thresholding parameter estimate is larger than that using the universal thresholding parameter estimate. The thresholding parameter estimates with the cross-validation are smaller than those with the universal thresholding in 8 data sets for the STh and in 7 data sets for the HTh. They are even smaller than the modified universal estimate in four data sets.

In terms of the test error, the smallest average test error for both STh and HTh algorithms is achieved by the cross-validation estimate. The second smallest average test error is achieved by the modified universal thresholding parameter estimate. The universal thresholding has the largest average test error in both STh and HTh algorithms. The universal

thresholding in the STh algorithm has similar test errors to those of the modified universal thresholding in four data sets. Therefore, the general suggestion from this study is that using the cross-validation to estimate the thresholding parameter tend to give better result than that based on the universal or modified universal thresholding parameter estimates for both soft and hard thresholding algorithms.

Table 4.3: *Comparison of different thresholding parameter estimates for the order thresholding algorithm (OTh) based on percent of misclassification error for test samples (test error) and the number of selected genes (selec. genes). The thresholding parameters were estimated either using cross-validation based on 100 runs or Kim and Akritas's formula* $[\log \eta]^{3/2}$.

OTh				
	cross-validation		Kim and Akritas	
Dataset	test error	selec. genes	test error	selec. genes
SRBCT	5	48	5	28
Breast	4.9	1233	36.67	36
Cancers	11.84	1824	60.81	41
DLBCL	7.37	829	30	33
GCM	54	3881	67.39	44
Leukemia1	12.06	179	14.71	30
Leukemia2	25.4	2506	6.67	28
Lung1	19.94	604	21.88	30
Lung2	4.45	4419	11.94	36

The result for the order thresholding algorithm (OTh) is presented in Table 4.3. The thresholding parameter for this algorithm was estimated by either using cross-validation or Kim and Akritas (2010) estimate $(\log \eta)^{3/2}$, where η represent the number of test statistics before thresholding (i.e. the number of genes multiplied by the number of classes). It is very clear from the result in Table 4.3 that the thresholding parameter estimates based on Kim and Akritas (2010) are larger than the cross-validation parameter estimates for all data

sets. Therefore, the number of genes that survived thresholding with the Kim and Akritas (2010) parameter estimates are much smaller than that based on cross-validation estimates. Unfortunately, the small number of genes selected by Kim and Akritas (2010) parameter estimate did not yield better performance. Instead, using the cross-validation to estimate the order thresholding parameter results in a better performance for the OTh in most data sets. Leukemia2 is the only dataset that Kim and Akritas's thresholding parameter estimate results in smaller test error.

In conclusion, it can be seen from the results of this section that none of the four thresholding parameter estimates are absolutely the best in every single dataset. However, the overall comparison across all nine data sets is in favor of the thresholding parameter estimates obtained from cross-validation in all three algorithms, STh, HTh, and OTh.

4.2 Probability of misclassification

Motivated by Hall et al. (2008) who compared the theoretical performance of classifiers by obtaining their classification boundaries and following the foot steps of Fan and Fan (2008) who used the probability of misclassification to estimate the thresholding parameter, this section is devoted to deriving the probability of misclassification for the two classes case (binary classification). That is, the probability of misclassifying a sample from one class as coming from the other class. In the first part of this section we derive the probability of misclassification for the exact discriminant function. In the second part we derive the probability of misclassification for the PAM discriminant function under hard thresholding.

The discriminant function is derived from the posterior probability of the class label y given a set of variables $x^* = (x_1^*, x_2^*, ..., x_p^*)$. This posterior probability can be written as

$$\begin{aligned} P(y|x_1^*, x_2^*, ..., x_p^*) &= \frac{P(y)P(x_1^*, x_2^*, ..., x_p^*|y)}{P(x_1^*, x_2^*, ..., x_p^*)} \\ &= \frac{P(y)\prod_{i=1}^{p} P(x_i^*|y)}{P(x_1^*, x_2^*, ..., x_p^*)} \quad \text{(using the naive independence assumption)} \end{aligned} \tag{4.2.1}$$

Since the denominator is constant given the sample x^*, the comparison of the posterior probability for different classes will depend only on the numerator

$$P(y|x_1^*, x_2^*, ..., x_p^*) \propto P(y) \prod_{i=1}^{p} P(x_i^*|y).$$

Hence, the predicted class label (i.e. $\hat{y}$) is the one that achieves the highest posterior probability.

$$\hat{y} = \underset{y \in \{1,2,...,K\}}{argmax}\ P(y) \prod_{i=1}^{p} P(x_i^*|y).$$

As in PAM, assume that the variables given the class label have a Gaussian distribution with common variance among classes (i.e. $x_i^*|y = k \sim N[\mu_{ki}, \sigma_i^2]$). Under this assumption the posterior probability for class k (i.e. $y = k$) can be written as

$$\begin{aligned} P(y = k|x_1^*, x_2^*, ..., x_p^*) &\propto P(y = k) \prod_{i=1}^{p} P(x_i^*|y = k) \\ &\propto P(y = k) \prod_{i=1}^{p} \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left\{ \frac{-(x_i^* - \mu_{ki})^2}{2\sigma_i^2} \right\}. \end{aligned}$$

Take the logarithm for both sides,

$$\begin{aligned} \log\{P(y = k|x_1^*, x_2^*, ..., x_p^*)\} &\propto \log\{P(y = k)\} - \sum_{i=1}^{p} \log\left\{\sigma_i \sqrt{2\pi}\right\} - \sum_{i=1}^{p} \left\{ \frac{(x_i^* - \mu_{ki})^2}{2\sigma_i^2} \right\} \\ &\propto 2\log\{P(y = k)\} - \sum_{i=1}^{p} \left\{ \frac{(x_i^* - \mu_{ki})^2}{\sigma_i^2} \right\}. \end{aligned}$$

Since maximizing the log posterior probability is equivalent to minimizing the negative log posterior, the discriminant function for class k can be written as

$$\delta_k(x^*) = \sum_{i=1}^{p} \left\{ \frac{(x_i^* - \mu_{ki})^2}{\sigma_i^2} \right\} - 2\log(\pi_k), \tag{4.2.2}$$

where π_k denotes the class prior probability $P(y = k)$.

Therefore, the class with the smallest discriminant score (i.e. $\underset{k}{argmin}\ \delta_k(x^*)$) is the one that has the largest posterior probability given the sample x^*.

4.2.1 Probability of misclassification using the exact discriminant function

Considering the two classes case, without loss of generality we assume that the new sample $x^* = (x_1^*, x_2^*, ..., x_p^*)$ is from class 1 (i.e. $x_i^* \sim N[\mu_{1i}, \sigma_i^2]$) but is classified to class 2. That is, for sample x_i^*, class 2 has smaller discriminant score than class 1. Hence, the probability of misclassification using the discriminant function in (4.2.2) is

$$
\begin{aligned}
& P[\delta_2(x^*) < \delta_1(x^*)] \\
&= P[\delta_2(x^*) - \delta_1(x^*) < 0] \\
&= P\left[\sum_{i=1}^{p}\left\{\frac{(x_i^* - \mu_{2i})^2 - (x_i^* - \mu_{1i})^2}{\sigma_i^2}\right\} - 2\log\left(\frac{\pi_2}{\pi_1}\right) < 0\right] \\
&= P\left[\sum_{i=1}^{p}\left\{\frac{(2x_i^* - \mu_{2i} - \mu_{1i})(-\mu_{2i} + \mu_{1i})}{\sigma_i^2}\right\} - 2\log\left(\frac{\pi_2}{\pi_1}\right) < 0\right] \\
&= P\left[\sum_{i=1}^{p}\left\{\frac{(2x_i^* - \mu_{1i} + \mu_{1i} - \mu_{2i} - \mu_{1i})(\mu_{1i} - \mu_{2i})}{\sigma_i^2}\right\} - 2\log\left(\frac{\pi_2}{\pi_1}\right) < 0\right] \\
&= P\left[\sum_{i=1}^{p}\left\{\frac{([2x_i^* - 2\mu_{1i}] + [\mu_{1i} - \mu_{2i}])(\mu_{1i} - \mu_{2i})}{\sigma_i^2}\right\} - 2\log\left(\frac{\pi_2}{\pi_1}\right) < 0\right] \\
&= P\left[\sum_{i=1}^{p}\left\{\frac{2(x_i^* - \mu_{1i})(\mu_{1i} - \mu_{2i})}{\sigma_i^2}\right\} + \sum_{i=1}^{p}\left\{\frac{(\mu_{1i} - \mu_{2i})^2}{\sigma_i^2}\right\} - 2\log\left(\frac{\pi_2}{\pi_1}\right) < 0\right] \\
&= P\left[\sum_{i=1}^{p}\left\{\frac{(x_i^* - \mu_{1i})(\mu_{1i} - \mu_{2i})}{\sigma_i^2}\right\} < \log\left(\frac{\pi_2}{\pi_1}\right) - \frac{1}{2}\sum_{i=1}^{p}\left\{\frac{(\mu_{1i} - \mu_{2i})^2}{\sigma_i^2}\right\}\right] \\
&= P\left[\sum_{i=1}^{p}\left\{\frac{(x_i^* - \mu_{1i})}{\sigma_i} \cdot \frac{(\mu_{1i} - \mu_{2i})}{\sigma_i}\right\} < \log\left(\frac{\pi_2}{\pi_1}\right) - \frac{1}{2}\sum_{i=1}^{p}\left\{\frac{(\mu_{1i} - \mu_{2i})^2}{\sigma_i^2}\right\}\right] \quad (4.2.3)
\end{aligned}
$$

$(x_i^* - \mu_{1i})/\sigma_i$ is standard normal because x^* is from class 1 by assumption. Hence the summation on the left hand side of the inequality in (4.2.3) follows $N\left[0, \sum_{i=1}^{p}\frac{(\mu_{1i} - \mu_{2i})^2}{\sigma_i^2}\right]$

by using the independence assumption. Now by dividing both sides of the inequality by the standard deviation of the left hand side, the probability of misclassification becomes

$$= P\left[\frac{\sum_{i=1}^{p}\left\{\frac{(x_i^*-\mu_{1i})}{\sigma_i}\cdot\frac{(\mu_{1i}-\mu_{2i})}{\sigma_i}\right\}}{\sqrt{\sum_{i=1}^{p}\frac{(\mu_{1i}-\mu_{2i})^2}{\sigma_i^2}}} < \frac{\log\left(\frac{\pi_2}{\pi_1}\right)-\frac{1}{2}\sum_{i=1}^{p}\left\{\frac{(\mu_{1i}-\mu_{2i})^2}{\sigma_i^2}\right\}}{\sqrt{\sum_{i=1}^{p}\frac{(\mu_{1i}-\mu_{2i})^2}{\sigma_i^2}}}\right]$$

Therefore, the left hand side of the inequality is now standard normal and the probability of misclassification can be written in terms of the standard normal cumulative density function $\Phi(\cdot)$ as

$$\Phi\left[\frac{\log\left(\frac{\pi_2}{\pi_1}\right)-\frac{1}{2}\sum_{i=1}^{p}\left\{\frac{(\mu_{1i}-\mu_{2i})^2}{\sigma_i^2}\right\}}{\sqrt{\sum_{i=1}^{p}\frac{(\mu_{1i}-\mu_{2i})^2}{\sigma_i^2}}}\right] \tag{4.2.4}$$

From (4.2.4) we can clearly see that the probability of misclassification does not depend on the variables that have identical means for class 1 and 2. In other words, variables with equal class means contribute nothing to the classification process. This is why the T-test statistics is used in the PAM to filter out all these variables.

4.2.2 Probability of misclassification using PAM discriminant function with hard thresholding

Considering the two classes case, assume a given set of n training samples from 2 different classes and each is a vector with p variables, the single entry x_{ij} represents the value for variable i of sample j and y_j represents the class label for sample j. Without loss of generality we can assume the labels for the two classes are 1 and 2. Let n_k represent the number of samples from class k and C_k be the set of indices for those samples.

PAM estimates the parameters (π_k, μ_{ki}, and σ_i) in the discriminant function (4.2.2) from the training set. Specifically, π_k is estimated by the relative frequency $\hat{\pi}_k = n_k/n$, μ_{ik} is estimated by the class centroid $\overline{x}_i^{(k)} = \sum_{j\in C_k} x_{ij}/n_k$, and σ_i is estimated by the sample

pooled standard deviation $s_i = \sqrt{\sum_k \sum_{j \in C_k} (x_{ij} - \overline{x}_i^{(k)})^2/(n-2)}$. Then considering a given set of m variables that survived hard thresholding, the discriminant function of class k can be written as

$$\delta_k(x^*) = \sum_{i=1}^{m} \left\{ \frac{(x_i^* - \overline{x}_i^{\backprime(k)})^2}{(s_i + s_0)^2} \right\} - 2\log(\hat{\pi}_k), \tag{4.2.5}$$

where $\overline{x}_i^{\backprime(k)}$ is the shrunken centroid for class k, and $s_0 = median\{s_1, s_2, ..., s_p\}$.

In the following theorem we will give approximation to the probability of misclassification if this discriminant function (4.2.5) is used in the classification algorithm. Denote $\Phi(\cdot)$ and $\phi(\cdot)$ as the CDF and pdf of the standard normal distribution.

Theorem 4.2.1. *Without loss of generality, denote the selected m variables as $X_1, ..., X_m$. Suppose the selected m variables have mean $\mu_{1i}, \mu_{2i}, i = 1, ..., m$ in class 1 and 2, respectively. Assume $\mu_{1i}, \mu_{2i}, i = 1, ..., m$ are fixed values, such that $\mu_{1i} \neq \mu_{2i}$ for all $i \in \{1, ..., m\}$. Furthermore, assume $m = o(\min(n_1, n_2))$. Then the probability of misclassification for the two classes case using the above estimated discriminant function (4.2.5) of the hard thresholding algorithm is*

$$E\left[\Phi\left(\sqrt{\frac{n_1 n_2}{n}}\, C_{1n}^{-1/2}\left[\log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right) - a_n\right]\right)\right] + O\left(\frac{\sqrt{m}}{\sqrt{n}}\right) + O\left(\frac{m}{n_1}\right),$$

where $\hat{\pi}_1 = \dfrac{n_1}{n}$, $\hat{\pi}_2 = \dfrac{n_2}{n}$, $\sigma_0 = median\{\sigma_1, ..., \sigma_p\}$,

$$C_{1n} = \sum_{i=1}^{m} T_{ni}^2 \cdot \frac{\sigma_i^4}{(\sigma_i + \sigma_0)^4},$$

with $T_{ni} = \dfrac{\overline{x}_i^{(1)} - \overline{x}_i^{(2)}}{s_i\sqrt{\frac{n}{n_1 n_2}}}$, *i=1, ..., p independently distributed as noncentral t distribution with $n-2$ degrees of freedom and noncentrality parameter* $\frac{\mu_{1i} - \mu_{2i}}{\sigma_i\sqrt{\frac{n}{n_1 n_2}}}$,

and

$$a_n = \frac{1}{2}\sum_{i=1}^{m} T_{ni}^2 \cdot \left(\frac{\sigma_i}{\sigma_i + \sigma_0}\right)^2 \cdot \frac{n}{n_1 n_2}.$$

Proof of Theorem 4.2.1: Assume that the new sample $x^* = (x_1^*, x_2^*, ..., x_p^*)$ is from class 1 (i.e. $x_i^* \sim N[\mu_{1i}, \sigma_i^2]$), but it is classified to class 2 that has distribution $N[\mu_{2i}, \sigma_i^2]$ for variable x_i^*. In this case, for sample x_i^*, class 2 has smaller discriminant score than class 1. Hence, the probability of misclassification in this case is

$$
\begin{aligned}
&P[\delta_2(x^*) < \delta_1(x^*)] \\
&= P[\delta_2(x^*) - \delta_1(x^*) < 0] \\
&= P\left[\sum_{i=1}^{m}\left\{\frac{(x_i^* - \overline{x}_i^{\backslash(2)})^2 - (x_i^* - \overline{x}_i^{\backslash(1)})^2}{(s_i+s_0)^2}\right\} - 2\log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right) < 0\right] \\
&= P\left[\sum_{i=1}^{m}\left\{\frac{(2x_i^* - \overline{x}_i^{\backslash(2)} - \overline{x}_i^{\backslash(1)})(-\overline{x}_i^{\backslash(2)} + \overline{x}_i^{\backslash(1)})}{(s_i+s_0)^2}\right\} - 2\log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right) < 0\right] \\
&= P\left[\sum_{i=1}^{m}\left\{\frac{(x_i^* - 0.5\overline{x}_i^{\backslash(2)} - 0.5\overline{x}_i^{\backslash(1)})(\overline{x}_i^{\backslash(1)} - \overline{x}_i^{\backslash(2)})}{(s_i+s_0)^2}\right\} - \log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right) < 0\right] \\
&= P\left[\sum_{i=1}^{m}\left\{\frac{(x_i^* - \mu_{1i} + \mu_{1i} - 0.5[\overline{x}_i^{\backslash(2)} + \overline{x}_i^{\backslash(1)}])(\overline{x}_i^{\backslash(1)} - \overline{x}_i^{\backslash(2)})}{(s_i+s_0)^2}\right\} - \log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right) < 0\right].
\end{aligned}
$$

Denote $\hat{\tau}_i = \dfrac{(\overline{x}_i^{\backslash(1)} - \overline{x}_i^{\backslash(2)})}{(s_i+s_0)}$ and $\hat{\alpha}_i = 0.5[\overline{x}_i^{\backslash(2)} + \overline{x}_i^{\backslash(1)}]$. Then

$$
\begin{aligned}
P[\delta_2(x^*) \;<\; \delta_1(x^*)] \\
&= P\left[\sum_{i=1}^{m}\left\{\frac{(x_i^* - \mu_{1i} + \mu_{1i} - \hat{\alpha}_i)\hat{\tau}_i}{(s_i+s_0)}\right\} - \log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right) < 0\right] \\
&= P\left[\sum_{i=1}^{m}\left\{\frac{(x_i^* - \mu_{1i})\hat{\tau}_i}{(s_i+s_0)}\right\} + \sum_{i=1}^{m}\left\{\frac{(\mu_{1i} - \hat{\alpha}_i)\hat{\tau}_i}{(s_i+s_0)}\right\} - \log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right) < 0\right] \\
&= P\left[\sum_{i=1}^{m}\left\{\frac{(x_i^* - \mu_{1i})\hat{\tau}_i}{(s_i+s_0)}\right\} < \log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right) - \sum_{i=1}^{m}\left\{\frac{(\mu_{1i} - \hat{\alpha}_i)\hat{\tau}_i}{(s_i+s_0)}\right\}\right] \\
&= P\left[\sum_{i=1}^{m}\left\{\frac{(x_i^* - \mu_{1i})}{\sigma_i}\cdot\frac{\hat{\tau}_i\sigma_i}{(s_i+s_0)}\right\} < \log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right) - \sum_{i=1}^{m}\left\{\frac{(\mu_{1i} - \hat{\alpha}_i)\hat{\tau}_i}{(s_i+s_0)}\right\}\right]
\end{aligned}
$$

Since x^* is from class 1 by assumption, $(x_i^* - \mu_{1i})/\sigma_i$ follows standard normal distribution. So conditional on the observed data $X_1, ..., X_n$, the summation in the left hand side of the inequality follows $N\left[0, \sum_{i=1}^{m} \frac{\hat{\tau}_i^2 \sigma_i^2}{(s_i + s_0)^2}\right]$ by using the independence assumption. Now dividing both sides of the inequality by the conditional standard deviation of the left hand side, the probability of misclassification becomes

$$P\left[\frac{\sum_{i=1}^{m}\left\{\frac{(x_i^*-\mu_{1i})}{\sigma_i} \cdot \frac{\hat{\tau}_i\sigma_i}{(s_i+s_0)}\right\}}{\sqrt{\sum_{i=1}^{m}\frac{\hat{\tau}_i^2\sigma_i^2}{(s_i+s_0)^2}}} < \frac{\log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right) - \sum_{i=1}^{m}\left\{\frac{(\mu_{1i}-\hat{\alpha}_i)\hat{\tau}_i}{(s_i+s_0)}\right\}}{\sqrt{\sum_{i=1}^{m}\frac{\hat{\tau}_i^2\sigma_i^2}{(s_i+s_0)^2}}}\right]$$

Denote the ratio in the left hand side of the inequality as Z. Then given the n training samples, Z has standard normal distribution. Hence, the probability can be written as the expectation of the conditional probability on the given training samples as

$$E\left[P\left(Z < \frac{\log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right) - \sum_{i=1}^{m}\left\{\frac{(\mu_{1i}-\hat{\alpha}_i)\hat{\tau}_i}{(s_i+s_0)}\right\}}{\sqrt{\sum_{i=1}^{m}\frac{\hat{\tau}_i^2\sigma_i^2}{(s_i+s_0)^2}}} \,\middle|\, \text{training samples } (X_1, Y_1), ..., (X_n, Y_n)\right)\right]$$

$$= E\left[\Phi\left(\frac{\log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right) - \sum_{i=1}^{m}\left\{\frac{(\mu_{1i}-\hat{\alpha}_i)\hat{\tau}_i}{(s_i+s_0)}\right\}}{\sqrt{\sum_{i=1}^{m}\frac{\hat{\tau}_i^2\sigma_i^2}{(s_i+s_0)^2}}}\right)\right] \tag{4.2.6}$$

How is this expression related to the test statistic in PAM algorithm? Note that the test

statistics used in the PAM algorithm for variable selection is

$$d_{ki} = \frac{\overline{x}_i^{(k)} - \overline{x}_i}{(s_i + s_0)\sqrt{\frac{1}{n_k} - \frac{1}{n}}}, \tag{4.2.7}$$

where $\overline{x}_i = \sum_{k=1}^{2} \sum_{j=1}^{n_k} x_{ij}/n$.

For the two classes case, the test statistics in (4.2.7) can be written as

$$d_{ki} = \frac{\overline{x}_i^{(k)} - \frac{n_1\overline{x}_i^{(1)} + n_2\overline{x}_i^{(2)}}{n}}{(s_i + s_0)\sqrt{\frac{1}{n_k} - \frac{1}{n}}} = \frac{n\overline{x}_i^{(k)} - n_1\overline{x}_i^{(1)} - n_2\overline{x}_i^{(2)}}{n(s_i + s_0)\sqrt{\frac{1}{n_k} - \frac{1}{n}}}$$

$$= \begin{cases} \frac{n_2(\overline{x}_i^{(1)} - \overline{x}_i^{(2)})}{n(s_i+s_0)\sqrt{\frac{1}{n_1} - \frac{1}{n}}} & for \quad k = 1 \\ \\ \frac{n_1(\overline{x}_i^{(2)} - \overline{x}_i^{(1)})}{n(s_i+s_0)\sqrt{\frac{1}{n_2} - \frac{1}{n}}} & for \quad k = 2 \end{cases} = \begin{cases} \frac{\overline{x}_i^{(1)} - \overline{x}_i^{(2)}}{(s_i+s_0)\sqrt{\frac{n}{n_1 n_2}}} & for \quad k = 1 \\ \\ \frac{\overline{x}_i^{(2)} - \overline{x}_i^{(1)}}{(s_i+s_0)\sqrt{\frac{n}{n_1 n_2}}} & for \quad k = 2 \end{cases}$$

$$= \frac{(-1)^{k+1}(\overline{x}_i^{(1)} - \overline{x}_i^{(2)})}{(s_i + s_0)\sqrt{\frac{n}{n_1 n_2}}}.$$

Notice that in the two classes case $d_{1i}^{\setminus} = -d_{2i}^{\setminus}$. Hence, without loss of generality we can use the test statistic for the first class $d_{1i}^{\setminus} = \frac{\overline{x}_i^{\setminus(1)} - \overline{x}_i^{\setminus(2)}}{(s_i + s_0)\sqrt{\frac{n}{n_1 n_2}}} = \frac{\hat{\tau}_i}{\sqrt{\frac{n}{n_1 n_2}}}$ to rewrite the fraction in (4.2.6) so that the probability of misclassification is

$$E\left[\Phi\left(\frac{\log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right) - \sum_{i=1}^{m}\left\{\left(\frac{(\mu_{1i} - \overline{x}_i^{\setminus(1)})}{(s_i+s_0)} + \frac{1}{2}d_{1i}^{\setminus}\sqrt{\frac{n}{n_1 n_2}}\right) d_{1i}^{\setminus}\sqrt{\frac{n}{n_1 n_2}}\right\}}{\sqrt{\sum_{i=1}^{m} d_{1i}^{\setminus 2}\left(\frac{n}{n_1 n_2}\right)\frac{\sigma_i^2}{(s_i+s_0)^2}}}\right)\right] \tag{4.2.8}$$

To derive (4.2.8) from (4.2.6), first rewrite the summation in the numerator as

$$\begin{aligned}
\sum_{i=1}^{m}\left\{\frac{(\mu_{1i}-\hat{\alpha}_i)\hat{\tau}_i}{(s_i+s_0)}\right\} &= \sum_{i=1}^{m}\left\{\frac{(\mu_{1i}-0.5[\overline{x}_i^{\backprime(2)}+\overline{x}_i^{\backprime(1)}])(\overline{x}_i^{\backprime(1)}-\overline{x}_i^{\backprime(2)})}{(s_i+s_0)^2}\right\} \\
&= \sum_{i=1}^{m}\left\{\frac{(\mu_{1i}-0.5[\overline{x}_i^{\backprime(2)}+\overline{x}_i^{\backprime(1)}])}{(s_i+s_0)}\cdot\frac{(\overline{x}_i^{\backprime(1)}-\overline{x}_i^{\backprime(2)})}{(s_i+s_0)}\right\} \\
&= \sum_{i=1}^{m}\left\{\frac{1}{2}\cdot\frac{(2\mu_{1i}-\overline{x}_i^{\backprime(2)}-\overline{x}_i^{\backprime(1)})}{(s_i+s_0)}\cdot\frac{(\overline{x}_i^{\backprime(1)}-\overline{x}_i^{\backprime(2)})}{(s_i+s_0)}\right\} \\
&= \sum_{i=1}^{m}\left\{\frac{1}{2}\cdot\frac{(2\mu_{1i}-\overline{x}_i^{\backprime(1)}+\overline{x}_i^{\backprime(1)}-\overline{x}_i^{\backprime(2)}-\overline{x}_i^{\backprime(1)})}{(s_i+s_0)}\cdot\frac{(\overline{x}_i^{\backprime(1)}-\overline{x}_i^{\backprime(2)})}{(s_i+s_0)}\right\} \\
&= \sum_{i=1}^{m}\left\{\frac{1}{2}\cdot\frac{2(\mu_{1i}-\overline{x}_i^{\backprime(1)})+(\overline{x}_i^{\backprime(1)}-\overline{x}_i^{\backprime(2)})}{(s_i+s_0)}\cdot\frac{(\overline{x}_i^{\backprime(1)}-\overline{x}_i^{\backprime(2)})}{(s_i+s_0)}\right\} \\
&= \sum_{i=1}^{m}\left\{\left(\frac{(\mu_{1i}-\overline{x}_i^{\backprime(1)})}{(s_i+s_0)}+\frac{1}{2}\cdot\frac{(\overline{x}_i^{\backprime(1)}-\overline{x}_i^{\backprime(2)})}{(s_i+s_0)}\right)\frac{(\overline{x}_i^{\backprime(1)}-\overline{x}_i^{\backprime(2)})}{(s_i+s_0)}\right\} \\
&= \sum_{i=1}^{m}\left\{\left(\frac{(\mu_{1i}-\overline{x}_i^{\backprime(1)})}{(s_i+s_0)}+\frac{1}{2}d_{1i}^{\backprime}\sqrt{\frac{n}{n_1n_2}}\right)d_{1i}^{\backprime}\sqrt{\frac{n}{n_1n_2}}\right\} \qquad (4.2.9)
\end{aligned}$$

For the summation in the denominator

$$\begin{aligned}
\sum_{i=1}^{m}\frac{\hat{\tau}_i^2\sigma_i^2}{(s_i+s_0)^2} &= \sum_{i=1}^{m}\frac{(\overline{x}_i^{\backprime(1)}-\overline{x}_i^{\backprime(2)})^2\sigma_i^2}{(s_i+s_0)^4} \\
&= \sum_{i=1}^{m}\frac{(\overline{x}_i^{\backprime(1)}-\overline{x}_i^{\backprime(2)})^2}{(s_i+s_0)^2}\cdot\frac{\sigma_i^2}{(s_i+s_0)^2} \\
&= \sum_{i=1}^{m}d_{1i}^{\backprime 2}(\frac{n}{n_1n_2})\frac{\sigma_i^2}{(s_i+s_0)^2} \qquad (4.2.10)
\end{aligned}$$

Putting (4.2.9) and (4.2.10) into (4.2.6) gives us (4.2.8).

Back to (4.2.8), we extend the second term in the numerator and write the probability as

$$E\left[\Phi\left(\frac{\log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right)-\sum_{i=1}^{m}\left\{\frac{(\mu_{1i}-\bar{x}_i^{\setminus(1)})}{(s_i+s_0)}\cdot\hat{d}_{1i}\sqrt{\frac{n}{n_1n_2}}\right\}-\frac{1}{2}\sum_{i=1}^{m}\hat{d}_{1i}^2(\frac{n}{n_1n_2})}{\sqrt{\sum_{i=1}^{m}\hat{d}_{1i}^2(\frac{n}{n_1n_2})\frac{\sigma_i^2}{(s_i+s_0)^2}}}\right)\right]$$

$$= E\left[\Phi\left(\frac{\log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right)-\frac{1}{2}\sum_{i=1}^{m}\hat{d}_{1i}^2(\frac{n}{n_1n_2})}{\sqrt{\sum_{i=1}^{m}\hat{d}_{1i}^2(\frac{n}{n_1n_2})\frac{\sigma_i^2}{(s_i+s_0)^2}}}+\frac{\sum_{i=1}^{m}\left\{\frac{(\bar{x}_i^{\setminus(1)}-\mu_{1i})}{(s_i+s_0)}\cdot\hat{d}_{1i}\sqrt{\frac{n}{n_1n_2}}\right\}}{\sqrt{\sum_{i=1}^{m}\hat{d}_{1i}^2(\frac{n}{n_1n_2})\frac{\sigma_i^2}{(s_i+s_0)^2}}}\right)\right]$$

$$= E\left[\Phi\left(\frac{\log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right)-\frac{1}{2}\sum_{i=1}^{m}\hat{d}_{1i}^2(\frac{n}{n_1n_2})}{\sqrt{\sum_{i=1}^{m}\hat{d}_{1i}^2(\frac{n}{n_1n_2})\frac{\sigma_i^2}{(s_i+s_0)^2}}}+\frac{\sum_{i=1}^{m}\frac{(\bar{x}_i^{\setminus(1)}-\mu_{1i})}{(s_i+s_0)}\cdot\hat{d}_{1i}}{\sqrt{\sum_{i=1}^{m}\hat{d}_{1i}^2\frac{\sigma_i^2}{(s_i+s_0)^2}}}\right)\right]. \tag{4.2.11}$$

To simplify the writing in our next steps, let we refer to the first term in (4.2.11) by A and the second term by Δ_1, i.e. the misclassification probability is equal to $E\left[\Phi\left(A+\Delta_1\right)\right]$.

Now, Taylor expansion can be used to approximate the cumulative distribution function $\Phi(\cdot)$ such that

$$\begin{aligned} E\left[\Phi\left(A+\Delta_1\right)\right] &= E\left[\Phi(A)+\Delta_1\phi(A)+O_p(\Delta_1^2)\right] \\ &= E[\Phi(A)]+E[\Delta_1\phi(A)]+E[O_p(\Delta_1^2)] \end{aligned} \tag{4.2.12}$$

Note that $\phi(A)$ is a function of $d_{1i}^{\wedge 2}$, $i = 1, ..., m$. Therefore,

$$\begin{aligned} E[\Delta_1 \phi(A)] &= E[E(\Delta_1 \phi(A) \,|\, d_{1i}^{\wedge}, \, s_i + s_0)] \\ &= E[\phi(A) \cdot E(\Delta_1 \,|\, d_{1i}^{\wedge}, \, s_i + s_0)] \end{aligned}$$

$$= E\left[\phi(A) \cdot \frac{\sum_{i=1}^{m} \frac{E[(\overline{x}_i^{\setminus(1)} - \mu_{1i})|\, d_{1i}^{\wedge}]}{(s_i + s_0)} \cdot d_{1i}^{\wedge}}{\sqrt{\sum_{i=1}^{m} d_{1i}^{\wedge 2} \frac{\sigma_i^2}{(s_i + s_0)^2}}} \right] = 0,$$

and

$$\Delta_1^2 = \frac{\left(\sum_{i=1}^{m} \frac{(\overline{x}_i^{\setminus(1)} - \mu_{1i})}{(s_i + s_0)} \cdot d_{1i}^{\wedge} \right)^2}{\sum_{i=1}^{m} d_{1i}^{\wedge 2} \frac{\sigma_i^2}{(s_i + s_0)^2}} \leq \frac{\left(\sum_{i=1}^{m} \frac{(\overline{x}_i^{\setminus(1)} - \mu_{1i})^2}{(s_i + s_0)^2} \right) \cdot \left(\sum_{i=1}^{m} d_{1i}^{\wedge 2} \right)}{\sum_{i=1}^{m} d_{1i}^{\wedge 2} \frac{\sigma_i^2}{(s_i + s_0)^2}} = O_p(m/n_1),$$

where the inequality is due to Hölder's inequality.

Hence, after simplifying the last two terms in (4.2.12), the probability of misclassification is equal to

$$E[\Phi(A)] + O_p(m/n_1) \tag{4.2.13}$$

For the term A we first write the test statistic $d_{1i}^{\wedge}$ as

$$d_{1i}^{\wedge} = \frac{\overline{x}_i^{\setminus(1)} - \overline{x}_i^{\setminus(2)}}{(s_i + s_0)\sqrt{\frac{n}{n_1 . n_2}}} = \frac{\overline{x}_i^{\setminus(1)} - \overline{x}_i^{\setminus(2)}}{s_i \sqrt{\frac{n}{n_1 . n_2}}} \cdot \frac{s_i}{(s_i + s_0)} = T_{ni} \cdot \Delta_{ni}.$$

Then A can be written in terms of T_{ni} and Δ_{ni} as

$$A = \frac{\log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right) - \frac{1}{2}\sum_{i=1}^{m} d_{1i}^2(\frac{n}{n_1 n_2})}{\sqrt{\sum_{i=1}^{m} d_{1i}^2(\frac{n}{n_1 n_2})\frac{\sigma_i^2}{(s_i+s_0)^2}}} = \frac{\log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right) - \frac{1}{2}\sum_{i=1}^{m} T_{ni}^2\Delta_{ni}^2(\frac{n}{n_1 n_2})}{\sqrt{\sum_{i=1}^{m} T_{ni}^2\Delta_{ni}^2(\frac{n}{n_1 n_2})\frac{\sigma_i^2}{(s_i+s_0)^2}}}. \tag{4.2.14}$$

Denote the denominator as C_n

$$C_n^{-1} = \left[\sum_{i=1}^{m} T_{ni}^2\left(\frac{s_i}{s_i+s_0}\right)^2 \frac{n}{n_1 n_2}\frac{\sigma_i^2}{(s_i+s_0)^2}\right]^{-1/2}.$$

Let $g(s_i, s_0) = \dfrac{s_i^2}{(s_i+s_0)^4}$. Then consider the Taylor expansion of $g(\cdot,\cdot)$ at (σ_i, σ_0), where σ_0 =median$\{\sigma_i, i = 1, 2, ..., p\}$.

$$g(s_i, s_0) = g(\sigma_i, \sigma_0) + (s_i - \sigma_i)\left.\frac{\partial g(s_i, s_0)}{\partial s_i}\right|_{s_i=\sigma_i, s_0=\sigma_0} + (s_0 - \sigma_0)\left.\frac{\partial g(s_i, s_0)}{\partial s_0}\right|_{s_i=\sigma_i, s_0=\sigma_0}$$

$$+O_p((s_i-\sigma_i)^2) + O_p((s_0-\sigma_0)^2) + O_p((s_i-\sigma_i)(s_0-\sigma_0)).$$

Note that

$$\frac{\partial g(s_i, s_0)}{\partial s_i} = 2s_i(s_i+s_0)^{-4} + s_i^2(-4)(s_i+s_0)^{-5}$$

$$\left.\frac{\partial g(s_i, s_0)}{\partial s_i}\right|_{s_i=\sigma_i, s_0=\sigma_0} = \frac{2\sigma_i}{(\sigma_i+\sigma_0)^4} - \frac{4\sigma_i^2}{(\sigma_i+\sigma_0)^5}$$

and

$$\frac{\partial g(s_i, s_0)}{\partial s_0} = -4s_i^2(s_i+s_0)^{-5}$$

$$\left.\frac{\partial g(s_i, s_0)}{\partial s_0}\right|_{s_i=\sigma_i, s_0=\sigma_0} = \frac{-4\sigma_i^2}{(\sigma_i+\sigma_0)^5}.$$

We know that $s_i - \sigma_i = O_p(n^{-1/2})$, and $s_0 - \sigma_0 = O_p(n^{-1/2})$. So,

$$g(s_i, s_0) = g(\sigma_i, \sigma_0) + (s_i - \sigma_i)\beta_i + (s_0 - \sigma_0)\beta_{0i} + O_p(n^{-1}),$$

where $\beta_i = \dfrac{2\sigma_i}{(\sigma_i + \sigma_0)^4} - \dfrac{4\sigma_i^2}{(\sigma_i + \sigma_0)^5}$, and $\beta_{0i} = \dfrac{-4\sigma_i^2}{(\sigma_i + \sigma_0)^5}$.

Now,

$$\begin{aligned}
C_n^{-1} &= \sqrt{\frac{n_1 n_2}{n}} \left\{ \sum_{i=1}^{m} T_{ni}^2 \cdot g(s_i, s_0) \cdot \sigma_i^2 \right\}^{-1/2} \\
&= \sqrt{\frac{n_1 n_2}{n}} \left\{ \sum_{i=1}^{m} T_{ni}^2 \cdot g(\sigma_i, \sigma_0) \cdot \sigma_i^2 \right. \\
&\qquad \left. + \sum_{i=1}^{m} T_{ni}^2 \cdot [(s_i - \sigma_i)\beta_i + (s_0 - \sigma_0)\beta_{0i} + O_p(n^{-1})] \cdot \sigma_i^2 \right\}^{-1/2} \\
&= \sqrt{\frac{n_1 n_2}{n}} \left\{ C_{1n} + C_{2n} + \sum_{i=1}^{m} T_{ni}^2 \sigma_i^2 O_p(n^{-1}) \right\}^{-1/2},
\end{aligned}$$

where $C_{1n} = \sum_{i=1}^{m} T_{ni}^2 \cdot g(\sigma_i, \sigma_0) \cdot \sigma_i^2$, and $C_{2n} = \sum_{i=1}^{m} T_{ni}^2 \cdot [(s_i - \sigma_i)\beta_i + (s_0 - \sigma_0)\beta_{0i}]\sigma_i^2$.

Note that T_{ni}^2, i=1, ..., p are independently distributed as noncentral F distribution with $(1, n-2)$ degrees of freedom and noncentrality parameter

$$\lambda = \left(\frac{\mu_{1i} - \mu_{2i}}{\sigma_i \sqrt{\frac{n}{n_1 n_2}}} \right)^2 = \frac{n_1 n_2}{n} \left(\frac{\mu_{1i} - \mu_{2i}}{\sigma_i} \right)^2$$

So,

$$E(T_{ni}^2) = \frac{(n-2)(1+\lambda)}{n-4} = O\left(\frac{n_1 n_2}{n} \left(\frac{\mu_{1i} - \mu_{2i}}{\sigma_i} \right)^2 \right)$$

$$
\begin{aligned}
Var(T_{ni}^2) &= 2\frac{(1+\lambda)^2+(1+2\lambda)(n-4)}{(n-4)^2(n-6)} = O\left(\frac{n_1^2\,n_2^2}{n^3\,n^2}\right) + O\left(\frac{n_1\,n_2}{n\,n^2}\right) \\
&= O\left(\frac{n_1^2\,n_2^2}{n^5}\right) + O\left(\frac{n_1\,n_2}{n^3}\right) = o(1)
\end{aligned}
$$

Therefore, by Theorem 14.4-1 of Bishop et al. (2007), we know

$$
\begin{aligned}
\sum_{i=1}^{m} T_{ni}^2\sigma_i^2 &= O_p\left(\sum_{i=1}^{m} E\left[T_{ni}^2\right]\sigma_i^2\right) + O_p\left(\sqrt{\sum_{i=1}^{m} Var\left[T_{ni}^2\right]\sigma_i^4}\right) \\
&= O_p\left(\frac{n_1\,n_2}{n}\sum_{i=1}^{m}(\mu_{1i}-\mu_{2i})^2\right) = O_p\left(\frac{m\,n_1\,n_2}{n}\right).
\end{aligned}
$$

So,

$$
C_n^{-1} = \sqrt{\frac{n_1 n_2}{n}}\left\{C_{1n} + C_{2n} + O_p\left(\frac{m\,n_1\,n_2}{n^2}\right)\right\}^{-1/2}. \tag{4.2.15}
$$

Compared to C_{1n}, $C_{2n} + O_p\left(\frac{m\,n_1\,n_2}{n^2}\right)$ is negligible as $n \to \infty$. So, we can apply Taylor's expansion to $(x+\Delta x)^{-1/2}$ at $x_0 \neq 0$,

$$
(x+\Delta x)^{-1/2} = x^{-1/2} - \frac{1}{2}x^{-3/2}\Delta x + O(x^{-5/2}(\Delta x)^2).
$$

That is,

$$
\left[C_{1n} + C_{2n} + O_p\left(\frac{m\,n_1\,n_2}{n^2}\right)\right]^{-1/2}
$$

$$
= C_{1n}^{-1/2} - \frac{1}{2}\left[C_{2n}C_{1n}^{-3/2} + O_p\left(\frac{m\,n_1\,n_2}{n^2}\right)C_{1n}^{-3/2}\right] + O_p\left(C_{1n}^{-5/2}\left[C_{2n} + O_p\left(\frac{m\,n_1\,n_2}{n^2}\right)\right]^2\right)
$$

$$
= C_{1n}^{-1/2} - \frac{1}{2}C_{2n}C_{1n}^{-3/2} + O_p\left(\frac{m\,n_1\,n_2}{n^2}\cdot C_{1n}^{-3/2}\right)
$$

$$
+O_p\left(C_{1n}^{-5/2}\left[C_{2n}^2 + 2C_{2n}\frac{m\,n_1\,n_2}{n^2} + \frac{(m\,n_1\,n_2)^2}{n^4}\right]\right). \tag{4.2.16}
$$

Similar to the order of $\sum_{i=1}^{m} T_{ni}^2 \sigma_i^2$, it can be shown that

$$\begin{aligned} C_{1n} &= \sum_{i=1}^{m} T_{ni}^2 \cdot g(\sigma_i, \sigma_0) \cdot \sigma_i^2 \\ &= O_p\left(\sum_{i=1}^{m} E[T_{ni}^2] \cdot g(\sigma_i, \sigma_0) \cdot \sigma_i^2 \right) + O_p\left(\sqrt{\sum_{i=1}^{m} Var[T_{ni}^2] \cdot g(\sigma_i, \sigma_0) \cdot \sigma_i^2} \right) \\ &= O_p(\frac{m\, n_1\, n_2}{n}) \end{aligned}$$

$$\begin{aligned} C_{2n} &= \sum_{i=1}^{m} T_{ni}^2 (s_i - \sigma_i) \beta_i \sigma_i^2 + \sum_{i=1}^{m} T_{ni}^2 (s_0 - \sigma_0) \beta_{0i} \sigma_i^2 \\ &= \sum_{i=1}^{m} T_{ni}^2 \sigma_i^2 \beta_i O_p(n^{-1/2}) + \sum_{i=1}^{m} T_{ni}^2 \sigma_i^2 \beta_{0i} O_p(n^{-1/2}) \\ &= O_p\left(\frac{m\, n_1\, n_2}{n^{3/2}}\right) \end{aligned} \tag{4.2.17}$$

Therefore, putting (4.2.17) and (4.2.17) into (4.2.16) gives

$$\begin{aligned} &\left[C_{1n} + C_{2n} + O_p\left(\frac{m\, n_1\, n_2}{n^2}\right)\right]^{-1/2} \\ &= C_{1n}^{-1/2} - \frac{1}{2} C_{2n} C_{1n}^{-3/2} + O_p\left(\frac{m\, n_1\, n_2}{n^2} \cdot \left[\frac{m\, n_1\, n_2}{n}\right]^{-3/2} \right) \\ &\quad + O_p\left(\left(\frac{m\, n_1\, n_2}{n}\right)^{-5/2} \left[\frac{(m\, n_1\, n_2)^2}{n^3} + \frac{m\, n_1\, n_2}{\sqrt{n^3}} \cdot \frac{m\, n_1\, n_2}{n^2} + \frac{(m\, n_1\, n_2)^2}{n^4} \right] \right) \\ &= C_{1n}^{-1/2} - \frac{1}{2} C_{2n} C_{1n}^{-3/2} + O_p\left(\frac{1}{\sqrt{m\, n_1\, n_2\, n}} \right) \\ &\quad + O_p\left(\left(\frac{m\, n_1\, n_2}{n}\right)^{-1/2} \frac{1}{n} + \left(\frac{m\, n_1\, n_2}{n}\right)^{-1/2} \frac{1}{n^{3/2}} + \left(\frac{m\, n_1\, n_2}{n}\right)^{-1/2} \frac{1}{n^2} \right) \\ &= C_{1n}^{-1/2} - \frac{1}{2} C_{2n} C_{1n}^{-3/2} + O_p\left(\frac{1}{\sqrt{m\, n_1\, n_2\, n}} \right). \end{aligned} \tag{4.2.18}$$

Put (4.2.18) into (4.2.15). We have

$$C_n^{-1} = \sqrt{\frac{n_1 n_2}{n}}\, C_{1n}^{-1/2} - \frac{1}{2}\sqrt{\frac{n_1 n_2}{n}}\, C_{1n}^{-3/2} C_{2n} + O_p\left(\frac{1}{n\sqrt{m}}\right). \tag{4.2.19}$$

Next, use $\Delta_{ni} = \dfrac{s_i}{s_i + s_0} = \dfrac{\sigma_i + O_p(n^{-1/2})}{\sigma_i + \sigma_0 + O_p(n^{-1/2})} = \dfrac{\sigma_i}{\sigma_i + \sigma_0} + O_p(n^{-1/2})$ to rewrite the numerator of A.

$$\begin{aligned}
&\log\left(\frac{\hat\pi_2}{\hat\pi_1}\right) - \frac{1}{2}\sum_{i=1}^{m} T_{ni}^2 \Delta_{ni}^2 \left(\frac{n}{n_1 n_2}\right) \\
&= \log\left(\frac{\hat\pi_2}{\hat\pi_1}\right) - \frac{1}{2}\sum_{i=1}^{m} T_{ni}^2 \left(\frac{\sigma_i}{\sigma_i + \sigma_0} + O_p(n^{-\frac{1}{2}})\right)^2 \left(\frac{n}{n_1 n_2}\right) \\
&= \log\left(\frac{\hat\pi_2}{\hat\pi_1}\right) - \frac{1}{2}\sum_{i=1}^{m} T_{ni}^2 \left(\frac{\sigma_i}{\sigma_i + \sigma_0}\right)^2 \left(\frac{n}{n_1 n_2}\right) + O_p\left(\frac{m\, n_1\, n_2}{n\sqrt{n}} \cdot \frac{n}{n_1 n_2}\right) \\
&= \log\left(\frac{\hat\pi_2}{\hat\pi_1}\right) - a_n + O_p\left(\frac{m}{\sqrt{n}}\right),
\end{aligned} \tag{4.2.20}$$

where $a_n = \frac{1}{2}\sum_{i=1}^{m} T_{ni}^2 \left(\frac{\sigma_i}{\sigma_i + \sigma_0}\right)^2 \left(\frac{n}{n_1 n_2}\right)$.

Therefore, putting (4.2.19) and (4.2.20) into (4.2.14) gives

$$A = \left[\sqrt{\frac{n_1 n_2}{n}}\, C_{1n}^{-1/2} - \frac{1}{2}\sqrt{\frac{n_1 n_2}{n}}\, C_{1n}^{-3/2} C_{2n} + O_p\left(\frac{1}{n\sqrt{m}}\right)\right]\left[\log\left(\frac{\hat\pi_2}{\hat\pi_1}\right) - a_n + O_p\left(\frac{m}{\sqrt{n}}\right)\right]$$

$$= \sqrt{\frac{n_1 n_2}{n}}\, C_{1n}^{-1/2}\left[\log\left(\frac{\hat\pi_2}{\hat\pi_1}\right) - a_n\right] - \frac{1}{2}\sqrt{\frac{n_1 n_2}{n}}\, C_{1n}^{-3/2} C_{2n} \log\left(\frac{\hat\pi_2}{\hat\pi_1}\right) \tag{4.2.21}$$

$$+\frac{1}{2}\sqrt{\frac{n_1 n_2}{n}}\, C_{1n}^{-3/2} C_{2n}\, a_n + \sqrt{\frac{n_1 n_2}{n}}\, C_{1n}^{-1/2} O_p\left(\frac{m}{\sqrt{n}}\right) \tag{4.2.22}$$

$$-\frac{1}{2}\sqrt{\frac{n_1 n_2}{n}} O_p\left(C_{1n}^{-3/2} C_{2n} \frac{m}{\sqrt{n}}\right) + O_p\left(\frac{1}{n\sqrt{m}} \cdot \frac{m}{\sqrt{n}}\right) \tag{4.2.23}$$

$$+O_p\left(\frac{1}{n\sqrt{m}} \log\left(\frac{\hat\pi_2}{\hat\pi_1}\right)\right) - O_p\left(\frac{1}{n\sqrt{m}}\, a_n\right). \tag{4.2.24}$$

Note that $a_n = O_p(m)$. (4.2.25)

By (4.2.17), (4.2.17), and (4.2.25) we know that:

The first term in (4.2.22) is equal to $O_p\left(\sqrt{\frac{n_1 n_2}{n}} \cdot (\frac{m\, n_1\, n_2}{n})^{-3/2} \cdot \frac{m\, n_1\, n_2}{n\sqrt{n}} \cdot m\right) = O_p\left(\frac{\sqrt{m}}{\sqrt{n}}\right)$.

The second term in (4.2.22) is equal to $O_p\left(\sqrt{\frac{n_1 n_2}{n}} \cdot (\frac{m\, n_1\, n_2}{n})^{-1/2} \cdot \frac{m}{\sqrt{n}}\right) = O_p\left(\frac{\sqrt{m}}{\sqrt{n}}\right)$

The first term in (4.2.23) is equal to $O_p\left(\sqrt{\frac{n_1 n_2}{n}} \cdot (\frac{m\, n_1\, n_2}{n})^{-3/2} \cdot \frac{m\, n_1\, n_2}{n\sqrt{n}} \cdot \frac{m}{\sqrt{n}}\right) = O_p\left(\frac{\sqrt{m}}{n}\right)$.

The second term in (4.2.23) is equal to $O_p\left(\frac{\sqrt{m}}{n\sqrt{n}}\right)$.

The first term in (4.2.24) is equal to $O_p\left(\frac{1}{n\sqrt{m}} \log\left(\frac{n_2}{n_1}\right)\right)$.

The second term in (4.2.24) is equal to $O_p\left(\frac{1}{n\sqrt{m}} \cdot m\right) = O_p\left(\frac{\sqrt{m}}{n}\right)$.

Therefore,

$$\begin{aligned} A &= \sqrt{\frac{n_1 n_2}{n}}\, C_{1n}^{-1/2} \left[\log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right) - a_n\right] - \frac{1}{2}\sqrt{\frac{n_1 n_2}{n}}\, C_{1n}^{-3/2} C_{2n} \log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right) \\ &\quad + O_p\left(\frac{\sqrt{m}}{\sqrt{n}}\right) + O_p\left(\frac{1}{n\sqrt{m}} \log\left(\frac{n_2}{n_1}\right)\right). \end{aligned}$$

Denote the first two terms of A by b, that is

$$b = \sqrt{\frac{n_1 n_2}{n}}\, C_{1n}^{-1/2} \left[\log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right) - a_n\right] - \frac{1}{2}\sqrt{\frac{n_1 n_2}{n}}\, C_{1n}^{-3/2} C_{2n} \log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right)$$

Apply Taylor's expansion to $\Phi(x)$ at $x_0 = b$, $\Phi(x) = \Phi(x_0) + (x - x_0)\phi(x_0) + o(x - x_0)$.

$$\Phi(A) = \Phi(b) + O_p\left(\frac{\sqrt{m}}{\sqrt{n}}\right) + O_p\left(\frac{1}{n\sqrt{m}} \log\left(\frac{n_2}{n_1}\right)\right).$$

Then,

$$E\left[\Phi(A)\right] = E\left[\Phi(b)\right] + O\left(\frac{\sqrt{m}}{\sqrt{n}}\right) + O\left(\frac{1}{n\sqrt{m}}\log\left(\frac{n_2}{n_1}\right)\right). \tag{4.2.26}$$

To Calculate $E\left[\Phi(b)\right]$, note that the second term in b is of smaller order than the first term. The order of the second term is

$$O_p\left(\sqrt{\frac{n_1 n_2}{n}}\cdot\log\left(\frac{n_2}{n_1}\right)\cdot\left(\frac{m\,n_1\,n_2}{n}\right)^{-3/2}\cdot\frac{m\,n_1\,n_2}{n\sqrt{n}}\right) = O_p\left(\frac{1}{\sqrt{m\,n}}\log\left(\frac{n_2}{n_1}\right)\right) = o_p(1).$$

Apply Taylor's expansion to $\Phi(b)$ again, we get

$$\begin{aligned}
\Phi(b) &= \Phi\left(\sqrt{\frac{n_1 n_2}{n}}\,C_{1n}^{-1/2}\left[\log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right) - a_n\right]\right) \\
&\quad -\frac{1}{2}\sqrt{\frac{n_1 n_2}{n}}\cdot\log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right)C_{1n}^{-3/2}C_{2n}\cdot\phi\left(\sqrt{\frac{n_1 n_2}{n}}\,C_{1n}^{-1/2}\left[\log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right) - a_n\right]\right) \\
&\quad +O_p\left(\left[\sqrt{\frac{n_1 n_2}{n}}\,C_{1n}^{-3/2}C_{2n}\log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right)\right]^2\right)\phi'\left(\sqrt{\frac{n_1 n_2}{n}}\,C_{1n}^{-1/2}\left[\log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right) - a_n\right]\right).
\end{aligned}$$

The last term in $\Phi(b)$ is of order

$$O_p\left(\frac{n_1 n_2}{n}\cdot\left[\log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right)\right]^2\cdot\left(\frac{m\,n_1\,n_2}{n}\right)^{-3}\cdot\frac{(m\,n_1\,n_2)^2}{n^3}\right) = O_p\left(\frac{1}{nm}\cdot\left[\log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right)\right]^2\right) = o_p(1).$$

Therefore, $E\left[\Phi(b)\right]$

$$= E\left[\Phi\left(\sqrt{\frac{n_1 n_2}{n}}\,C_{1n}^{-1/2}\left[\log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right) - a_n\right]\right)\right] \tag{4.2.27}$$

$$-\frac{1}{2}\sqrt{\frac{n_1 n_2}{n}}\cdot\log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right)\cdot E\left[\frac{C_{2n}}{C_{1n}^{3/2}}\cdot\phi\left(\sqrt{\frac{n_1 n_2}{n}}\,C_{1n}^{-1/2}\left[\log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right) - a_n\right]\right)\right] \tag{4.2.28}$$

$$+O_p\left(\frac{1}{nm}\cdot\left[\log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right)\right]^2\right). \tag{4.2.29}$$

To calculate (4.2.28), note that C_{1n} and a_n are both functions of $T_{ni}^2, i = 1, 2, ..., p$. C_{2n} is the only term that is a function of both $T_{ni}^2, i = 1, 2, ..., p$ and $s_i, s_0, i = 1, 2, ..., p$.

So, the term (4.2.28) can be written as

$$E\left[E\left(-\frac{1}{2}\sqrt{\frac{n_1n_2}{n}}\cdot\log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right)\cdot\frac{C_{2n}}{C_{1n}^{3/2}}\cdot\phi\left(\sqrt{\frac{n_1n_2}{n}}\,C_{1n}^{-1/2}\left[\log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right)-a_n\right]\right)\,\middle|\,T_{n1},...,T_{np}\right)\right]$$

The order of

$$-\frac{1}{2}\sqrt{\frac{n_1n_2}{n}}\cdot\log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right)\cdot\frac{C_{2n}}{C_{1n}^{3/2}} \quad \text{is} \quad O\left(\frac{1}{\sqrt{nm}}\log\left(\frac{n_2}{n_1}\right)\right)=o_p(1).$$

Since $\phi(\cdot)$ is bounded in probability, we known the term in (4.2.28) is of order

$$O\left(\frac{1}{\sqrt{nm}}\log\left(\frac{n_2}{n_1}\right)\right). \tag{4.2.30}$$

Finally, putting (4.2.13), (4.2.26), (4.2.27-4.2.29), and (4.2.30) all together, we can write the probability of misclassification as

$$\begin{aligned}
& E[\Phi(A+\Delta_1)] = E[\Phi(A)] + O(m/n_1) \\
= \;& E[\Phi(b)] + O\left(\frac{\sqrt{m}}{\sqrt{n}}\right) + O\left(\frac{1}{n\sqrt{m}}\cdot\log\left(\frac{n_2}{n_1}\right)\right) + O\left(\frac{m}{n_1}\right) \\
= \;& E\left[\Phi\left(\sqrt{\frac{n_1n_2}{n}}\,C_{1n}^{-1/2}\left[\log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right)-a_n\right]\right)\right] \\
& + O\left(\frac{1}{\sqrt{nm}}\cdot\log\left(\frac{n_2}{n_1}\right)\right) + O\left(\frac{\sqrt{m}}{\sqrt{n}}\right) + O\left(\frac{1}{n\sqrt{m}}\cdot\log\left(\frac{n_2}{n_1}\right)\right) + O\left(\frac{m}{n_1}\right) \\
= \;& E\left[\Phi\left(\sqrt{\frac{n_1n_2}{n}}\,C_{1n}^{-1/2}\left[\log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right)-a_n\right]\right)\right] + O\left(\frac{\sqrt{m}}{\sqrt{n}}\right) + O\left(\frac{m}{n_1}\right).\square
\end{aligned}$$

The approximation expression of the probability of misclassification in Theorem 4.2.1 clearly shows how the signal and the noise affect the misclassification error as the number

of selected variables m increases. In particular, as m increases the expectation in the first term of the approximation decreases, while the other two terms of the approximation will increase. This relation reflects the trade-off between the signal and noise as the number of selected variables m increases. Moreover, the approximation shows that the number of selected variables m should be less than the sample size of the class 1 (n_1), otherwise the probability of misclassification will be more than one. Note that class 1 is the true class of the sample under classification.

Note that what inside the probability in the first term of the approximation in Theorem 4.2.1 is of order

$$\begin{aligned}\sqrt{\frac{n_1 n_2}{n}}\, C_{1n}^{-1/2}\left[\log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right) - a_n\right] &= \sqrt{\frac{n_1 n_2}{n}}\, C_{1n}^{-1/2}\log\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right) - \sqrt{\frac{n_1 n_2}{n}}\, C_{1n}^{-1/2} a_n \\ &= O_p\left(\frac{1}{\sqrt{m}}\log\left(\frac{n_2}{n_1}\right) - m^{-1/2}\, m\right) \\ &= O_p\left(\frac{1}{\sqrt{m}}\log\left(\frac{n_2}{n_1}\right) - \sqrt{m}\right) \qquad (4.2.31)\end{aligned}$$

Considering the order of (4.2.31), we can quantify the probability of misclassification for different situations as follows

Case 1: Assume $m \to \infty$.

- With $m \to \infty$, and $\frac{1}{\sqrt{m}}\log\left(\frac{n_2}{n_1}\right) = o(\sqrt{m})$, then the term $\left(\frac{1}{\sqrt{m}}\log\left(\frac{n_2}{n_1}\right) - \sqrt{m}\right) \to -\infty$ and hence the probability of misclassification goes to zero.
- With $m \to \infty$, $m = o\left(\log\left(\frac{n_2}{n_1}\right)\right)$, and
  - $n_2/n_1 \to \infty$. That is, $\sqrt{m} = o\left(\frac{1}{\sqrt{m}}\log\left(\frac{n_2}{n_1}\right)\right)$ and $n_2/n_1 \to \infty$, then the probability of misclassification goes to one.
  - $n_2/n_1 \to 0$, then the probability of misclassification goes to zero.
- With $m \to \infty$, and $\log(n_2/n_1)$ converges to some constant $\log(a)$, then the probability of misclassification goes to zero.

Case 2: Assume m stays fixed.

- With fixed m and $\frac{n_2}{n_1} = 1 + o(1)$, then $\frac{1}{\sqrt{m}} \log\left(\frac{n_2}{n_1}\right) \to 0$ and the term $\frac{1}{\sqrt{m}} \log\left(\frac{n_2}{n_1}\right) - \sqrt{m} \to -\sqrt{m}$. Therefore, the probability of misclassification goes to $\Phi(-\sqrt{m})$.
- With fixed m and $\log\left(\frac{n_2}{n_1}\right) \to \infty$, then $\Phi\left(\frac{1}{\sqrt{m}} \log\left(\frac{n_2}{n_1}\right) - \sqrt{m}\right) \to 1$. That is, if the new sample x^* comes from class 1, but the number of training samples from class 1 is limited while the number of training samples (n_2) from class 2 $\to \infty$, then this new sample will be misclassified with probability 1.
- With fixed m and $\log\left(\frac{n_2}{n_1}\right)$ converges to some constant $\log(a)$, then the probability of misclassification is approximately $\Phi\left(\frac{1}{\sqrt{m}} \log(a) - \sqrt{m}\right)$.

Corollary 4.2.2. *Under the assumptions of Theorem 4.2.1, if $\sigma_i = \sigma, \forall i = 1, ..., p$, then the probability of misclassification for the two classes case with hard thresholding is*

$$E\left[\Phi\left(\sqrt{2}\,(a_n)^{-1/2}\left[\log\left(\frac{n_2}{n_1}\right) - a_n\right]\right)\right] + O\left(\frac{\sqrt{m}}{\sqrt{n}}\right) + O\left(\frac{m}{n_1}\right),$$

where $a_n = \frac{1}{8}\frac{n}{n_1 n_2}\sum_{i=1}^{m} T_{ni}^2$.

Note that the expectation in the probability of misclassification only depends on the distribution of $T_{ni}, i = 1, ..., p$, which is noncentral T-distribution with $n - 2$ degrees of freedom and $\frac{\mu_{1i} - \mu_{2i}}{\sigma_i \sqrt{\frac{n}{n_1 n_2}}}$ noncentrally parameter. Also, $T_{n1}, ..., T_{np}$ are independent.

Corollary 4.2.3. *Assuming equal number of samples for both classes (i.e. $n_1 = n_2 = n_0$). Under the assumptions of Theorem 4.2.1, if $\sigma_i = \sigma, \forall i = 1, ..., p$, then the probability of misclassification for the two classes case with hard thresholding is*

$$E\left[1 - \Phi\left(\sqrt{\frac{1}{2\,n_0}\sum_{i=1}^{m} T_{ni}^2}\right)\right] + O\left(\frac{\sqrt{m}}{\sqrt{n_0}}\right)$$

$$= E\left[1 - \Phi\left(\frac{1}{2}\sqrt{\sum_{i=1}^{m}\left(\frac{\overline{x}_i^{(1)} - \overline{x}_i^{(2)}}{s_i}\right)^2}\right)\right] + O\left(\frac{\sqrt{m}}{\sqrt{n_0}}\right).$$

Chapter 5

Feature Selection and Classification Based on Heteroscedastic Models

5.1 Introduction

Assume a given set of n training samples from K different classes. Let n_k represent the number of samples from class k and C_k be the set of indices for those samples. Each sample is a vector of expression values for p genes. Let x_{ij} represent the gene expression for gene i of sample j. The ith element of class k centroid is the average gene expression value, $\overline{x}_i^{(k)} = \sum_{j \in C_k} x_{ij}/n_k$, for gene i. The ith element of the overall centroid is the average gene expression values over all training samples in all classes, $\overline{x}_i = \sum_{j=1}^{n} x_{ij}/n$.

The PAM test statistic for comparing class k to the overall centroid is

$$d_{ik} = \frac{\overline{x}_i^{(k)} - \overline{x}_i}{m_k(s_i + s_0)}, \tag{5.1.1}$$

where $m_k = \sqrt{1/n_k - 1/n}$ and s_i is the pooled within-class standard deviation for gene i. The s_0 is a constant to guard against large test statistics values caused by the possibility of small gene expression values. In the PAM algorithm, s_0 is set to be the median of the s_i values. After thresholding the test statistics values, the resulting values are used to compute the shrunken centroids, $\overline{x}_i^{\backprime(k)}$. These shrunken centroids will then be used for classifying any new sample, say $x^* = (x_1^*, x_2^*, \ldots, x_p^*)$, by comparing the discriminant scores for all classes.

The discriminant function for class k is

$$\delta_k(x^*) = \sum_{i=1}^{p} \frac{(x_i^* - \overline{x}_i^{\setminus(k)})^2}{(s_i + s_0)^2} - 2\log \pi_k, \tag{5.1.2}$$

where π_k is the class prior probability, which can be estimated from the training samples by n_k/n. The sample x^* will be assigned to the class with the smallest discriminant score (i.e. $\underset{k}{argmin}\ \delta_k(x^*)$).

The PAM test statistic (5.1.1) and discriminant function (5.1.2) use the pooled within-class standard deviation

$$s_i = \sqrt{\frac{\sum_k \sum_{j \in C_k} (x_{ij} - \overline{x}_i^{(k)})^2}{n - K}}. \tag{5.1.3}$$

This pooled standard deviation assumes homogeneity over different classes for the same gene. However, the assumption of constant variance (over different classes) is often not reasonable in practice. This can be seen in the heatmap of the sample standard deviations for some genes over different classes. Figures 5.1, 5.2 and 5.3 are the heatmaps of the sample standard deviation for the DLBCL, Leukemia2, and Lung2 data sets, respectively. The heatmaps for the other real data sets listed in Table 3.2 can be found in Appendix A. It is very clear from these heatmaps that many genes have completely different standard deviations, as represented by different colors for different classes. For some data sets, the difference in the standard deviation among different classes for some genes are more than 10,000 such as in Leukemia2 dataset (see Figure 5.2).

In this chapter we present an improved version of the PAM algorithm for the heteroscedastic situation. Starting in next Section 5.2 we present our heteroscedastic case test statistic and discriminant function. Then we describe the thresholding of the heteroscedastic test statistic using the different thresholding methods discussed in this dissertation. In Section 5.3 we present the performance of our heteroscedastic algorithms using both simulation and real data analysis.

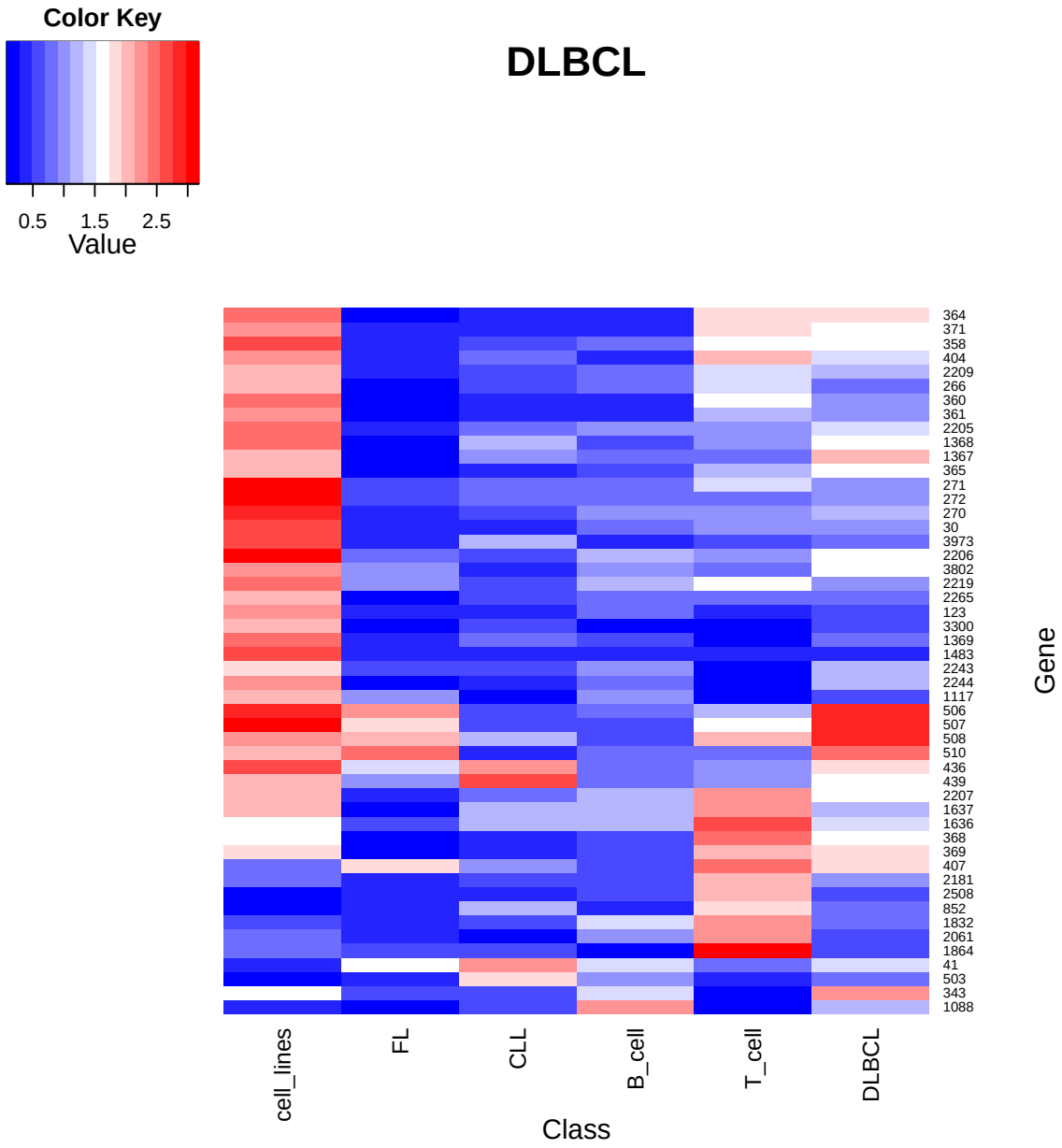


Figure 5.1: *Heatmap of the sample standard deviation for 50 genes from the DLBCL cancer dataset. These are for the top 50 genes that have the highest range of the standard deviations for different classes. For most genes, the cell_lines class has the highest standard deviation which is 3 times more than standard deviation in other classes for some cases.*

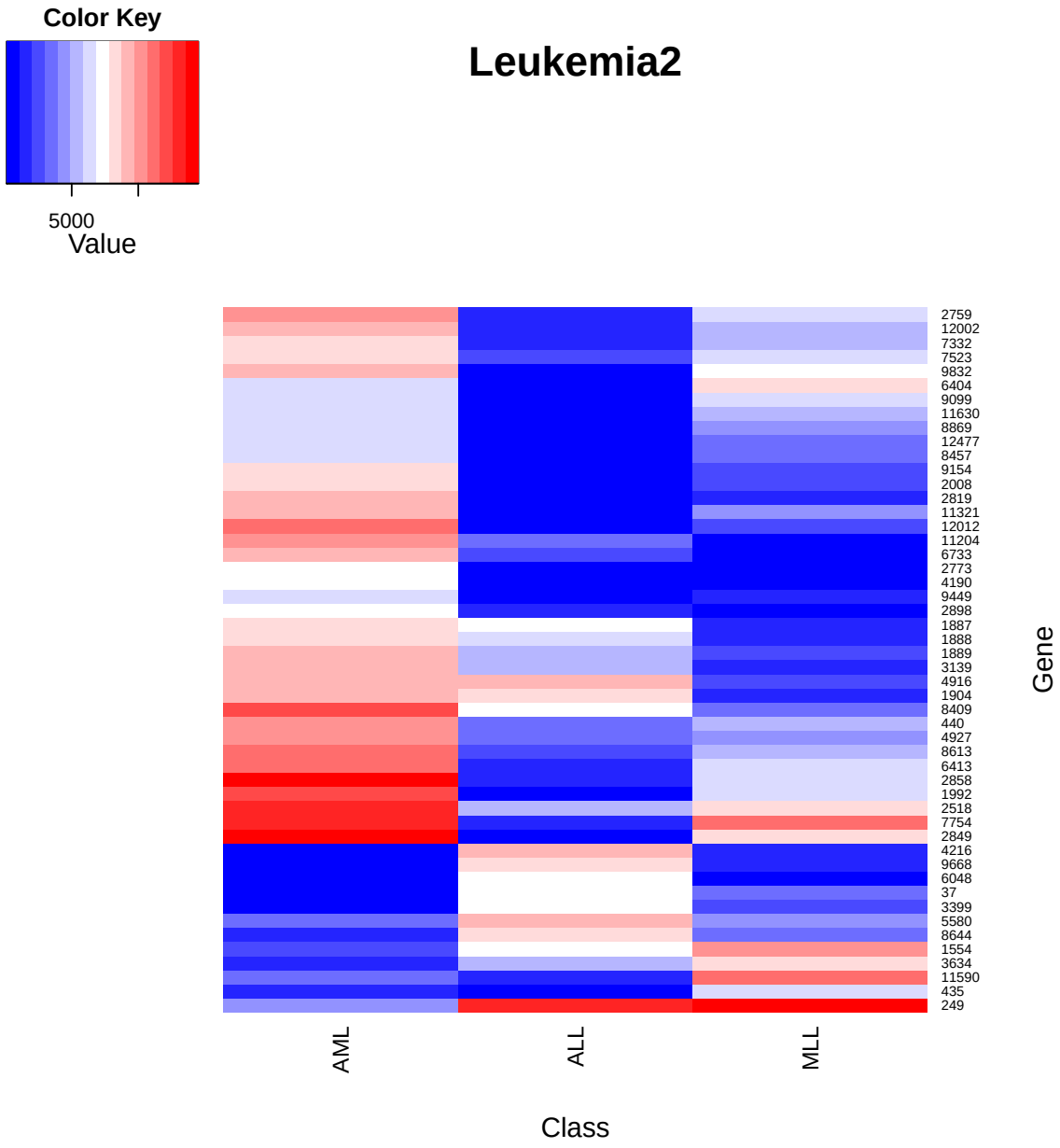


Figure 5.2: *Heatmap of the sample standard deviation for 50 genes from the Leukemia2 cancer dataset. These are for the top 50 genes that have the highest range of the standard deviations for different classes. For most genes, the AML class has the highest standard deviation. The range of the standard deviations among different classes is more than 10,000 for a lot of genes.*

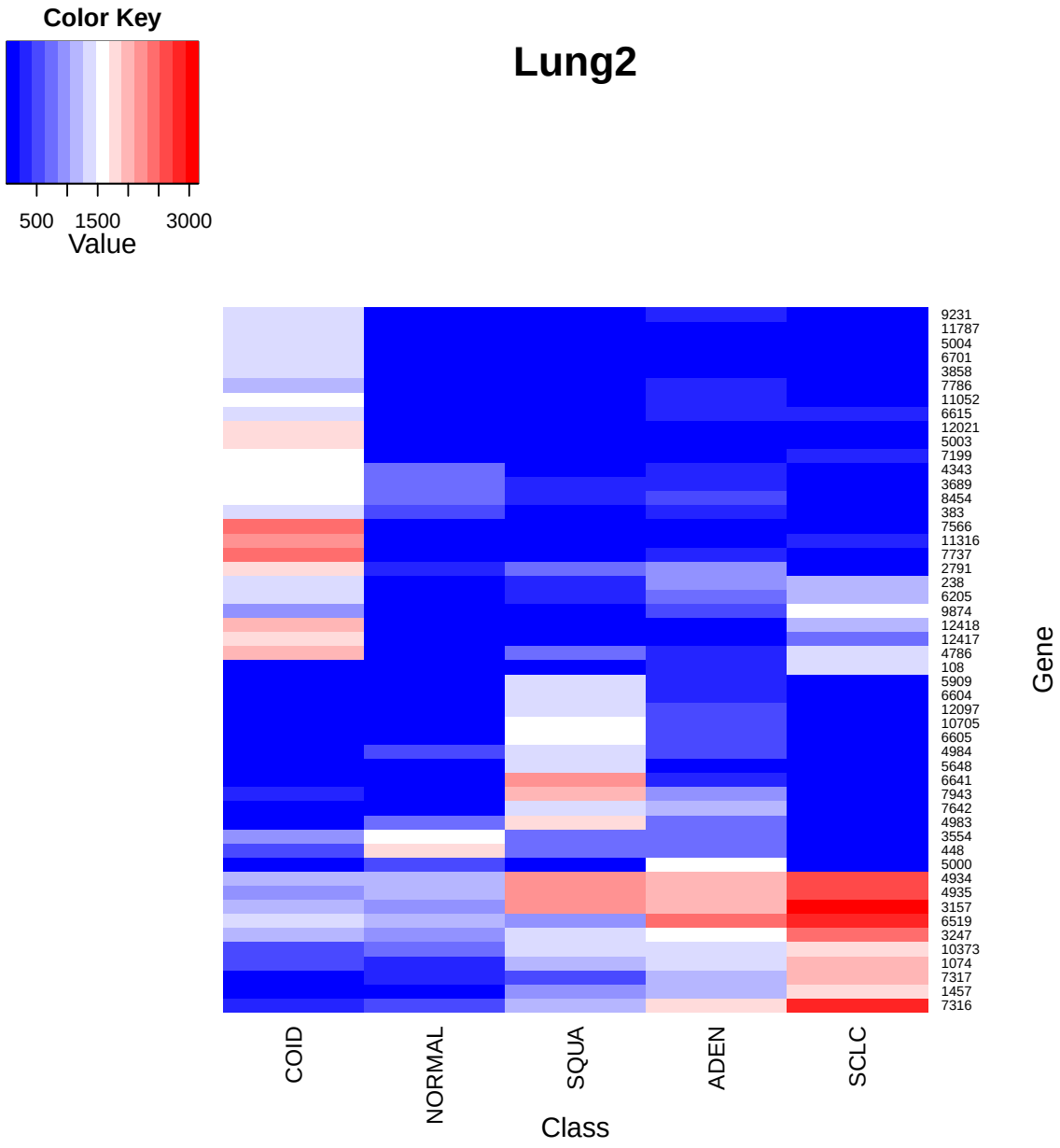


Figure 5.3: *Heatmap of the sample standard deviation for 50 genes from the Lung2 cancer dataset. These are for the top 50 genes that have the highest range of the standard deviations for different classes. The range of the standard deviations among different classes is up to 3,000 for some genes. For most genes, the NORMAL class has the lowest standard deviation.*

5.2 Method

5.2.1 Heteroscedastic case test statistic and discriminant function

In this section we will present our proposed test statistic and discriminant function for the heteroscedastic case. Under the assumption of heterogeneity the numerator of the PAM test statistic (5.1.1) has a different form of standard error. In heterogeneity case, the variance of the difference between the class k centroid and the overall centroid is

$$\begin{aligned} Var(\overline{x}_i^{(k)} - \overline{x}_i) &= Var(\overline{x}_i^{(k)}) + Var(\overline{x}_i) - 2Cov(\overline{x}_i^{(k)}, \overline{x}_i) \\ &= Var(\overline{x}_i^{(k)}) + Var\left(\frac{\sum_{m=1}^{K} n_m \overline{x}_i^{(m)}}{n}\right) - 2Cov\left(\overline{x}_i^{(k)}, \frac{\sum_{m=1}^{K} n_m \overline{x}_i^{(m)}}{n}\right) \\ &= Var(\overline{x}_i^{(k)}) + \frac{1}{n^2}\sum_{m=1}^{K} n_m^2 Var(\overline{x}_i^{(m)}) - 2Cov\left(\overline{x}_i^{(k)}, \frac{n_k}{n}\overline{x}_i^{(k)}\right), \qquad (5.2.1) \end{aligned}$$

where the last equality (5.2.1) assumed that the genes are independent as in the original PAM algorithm.

$$\begin{aligned} \text{Then,} \quad Var(\overline{x}_i^{(k)} - \overline{x}_i) &= Var(\overline{x}_i^{(k)}) + \frac{1}{n^2}\sum_{m=1}^{K} n_m^2 Var(\overline{x}_i^{(m)}) - 2\frac{n_k}{n} Var(\overline{x}_i^{(k)}) \\ &= \frac{n-2n_k}{n} Var(\overline{x}_i^{(k)}) + \frac{1}{n^2}\sum_{m=1}^{K} n_m^2 Var(\overline{x}_i^{(m)}) \\ &= \frac{n-2n_k}{n}\frac{Var(x_i^{(k)})}{n_k} + \frac{1}{n^2}\sum_{m=1}^{K} n_m^2 \frac{Var(x_i^{(m)})}{n_m} \\ &= \left(\frac{1}{n_k} - \frac{2}{n}\right) Var(x_{ij}^{(k)}) + \frac{1}{n^2}\sum_{m=1}^{K} n_m Var(x_{ij}^{(m)}) \end{aligned}$$

Therefore, the estimated standard error of the difference between the class k centroid and the overall centroid (i.e. the numerator of the PAM test statistic) under the heterogeneity assumption is

$$\sqrt{\left(\frac{1}{n_k}-\frac{2}{n}\right)s_{ik}^2+\frac{1}{n^2}\sum_{m=1}^{K}n_m s_{im}^2}, \tag{5.2.2}$$

where s_{ik} is the sample standard deviation for gene i under class k, which can be calculated from the training samples as follows:

$$s_{ik}=\sqrt{\frac{\sum_{j\in C_k}(x_{ij}-\overline{x}_i^{(k)})^2}{n-1}}. \tag{5.2.3}$$

Accordingly, the test statistics we propose for the heteroscedastic case is

$$d_{ik}=\frac{\overline{x}_i^{(k)}-\overline{x}_i}{\sqrt{(\frac{1}{n_k}-\frac{2}{n})(s_{ik}^2+s_{0k}^2)+\frac{1}{n^2}\sum_{m=1}^{K}n_m(s_{im}^2+s_{0m}^2)}}, \tag{5.2.4}$$

where s_{0k} is a constant, for class k, to guard against large test statistics values caused by the possibility of small gene expression values. We set s_{0k} to be the median of the genes' standard deviation values for class k (i.e. $s_{0k}=median(s_{1k},s_{2k},...,s_{pk})$).

For deriving the discriminant function under the assumption of heterogeneity, consider the Naive Bayes classifier. According to the Naive Bayes, for a new sample with a given set of variables $x^*=(x_1^*,x_2^*,...,x_p^*)$, the predicted class label (i.e. $\hat{y}$) is the one that achieves the highest posterior probability. This posterior probability can be written as

$$\begin{aligned} P(y|x_1^*,x_2^*,...,x_p^*) &= \frac{P(y)P(x_1^*,x_2^*,...,x_p^*|y)}{P(x_1^*,x_2^*,...,x_p^*)} \\ &= \frac{P(y)\prod_{i=1}^{p}P(x_i^*|y)}{P(x_1^*,x_2^*,...,x_p^*)} \quad \text{(using the naive independence assumption)} \end{aligned} \tag{5.2.5}$$

Since the denominator is constant given the sample x^*, then the comparison of the posterior probability will depend only on the numerator

$$P(y|x_1^*,x_2^*,...,x_p^*)\propto P(y)\prod_{i=1}^{p}P(x_i^*|y).$$

Hence, the predicted class label is

$$\hat{y} = \underset{y \in \{1,2,...,K\}}{argmax}\ P(y) \prod_{i=1}^{p} P(x_i^*|y).$$

Therefore, the posterior for class k (i.e. $y = k$) for the Gaussian case as assumed by PAM is

$$\begin{aligned} P(y = k|x_1^*, x_2^*, ..., x_p^*) &\propto P(y = k) \prod_{i=1}^{p} P(x_i^*|y = k) \\ &\propto P(y = k) \prod_{i=1}^{p} \frac{1}{\sigma_{ik}\sqrt{2\pi}} \exp\left\{\frac{-(x_i^* - \mu_{ik})^2}{2\sigma_{ik}^2}\right\}. \end{aligned}$$

Taking the logarithm for both sides,

$$\begin{aligned} \log\{P(y = k|x_1^*, x_2^*, ..., x_p^*)\} &\propto \log\{P(y = k)\} - \sum_{i=1}^{p} \log\left\{\sigma_{ik}\sqrt{2\pi}\right\} - \sum_{i=1}^{p}\left\{\frac{(x_i^* - \mu_{ik})^2}{2\sigma_{ik}^2}\right\} \\ &\propto 2\log\{P(y = k)\} - 2\sum_{i=1}^{p} \log\{\sigma_{ik}\} - \sum_{i=1}^{p}\left\{\frac{(x_i^* - \mu_{ik})^2}{\sigma_{ik}^2}\right\}. \end{aligned}$$

In application, the prior probabilities $P(y = k)$ is estimated by the relative frequency $\hat{\pi}_k = n_k/n$ of class y in the training set. The parameters σ_{ik} and μ_{ik} remain to be estimated from the training set using the maximum likelihood. Specifically, σ_{ik} is estimated by the sample standard deviation s_{ik} (5.2.3) and μ_{ik} is estimated by the shrunken centroid $\overline{x}_i^{\prime(k)}$ that is defined in terms of the thresholded test statistic $d_{ik}^{\prime}$. The thresholding could be done using one of the three thresholding methods (soft, hard, or order) discussed earlier in this dissertation. That is the *ith* element of the shrunken centroid can be defined as

$$\overline{x}_i^{\prime(k)} = \overline{x}_i + d_{ik}^{\prime(M)} \sqrt{\left(\frac{1}{n_k} - \frac{2}{n}\right)(s_{ik}^2 + s_{0k}^2) + \frac{1}{n^2}\sum_{m=1}^{K} n_m(s_{im}^2 + s_{0m}^2)}, \tag{5.2.6}$$

where $d_{ik}^{\prime(M)}$ is the thresholded test statistic with method M= soft, hard, or order.

Therefore, the class label that maximizes the posterior (5.2.5) is the class with the minimum discriminant score. The discriminant score for class k is

$$\delta_k(x^*) = \sum_{i=1}^{p} \frac{(x_i^* - \overline{x}_i^{\setminus(k)})^2}{(s_{ik}^2 + s_{0k}^2)} + 2\sum_{i=1}^{p} \log\sqrt{s_{ik}^2 + s_{0k}^2} - 2\log\hat{\pi}_k, \tag{5.2.7}$$

where s_{0k} is given in (5.2.4) (i.e. $s_{0k} = median(s_{1k}, s_{2k}, ..., s_{pk})$) and $\overline{x}_i^{\setminus(k)}$ is given in (5.2.6).

5.2.2 Thresholding the test statistics in the heteroscedastic case

Previously in Chapter 3 we discussed three algorithms: STh for soft thresholding (the original PAM), HTh for the hard thresholding, and OTh with order thresholding. Those three algorithms use the original PAM test statistics (5.1.1) and discriminant function (5.1.2) but with different thresholding methods. As we mentioned at the beginning of this chapter, the original PAM test statistics and discriminant function use the pooled standard deviation that assumes homogeneity over different classes for all genes. In this section, we briefly describe how the heteroscedastic test statistics (5.2.4) and discriminant function (5.2.7) will be used to produce improved versions of these algorithms.

Starting with the training data that has a set of n training samples, each with p genes. This will give a $n \times p$ matrix with each entry x_{ij} represents the gene expression for gene i of the training sample j, where $i = 1, ..., p$ and $j = 1, ..., n$. Denote the class labels of the response variable as $1, 2, ..., K$ if the n training samples are from K different classes. Let n_k denote the number of samples from class k and C_k be the set of indices for those samples.

First, the three algorithms start by computing each class centroid $\overline{x}_i^{(k)} = \sum_{j \in C_k} x_{ij}/n_k$ and the overall centroid $\overline{x}_i = \sum_{j=1}^{n} x_{ij}/n$ to find the heteroscedastic test statistic for each gene i in class k using

$$d_{ik} = \frac{\overline{x}_i^{(k)} - \overline{x}_i}{\sqrt{(\frac{1}{n_k} - \frac{2}{n})(s_{ik}^2 + s_{0k}^2) + \frac{1}{n^2}\sum_{m=1}^{K} n_m(s_{im}^2 + s_{0m}^2)}},$$

where $s_{ik} = \sqrt{\sum_{j \in C_k} (x_{ij} - \overline{x}_i^{(k)})^2/(n-1)}$ and s_{0k} is the median of the genes' standard deviation values for class k (i.e. $s_{0k} = median(s_{1k}, s_{2k}, ..., s_{pk})$).

Then in case of using the soft thresholding, all the test statistic values will be thresholded using soft thresholding

$$d_{ik}^{\prime(S)} = sgn(d_{ik})(|d_{ik}| - \Delta_S)_+, \tag{5.2.8}$$

where + means positive part (i.e. $b_+ = b\, I\{b > 0\}$). The soft thresholding parameter Δ_S is chosen to be the thresholding value that minimizes the misclassification error in a 10-fold cross-validation of the training samples. The ith element of the shrunken centroid for class k is written in terms of the thresholded test statistic $d_{ik}^{\prime(S)}$ as

$$\overline{x}_i^{\prime(k)} = \overline{x}_i + d_{ik}^{\prime(S)} \sqrt{\left(\frac{1}{n_k} - \frac{2}{n}\right)(s_{ik}^2 + s_{0k}^2) + \frac{1}{n^2}\sum_{m=1}^{K} n_m (s_{im}^2 + s_{0m}^2)}.$$

The shrunken centroids for all K classes will be used to classify any new sample $x^* = (x_1^*, x_2^*, \ldots, x_p^*)$. This is done by computing the discriminant score for each class using

$$\delta_k(x^*) = \sum_{i=1}^{p} \frac{(x_i^* - \overline{x}_i^{\prime(k)})^2}{(s_{ik}^2 + s_{0k}^2)} + 2\sum_{i=1}^{p} \log\sqrt{s_{ik}^2 + s_{0k}^2} - 2\log\hat{\pi}_k, \tag{5.2.9}$$

where $\hat{\pi}_k = n_k/n$ is the estimated probability for class k using the training-set.

Then the decision is to classify the new sample x^* to class $c = \underset{1\le k\le K}{argmin}\ \delta_k(x^*)$. This algorithm using soft thresholding will be denoted by STh3 in further discussion.

In case of using the hard thresholding, we replace the soft thresholding in (5.2.8) by the hard thresholding

$$d_{ik}^{\prime(H)} = d_{ik} I\{|d_{ik}| > \Delta_H\}, \tag{5.2.10}$$

where Δ_H is the hard thresholding parameter. The optimal Δ_H can be determined by using a 10-fold cross-validation over the training-set. It is selected to be the one that provides the amount of shrinkage that minimizes the cross-validation misclassification error. This algorithm, which uses hard thresholding, will be denoted by HTh3.

In case of using the order thresholding, we replace the soft thresholding in (5.2.8) by the order thresholding

$$d_{ik}'^{(O)} = \begin{cases} d_{ik} & if \quad rank(|d_{ik}|) > n - \Delta_O \\ 0 & otherwise \end{cases} \tag{5.2.11}$$

where Δ_O is the order thresholding parameter, which can be determined by using a 10-fold cross-validation over the training-set as well. This algorithm, which uses order thresholding, will be denoted by OTh3.

5.3 Numerical comparisons

In this section we present simulation and real data analysis to show the performance of our proposed algorithms (STh3, HTh3, and OTh3) that use the heteroscedastic test statistics (5.2.4) and discriminant function (5.2.7) compared to those algorithms (STh, HTh, and OTh) that use the homoscedastic PAM test statistics (5.1.1) and discriminant function (5.1.2).

The R software, version 3.0.2, was used for programming all algorithms to be compared in this section. Previously in Chapter 3, we wrote our own codes for the HTh and OTh algorithms to calculate the class centroid and perform cross-validation using the training data, and to predict the class labels for the test samples. For a completely fair comparison, in the studies of this section we code all the algorithms by modifying the functions from the **pamr** package, which was developed by the authors of Tibshirani et al. (2002). The 3 different thresholding methods and the two versions of the test statistics and discriminant functions give 6 algorithms. The functions that we modified from the **pamr** package are: pamr.train, pamr.cv, pamr.predict, nsc, nsccv, and diag.disc. The soft.shrink function was replaced by the new hard.shrink function to perform hard thresholding or by the new order.shrink function to perform order thresholding. The refining process described at the end of Section 3.1.1 was also implemented in these algorithms. Specifically, this process refine the neighborhood of the thresholding value with the smallest cross-validation error to reach a better estimate of the optimal thresholding parameter. In all algorithms the number of

folds for the cross-validation with the training data is set to be 10 unless the dataset under study has some classes with sample size less than 10. In the later case, the fold is set to be the smallest class size.

All data sets, either generated or real data, that will be used in this section are divided into training and test samples. Classifiers are trained with the training samples and then prediction of the class label for the test samples are conducted. For the real data in our study, we will adopt the same partition of training and test samples that is already divided by the authors of Tan et al. (2005). In the multi-class classification problems, the proportion of correctly classified samples or proportion of misclassified samples is typically used in the literature as the comparison criterion to compare the performance of different classifiers. In our study we will mainly use the proportion of misclassified test samples (test error) in our comparison. It is defined as the number of misclassified test samples divided by the total number of test samples. We will also compare the number of selected variables (genes) used in each method when the misclassification rates are similar for different algorithms. It is widely accepted that the better method is the one that uses less genes to achieve the same accuracy as other methods using more genes.

The random partition of the training data in cross-validation could lead to different estimated thresholding parameter and hence possibly a different test error. In all our studies we repeat the classification process 100 times for each dataset and report the average percentage of misclassified samples and the number of informative genes from the 100 runs.

5.3.1 Simulation study

For the simulation study we generated different high dimensional data sets with clear heteroscedasticity among different classes. Then we discuss the performance of the three algorithms for the heteroscedastic case STh3, HTh3, and OTh3 compared to their counter parts STh, HTh, and OTh applied to these data sets.

Example 1: Two classes The first simulation setup is for two classes (binary

classification). For the first class we generated 20 training samples and 50 test samples and for the second class 30 training samples and 50 test samples. Each sample has 10,000 independent variables. For the first class, all 10,000 variables were generated from a standard normal distribution $N(0,1)$. For the second class, all the variables were generated from normal distribution with standard deviation 3. The means for the first 20 variables in the second class are $\frac{i}{10}$ for $i = 1, 2, ..., 20$ and the means for the rest 9,980 variables are zeros. For easier discussion, from now on we will refer to these 20 variables as the double-signal variables since the two classes differ for both mean and variance for these variables. In mathematical notation, the generated data for class k, variable i, and sample j is

$$x_{ij}^{(k)} \sim \begin{cases} N(0,1) & for \quad k = 1 \\ \\ N(\frac{i}{10}, 3^2) & for \quad k = 2, i = 1, 2, ..., 20 \\ N(0, 3^2) & for \quad k = 2, i = 21, 22, ..., 10,000. \end{cases} \tag{5.3.1}$$

Even though the first 20 variables have different means for the two classes, the signal to noise ration $(\frac{\mu}{\sigma})$ is between 1:30 to 2:3, which is very low. So this dataset is mainly used to evaluate performance at low signal to noise ratio case.

Additionally, we generated another dataset with a stronger difference in the mean signals by increasing the mean of the first 20 variables in class two. Specifically, the data for the first 20 variables of the second class were generated from $N(\frac{i}{4}, 3^2)$ for $i = 1, 2, ..., 20$. The mean $\frac{i}{4}$ provides 2.5 times stronger signal than the mean $\frac{i}{10}$ in the previous case. The other variables were still generated from $N(0, 3^2)$. In summary, the generated data in terms of the class k, variable i, and sample j is

$$x_{ij}^{(k)} \sim \begin{cases} N(0,1) & for \quad k = 1, \\ \\ N(\frac{i}{4}, 3^2) & for \quad k = 2, i = 1, 2, ..., 20 \\ N(0, 3^2) & for \quad k = 2, i = 21, 22, ..., 10,000. \end{cases} \tag{5.3.2}$$

The pdf of these variables are given in Figures 5.4 and 5.5, which correspond to the distributions in (5.3.1) and (5.3.2), respectively. For each variable there is a big overlap of possible values between the two classes. This makes the classification difficult.

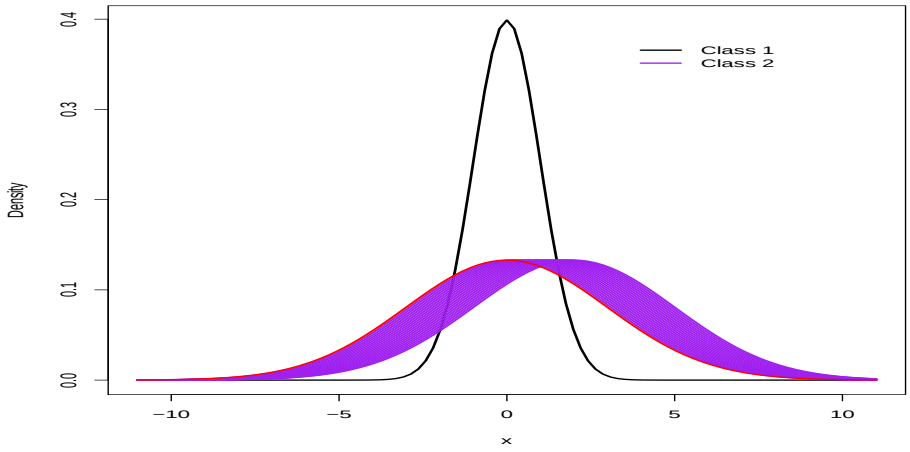


Figure 5.4: *The distributions of the different variables in each of the two classes in (5.3.1). The red curve corresponds to the pdf of the variables 21-10,000 in class 2*

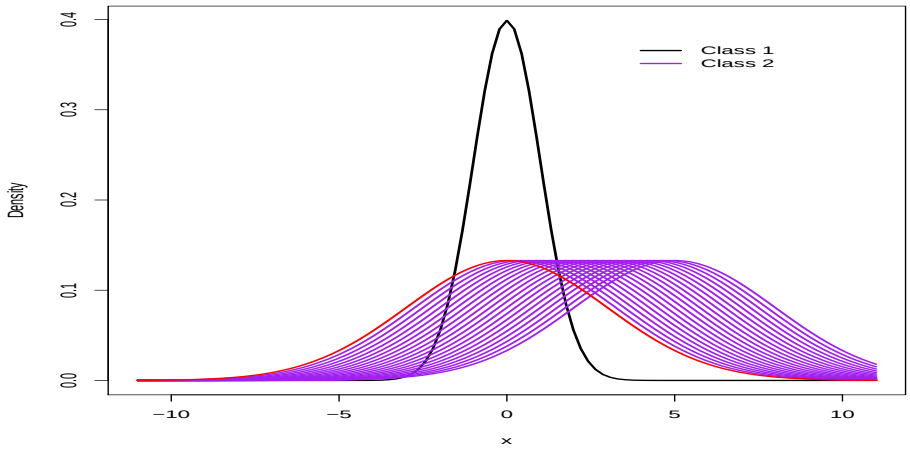


Figure 5.5: *The distributions of the different variables in each of the two classes in (5.3.2). The red curve corresponds to the pdf of the variables 21-10,000 in class 2*

The result for these two simulation settings is listed in Table 5.1. It can be seen that there is a clear advantage of using the heteroscedastic versions of the algorithms with any of the three thersholding methods. In the first dataset (data from (5.3.1)), all three heteroscedastic algorithms (STh3, HTh3, and OTh3) perform better than their counter parts (STh, HTh, and OTh) in terms of the test errors as there is a large reduction in the test error when using the heteroscedastic algorithms. Specifically, the test errors changed from 48.42, 38.52, and 34.89 for STh, HTh, and OTh to 1.71, 1.5, and 1.95 for STh3, HTh3, and OTh3 respectively. In the second dataset (data from (5.3.2)), which has a larger mean difference for the first

20 variables, the heteroscedastic algorithms still perform better than their counter parts in terms of the test errors. The test errors are 1.78, 2.22, and 1.94 for STh, HTh, and OTh while the errors are 0.71, 1.07, and 0.92 for STh3, HTh3, and OTh3 respectively which are about half of the errors for the homoscedastic algorithms. Increasing the signals by using dataset from (5.3.2) gives a drastic boost to the performance of the homoscedastic version algorithms. The test errors for STh, HTh, and OTh is much smaller than their test errors in the first dataset (5.3.1). They changed from 48.42, 38.52, and 34.89 in the first dataset to 1.78, 2.22, and 1.94 in the second dataset. On the other hand, from this simulation with the two data sets we see that the heteroscedastic version algorithms are less prone to classification error due to weak mean signals.

Table 5.1: *Example 1 simulation results: Percent of mean misclassification error for test samples (test error), the average number of selected variables (selected variables), and the average of how many selected variables are from the first 20 double-signal variables. All these results are based on 100 runs for each algorithm. STh, HTh, and OTh are the homoscedastic version algorithms. STh3, HTh3, and OTh3 are the heteroscedastic version algorithms.*

	Data from (5.3.1)				Data from (5.3.2)			
			out of the 20 double-signals				out of the 20 double-signals	
algorithm	test error	selected variables	yes	no	test error	selected variables	yes	no
STh	48.42	22.95	3.07	19.88	1.78	7.21	7.05	0.16
STh3	1.71	21.12	3.27	17.85	0.71	13.76	11.38	2.38
HTh	38.52	5.27	1.75	3.52	2.22	5.6	5.59	0.01
HTh3	1.5	27.57	3.65	23.92	1.07	5.01	4.98	0.03
OTh	34.89	7.11	2.06	5.05	1.94	6.45	6.22	0.23
OTh3	1.95	26.25	3.62	22.63	0.92	4.74	4.68	0.06

In terms of the selected variables, the stronger signal in the first 20 variables from dataset (5.3.2) makes it less necessary to use variables that only differ in variance for the two classes. For example, in the first dataset the average number of selected variables for the SHh3 is 21.12, among which 3.27 of them are from the 20 double-signals and 17.85 are from the

rest of the variables. In the second dataset an average 13.76 variables were selected, among which 11.38 are from the 20 mean signals and 2.38 are from the rest of the variables. For the other algorithms, the average number of selected noisy variables that has no mean signal ranges from 3.5 to 22.63 for data from (5.3.1). But for data from (5.3.2), an average of less than 1 such variables were selected.

Example 2: Three classes The second simulation setup is for three classes. For the first class we generated 20 training samples and 50 test samples. For each of the other two classes we generated 30 training samples and 50 test samples. Each sample has 10,000 independent variables. The distributions for the first two classes were generated similar to those in the first dataset of Example 1. That is, all 10,000 variables in the first class were generated from a standard normal distribution $N(0,1)$. For the second class, all the variables were generated from normal distribution with standard deviation 3. The means for the first 20 variables are $\frac{i}{10}$, $i = 1, 2, ..., 20$ and the means for the rest 9,980 variables are zeros. For the third class, all the variables were generated from normal distribution with standard deviation 5. The means for the first 20 variables are $\frac{21-i}{10}$, $i = 1, 2, ..., 20$ and the means for the rest 9,980 variables are zeros. In summary, the generated data in terms of the class k, variable i, and sample j is

$$x_{ij}^{(k)} \sim \begin{cases} N(0,1) & for \quad k = 1 \\ \\ N(\frac{i}{10}, 3^2) & for \quad k = 2, i = 1, 2, ..., 20 \\ N(0, 3^2) & for \quad k = 2, i = 21, 22, ..., 10,000. \\ \\ N(\frac{21-i}{10}, 5^2) & for \quad k = 3, i = 1, 2, ..., 20 \\ N(0, 5^2) & for \quad k = 3, i = 21, 22, ..., 10,000. \end{cases} \tag{5.3.3}$$

This is a case with low signal to noise ratio with multiple classes. The addition of third class with even bigger variation contributes further challenge to the classification. The results for this simulation are listed in Table 5.2. Similar to Example 1 simulation, all three heteroscedastic version algorithms (SHT3, HTh3, and OTh3) perform better than their counter parts (STh, HTh, and OTh) in terms of the test errors. The test errors and number

of selected variables in this example are larger than those in the first dataset of Example 1 (data from (5.3.1)) because of the additional class in this example. Still there is a large reduction in the test error when using the heteroscedastic version algorithms. The test errors for STh, HTh, and OTh were 63.33, 63.55, and 62.64 respectively. On the other hand the test errors for STh3, HTh3, and OTh3 were 20.95, 21.97, and 22.46 respectively. Moreover, we notice from the results of this simulation that a much large number of variables were selected by the original PAM algorithm STh to reach almost the same test error in both HTh and OTh, while it was reduced to when using the heteroscedastic algorithms. Also, the results show that the number of selected variables needed by the hard thresholding algorithms to make the classification were the smallest compared to the soft and order thresholding algorithms. Among the heteroscedastic version algorithms the STh3 has the smallest test error.

Table 5.2: *Example 2 simulation result: Percent of mean misclassification error for test samples (test error) and average number of selected variables (selected variables) based on 100 runs for each algorithm. The value in parenthesis is the median absolute deviation. STh, HTh, and OTh are the homoscedastic version algorithms. STh3, HTh3, and OTh3 are the heteroscedastic version algorithms.*

	Data from (5.3.3)		Data from (5.3.4)	
algorithm	test error	selected variables	test error	selected variables
STh	63.33(2)	226(7)	62.82(2)	84(6.5)
STh3	20.95(1.33)	40(10)	19.67(1.33)	49(19.5)
HTh	63.55(2.67)	10(1)	62.89(2.33)	13(1)
HTh3	21.97(2)	39(12.5)	20.91(2.33)	46(14)
OTh	62.64(2)	35(3.5)	62.86(2)	32(4)
OTh3	22.46(2.33)	68(15)	21.39(2.67)	91(29.5)

For different situation other than using the same variance for all variables in the same class, we considered using different variances for different variables. Specifically, we randomly generated 10,000 values from each of the distributions $N(1, .3^2)$, $N(3, .3^2)$, and

$N(5, .3^2)$ for class 1, 2, and 3 respectively. Then we used the absolute values of these generated values as the standard deviation for the variables in each class, respectively. This way we have different variances for different variables, but at the same time we insured the presence of heterogeneity among different classes. In summary, the generated data in terms of the class k, variable i, and sample j is

$$x_{ij}^{(k)} \sim \begin{cases} N(0, \sigma_1^2) & for \quad k = 1, \sigma_1 \sim N(1, 0.3^2) \\ \\ N(\frac{i}{10}, \sigma_2^2) & for \quad k = 2, \sigma_2 \sim N(3, 0.3^2), i = 1, 2, ..., 20 \\ N(0, \sigma_2^2) & for \quad k = 2, \sigma_2 \sim N(3, 0.3^2), i = 21, 22, ..., 10,000. \\ \\ N(\frac{21-i}{10}, \sigma_3^2) & for \quad k = 3, \sigma_3 \sim N(5, 0.3^2), i = 1, 2, ..., 20 \\ N(0, \sigma_3^2) & for \quad k = 3, \sigma_3 \sim N(5, 0.3^2), i = 21, 22, ..., 10,000. \end{cases} \tag{5.3.4}$$

The plots of the pdfs correspond to the distributions in (5.3.3) and (5.3.4) are in Figures 5.6 and 5.7, respectively.

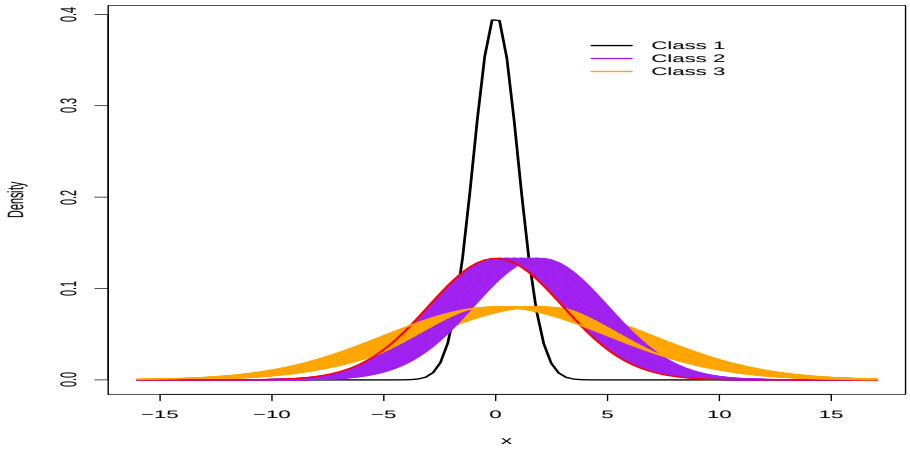


Figure 5.6: *The distributions of the different variables in each of the three classes in (5.3.3). The red curve corresponds to the pdf of the variables 21-10,000 in class 2*

The results of the analysis of this dataset are also given in Table 5.2. The test errors for STh, HTh, and OTh were 62.82, 62.89, and 62.86, while the test errors for STh3, HTh3, and OTh3 were 19.67, 20.91, and 21.39, respectively. The test errors for the heteroscedastic algorithms in this case are slightly smaller than those in the previous simpler case, which has common variance for all variables in the same class. Specifically, for data from (5.3.3)

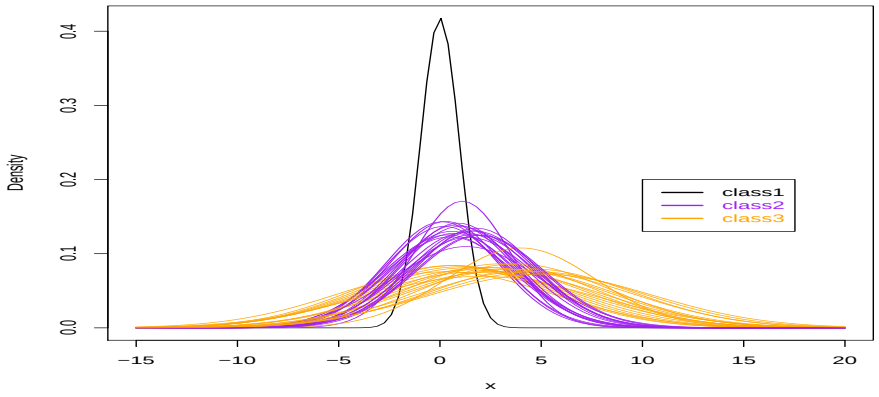


Figure 5.7: *The distributions of the different variables in each of the three classes in (5.3.4). The red curve corresponds to the pdf of the variables 21-10,000 in class 2*

the test errors for STh3, HTh3, and OTh3 were 20.95, 21.97, and 22.46 respectively, while for data from (5.3.4) the test errors were 19.67, 20.91, and 21.39 respectively. On the other hand, the number of selected variables used by the heteroscedastic algorithms are slightly larger than those for dataset (5.3.3). Similar to the common variance case (data from (5.3.3)), the number of selected variables needed by the hard thresholding algorithms to make the classification were the smallest. Also, among the heteroscedastic algorithms the STh3 has the smallest test error.

Example 3: Multi-class For this multi-class simulation, we mainely mimic the sizes (i.e. number of classes, number of variables, and number of samples) of the ten multi-class human cancers gene expression data sets that we are investigating in this dissertation. These summary of the ten data sets are listed in Table 5.3. The number of classes in those data sets ranges from 3 to 14 and the number of genes ranges from 2308 to 16063. Each dataset contains two parts, training samples part (training-set), and test samples part (test-set). The number of samples in this study range from 38 to 215 for the training samples and 15 to 112 for the test samples.

The detailed classes and the number of training and test samples in each class are listed in Table 5.4. The simulated data only differ from the real data by the data values. Here they

Table 5.3: *Settings of real data for simulation settings.*

Dataset abbreviation	No of classes	No of genes	No of samples	
			Training	Testing
SRBCT	4	2308	63	20
Breast	5	9216	54	30
Cancers	11	12533	100	74
DLBCL	6	4026	58	30
GCM	14	16063	144	46
Leukemia1	3	7129	38	34
Leukemia2	3	12582	57	15
Leukemia3	7	12558	215	112
Lung1	3	7129	64	32
Lung2	5	12600	136	67

were generated under different setups to simulate a clear heteroscedasticity among different classes. To distinguish between the generated data and the real one, we will add asterisk (*) at the end of the name of the generated data.

Without loss of generality, assume the classes for each dataset are labeled 1 through K (i.e. $k \in \{1, 2, ..., K\}$), where K is the total number of classes.

Scenario 1: For class k, all the variables were generated from normal distribution with standard deviation $2k-1$. The means for the first 20 variables are $\frac{i}{10}k$, $i = 1, 2, ..., 20$ and the means for the rest of the variables are zeros. In summary, for a dataset with p variables

$$x_{ij}^{(k)} \sim \begin{cases} N\left(\frac{i}{10}k, [2k-1]^2\right) & for \quad i = 1, 2, ..., 20 \\ N\left(0, [2k-1]^2\right) & for \quad i = 21, 22, ..., p. \end{cases} \tag{5.3.5}$$

It should be noted that this data generation scheme produces really big noise for some of the classes if the number of classes is big. For example, with the GCM dataset setting, the standard deviation ranges from 1 to 27.

The results for this simulation are listed in Table 5.5. The advantage of using the heteroscedastic versions algorithms with any of the three thersholding methods can be seen

Table 5.4: *Number of training samples (tr) and test samples (te) in each class of the data sets used in this dissertation.*

Dataset	Number of samples in each class													
SRBCT	class	BL	EWS	NB	RMS									
	tr	8	23	12	20									
	te	3	6	6	5									
Breast	class	basal	cell_lines	ERBB2	lumina	normal								
	tr	3	7	3	12	5								
	te	7	12	6	20	9								
Cancers	class	BL	BR	CO	GA	KI	LI	LU_A	LU_S	OV	PA	PR		
	tr	8	12	11	11	10	6	9	8	9	6	10		
	te	0	14	12	1	1	1	5	6	18	0	16		
DLBCL	class	B_Cell	Cell_lines	CLL	DLBCL	FL	T_Cell							
	tr	7	4	7	30	6	4							
	te	3	2	4	16	3	2							
GCM	class	Bl	Br	CNS	Co	Le	Lu	Ly	Mel	Mes	Ov	Pa	Pr	Re	Ut
	tr	8	8	16	8	24	8	16	8	8	8	8	8	8	8
	te	3	3	4	3	6	3	6	2	3	3	3	2	3	2
Leukemia1	class	AML	B_Cell	T_Cell										
	tr	11	19	8										
	te	14	19	1										
Leukemia2	class	ALL	AML	MLL										
	tr	20	20	17										
	te	4	8	3										
Leukemia3	class	ALL	AML	BCR	E2A	HYP	MLL	Others						
	tr	28	52	9	18	42	14	52						
	te	15	27	6	9	22	6	27						
Lung1	class	ADEN	COID	NORMAL										
	tr	44	13	7										
	te	23	6	3										
Lung2	class	ADEN	COID	NORMAL	SCLC	SQUA								
	tr	93	13	12	4	14								
	te	46	7	5	2	7								

in these results. The three heteroscedastic version algorithms (STh3, HTh3, and OTh3) have smaller test errors than their counter parts homoscedastic version algorithms (STh, HTh, and OTh) for all data sets except for the Cancers* dataset, which has the second largest number of classes (11). For the homoscedastic version algorithms there are 6 test errors that are more than 50%, while only 4 or less test errors that are more than 50% for the heteroscedastic version algorithms. For comparing the thresholding methods, the STH3 has the smallest test error in 4 of this simulation data sets, while the HTh3 and OTh3 have the smallest test error in 3 data sets each. Moreover, the number of selected variables in the

HTh3 were smaller than those in STh3 and OTh3 except for the DLBCL* and Lung2 data sets. Considering the significant reduction in the test error when using the heteroscedastic algorithms we can see that the homoscedastic test statistic (PAM test statistic) used in the STh, HTh, and OTh algorithms was not able to correctly identify variables with mean signals and instead it selects more of the noisy variables. For example, with the SRBCT dataset the HTh only selected 3 variables to reach 70.52 test error, while the HTh3 was able to detect more of the informative variables (25) to reach 42.36 test error. On the other hand, the overfitting might be another problem of the homoscedastic version algorithms. For example, in the Leukemia2* dataset the STh reached 53.20 test error by selecting 408 variables, while the STh3 only needed 82 variables to reach 10.60 test error.

Table 5.5: *Scenario 1 simulation result: Percent of mean misclassification error for test samples (test error) and average number of selected variables (selec. var.) based on 100 runs for each algorithms. STh, HTh, and OTh are the homoscedastic version algorithms. STh3, HTh3, and OTh3 are the heteroscedastic version algorithms. In all data sets the variables were generated according to (5.3.5) with 20 mean signals.*

	STh		STh3		HTh		HTh3		OTh		OTh3	
	test	selec.	test	selec.	test	selec.	test	selec.	test	selec.	test	selec.
Dataset	error	var.	error	var.	error	var.	error	var.	error	var.	error	var.
SRBCT*	73.04	86	40.24	33	70.52	3	42.36	25	69.64	31	41	40
Breast*	60.17	46	60.60	119	60.53	3	51.83	49	61.27	13	53.97	191
Cancers*	80.89	5091	93.14	1266	80.49	2244	96.96	28	80.82	4348	99.38	1098
DLBCL*	46.67	62	38.73	804	46.67	3	29.77	979	46.50	11	28.57	1106
GCM*	87.07	5192	81.50	287	86.93	1820	73.61	24	86.87	3864	74.83	171
Leukemia1*	45.06	200	1.97	170	45.35	20	5.15	31	44.88	32	3.88	65
Leukemia2*	53.20	408	10.60	82	55.87	59	15.73	57	57.73	87	13.47	121
Leukemia3*	61.42	6	54.71	17	64.37	5	44.79	10	64.06	8	45.10	11
Lung1*	28.09	16	4.78	182	27.94	51	9.06	55	27.72	91	8.41	147
Lung2*	31.34	7	14.98	522	31.06	18	10.73	549	30.48	45	10.64	897

It is interesting to note that the three dataset settings with 3 classes (Leukemia1, Leukemia2, lung1) lead to smallest test errors for the heteroscedastic algorithms. Even

though the signal to noise ratio is low, the misclassification error for the heteroscedastic algorithms can be as low as 1.97%.

Scenario 2: For class k, all the variables were generated from normal distribution with standard deviation $2k-1$. For the means, we randomly select 20 variables (denote this set of variables by ϕ_k) and set the means for those variables to be 2, while the means for the rest of the variables are zeros. The 20 randomly select variables are different for each class (not overlapping). This way the contribution of the mean signals are the same in magnitude, but they are from 20 different variables in different classes. In the case of gene expression dataset, this set of selected variables can be thought of as the identifiers (markers) for each class. In summary, the generated data in terms of the class k, variable i, and sample j is

$$x_{ij}^{(k)} \sim \begin{cases} N\left(2, [2k-1]^2\right) & for \quad i \in \phi_k \\ N\left(0, [2k-1]^2\right) & for \quad otherwise, \end{cases} \tag{5.3.6}$$

where ϕ_k is a set of 20 randomly select variables that are different for each class (i.e. $\phi_i \bigcap \phi_j = \emptyset$ for $i \neq j$).

Table 5.6: *Scenario 2 simulation result: Percent of mean misclassification error for test samples (test error) and average number of selected variables (selec. var.) based on 100 runs for each algorithms. STh, HTh, and OTh are the homoscedastic version algorithms. STh3, HTh3, and OTh3 are the heteroscedastic version algorithms. In all data sets the variables were generated according to (5.3.6) with 20 mean signals for each class.*

		No of	STh		STh3		HTh		HTh3		OTh		OTh3	
	No of	mean	test	selec.	test	selec.	test	selec.	test	selec.	test	selec.	test	selec.
Dataset	classes	signals	error	var.	error	var.	error	var.	error	var.	error	var.	error	var.
SRBCT*	4	80	74	143	39	31	72.28	45	41.36	28	72.04	72	41.08	31
Breast*	5	100	60.27	53	55.67	33	60.83	4	50.77	50	61.40	10	51.03	135
Cancers*	11	220	81.45	4269	90.22	398	80.18	1934	92.64	52	80.43	4347	99.08	1384
DLBCL*	6	120	46.67	58	36.13	723	46.67	3	29.60	918	46.50	12	28.27	1100
GCM*	14	280	86.89	4653	76.17	27	86.93	1219	75.09	28	87.02	2791	76.72	286
Leukemia1*	3	60	48.09	534	1.71	149	46.44	32	6	37	45.88	44	4.32	68
Leukemia2*	3	60	60.20	432	10.67	75	61.60	59	16.80	49	61.20	98	13.73	150
Leukemia3*	7	140	69.57	44	48.31	12	70.36	2	47.20	13	69.28	12	46.87	15
Lung1*	3	60	28.12	30	5	177	28.03	56	9.66	90	27.91	117	8.97	154
Lung2*	5	100	31.34	5	15.54	636	31.30	26	10.85	621	30.87	29	10.69	989

The results for this simulation are listed in Table 5.6. Note that the number of mean signals in this scenario depends on the number of classes since we generated 20 mean signals for each class. The number of the mean signals are listed in the third column of the Table 5.6. The advantage of using the heteroscedastic version algorithms with any of the three thersholding methods can be seen in this scenario results as well. The three heteroscedastic version algorithms (STh3, HTh3, and OTh3) have smaller test errors than their counter parts homoscedastic version algorithms (STh, HTh, and OTh) for all data sets except for the Cancers* dataset, which has the second largest number of classes (11).

Scenario 3: For class k, all the variables were generated from normal distribution with mean zero and standard deviation $2k-1$. This way the means for all variables are the same, but the variances differ by class. Therefore, the class variance will be the only effect on the signals. That is

$$x_{ij}^{(k)} \sim N\left(0, [2k-1]^2\right), \text{ where } k = 1, 2, ..., K. \tag{5.3.7}$$

Table 5.7: *Scenario 3 simulation result: Percent of mean misclassification error for test samples (test error) and average number of selected variables (selec. var.) based on 100 runs for each algorithms. STh, HTh, and OTh are the homoscedastic version algorithms. STh3, HTh3, and OTh3 are the heteroscedastic version algorithms. In all data sets the variables were generated according to (5.3.7) without mean signals.*

		STh		STh3		HTh		HTh3		OTh		OTh3	
Dataset	No. of variables	test error	selec. var.	test error	selec. var.	test error	selec. var.	test error	selec. var.	test error	selec. var.	test error	selec. var.
SRBCT*	2308	74.32	28	41.48	21	72.64	2	43.84	24	71.48	4	42.36	43
Breast*	9216	60.27	50	55.13	31	60.60	3	49.73	49	61.27	12	52.43	156
Cancers*	12533	81.57	3981	90.12	327	80.19	2059	92.76	63	80.27	3770	98.73	1374
DLBCL*	4026	46.67	62	38.57	666	46.67	3	29.87	984	46.50	11	29.17	1098
GCM*	16063	86.91	4773	75.74	9	86.98	1303	74.76	31	86.87	2413	76.63	347
Leukemia1*	7129	44.94	81	1.65	150	44.65	8	4.71	33	44.82	20	4.35	68
Leukemia2*	12582	60.20	639	11.80	84	63.13	53	15.93	59	61.93	101	14.13	145
Leukemia3*	12558	69.65	41	46.41	9	70.29	3	47.30	13	69.47	13	47.90	17
Lung1*	7129	28.13	17	4.97	169	27.97	39	9.72	56	27.84	67	8.28	181
Lung2*	12600	31.34	5	15.81	552	31.28	17	11.01	664	30.90	22	10.66	1017

The results for this simulation are listed in Table 5.7. Even though there is no mean signals in this scenario, the advantage of using the heteroscedastic versions algorithms with any of the three thersholding methods is still obvious. The three heteroscedastic version of the algorithms (STh3, HTh3, and OTh3) have smaller test errors than their counter parts (STh, HTh, and OTh) for all data sets except for the Cancers* dataset, which has the second largest number of classes (11). Similar to the previous scenarios, for the homoscedastic version algorithms there are 6 test errors that are more than 50%, while only 3 or less test errors that are more than 50% for the heteroscedastic version algorithms. Two of those aforementioned large test errors for the heteroscedastic algorithms are for settings with the largest number of classes, Cancers* and GCM*. The number of classes for these data sets are 11 and 14, respectively.

Table 5.8: *Comparing Scenario 1 simulation result to Scenario 3 simulation result for the soft thresholding method algorithms STh and STh3.*

	STh						STh3					
	Data from (5.3.5)				Data from (5.3.7)		Data from (5.3.5)				Data from (5.3.7)	
			out of the 20						out of the 20			
	test	selec.	mean signals		test	selec.	test	selec.	mean signals		test	selec.
Dataset	error	var.	yes	no	error	var.	error	var.	yes	no	error	var.
SRBCT*	73.04	86	3	83	74.32	28	40.24	33	9	24	41.48	21
Breast*	60.17	46	0	45	60.27	50	60.60	119	7	112	55.13	31
Cancers*	80.89	5091	11	5080	81.57	3981	93.14	1266	13	1254	90.12	327
DLBCL*	46.67	62	0	62	46.67	62	38.73	804	12	791	38.57	666
GCM*	87.07	5192	9	5184	86.91	4773	81.50	287	11	276	75.74	9
Leukemia1*	45.06	200	1	198	44.94	81	1.97	170	5	165	1.65	150
Leukemia2*	53.20	408	5	403	60.20	639	10.60	82	5	77	11.80	84
Leukemia3*	61.42	6	3	3	69.65	41	54.71	17	10	7	46.41	9
Lung1*	28.09	16	0	16	28.13	17	4.78	182	4	178	4.97	169
Lung2*	31.34	7	1	6	31.34	5	14.98	522	9	513	15.81	552

To see how well the homoscedastic algorithms identify variables with difference in mean signals compared to the heteroscedastic algorithms, in Table 5.8 we list the soft thresholding algorithms simulation results for the data with 20 mean signals (Scenario 1) and the data without any mean signals (Scenario 3). We can see that the homoscedastic test statistic (PAM test statistic) used in the STh algorithm was less able to correctly identify variables

with mean signals and instead it selects more of the noisy variables. The median number of selected variables out of the 20 mean signals for the STh is only 2, while for the STh3 the number of selected variables out of the 20 mean signals is 9.

5.3.2 Real data analysis

For this data analysis section we will use the actual human cancers gene expression data sets listed in Table 5.3. The class labels and number of training and test samples in each class are listed in Table 5.4. Leukemia3 dataset was removed from this analysis because about 71% of its data values are zeros.

The result for this real data analysis is listed in Table 5.9. From this data analysis, the smallest median test error is 6.73 achieved by the OTh3 algorithm. The result shows that for the order thresholding algorithms, the test errors of the heteroscedastic version were less than the test errors of the homeostatic version in 7 out of the 9 data sets. The test errors for the OTh ranged between 4.45 and 54 with median test error of 11.84, while for the OTh3 ranged between 0.6 and 51.48 with median test error of 6.73. For the hard thresholding algorithms, the test errors of the heteroscedastic version were less than the test errors of the homeostatic version in 6 out of the 9 data sets. The median test error for HTh is 11.73 and for HTh3 is 6.8. But for the soft thresholding algorithms, the median test error for STh is 9.2 and for STh3 is 12.78. Only in 2 out of the 9 data sets the test errors of STh3 were less than the test errors of STh. Mainly, the SRBCT and Breast data sets show a smaller test errors when using the homeostatic version with any of the thresholding methods. However, for the Breast dataset the test errors for the hard and order thresholding algorithms were smaller than the test errors for the soft thresholding algorithms. In general, the test errors for the HTh3 and OTh3 were smaller than those for the STh3 for all data sets except for the GCM dataset. One of the noticeable reduction in test error when using the heteroscedastic version algorithms is with the Leukemia2 dataset. For Leukemia2 dataset, the test errors the homoscedastic version algorithms (STh, HTh, and OTh) were 15.13, 11.73, and 25.4 , while the test errors the heteroscedastic version algorithms (STh3, HTh3, and OTh3) were

2.87, 0.13, and 0.6 respectively. In terms of the number of selected genes, the HTh3 and OTh3 algorithms selected a smaller number of genes than the STh3 algorithm in 7 data sets.

Table 5.9: *Real Data Analysis: Percent of mean misclassification error for test samples (test error) and average number of selected genes (selec. genes) based on 100 runs for each algorithms. STh, HTh, and OTh are the homoscedastic version algorithms. STh3, HTh3, and OTh3 are the heteroscedastic version algorithms.*

	STh		STh3		HTh		HTh3		OTh		OTh3	
Dataset	test error	selec. genes	test error	selec. genes	test error	selec. genes	test error	selec. genes	test error	selec. genes	test error	selec. genes
SRBCT	5	110	23.75	1093	5	40	18.9	215	5	48	19.9	169
Breast	9.2	4312	23.03	4840	5	866	6.8	907	4.9	1233	6.73	786
Cancers	11.97	1413	12.78	11645	12.01	1431	12.32	9958	11.84	1824	11.05	7887
DLBCL	8.2	3649	6.2	3901	7.8	721	3.17	3108	7.37	829	1.87	2022
GCM	44.17	2271	47.54	11991	54.59	4145	52.63	12779	54	3881	51.48	9869
Leukemia1	3.24	299	4.21	1059	13.29	94	3.06	927	12.06	179	2.35	945
Leukemia2	15.13	6061	2.87	3895	11.73	1630	0.13	1136	25.4	2506	0.6	1232
Lung1	21.78	121	26.78	523	19.53	83	18.97	119	19.94	604	17.19	571
Lung2	1.4	2303	3.58	7329	4.43	4275	1.79	3409	4.45	4419	2.13	3148
median	9.2		12.78		11.73		6.8		11.84		6.73	

Chapter 6

Summary and Future Research

6.1 Summary

In this dissertation, we studied different approaches to improve the performance of the widely used Prediction Analysis of Microarrays (PAM) by Tibshirani et al. (2002). We investigated three problems of the PAM algorithm: retaining too many features, estimating the thresholding parameter, and handling the heteroscedastic situation. The proposed approaches in this dissertation present alternative methods that alleviate these problems of the PAM algorithm. Retaining too many features is a result of the soft thresholding used in the PAM and hence our first approach was to consider different thresholding methods, hard and order thresholding. The estimation process of the thresholding parameter in the PAM algorithm could have a potential problem of missing the optimal thresholding parameter. This is a result of considering only a finite number of thresholding values in the PAM algorithm while the parameter space is continuous. The risk of using finite numbers of thresholding values will increase when considering the smallest cross-validation error as a single selection criteria. This can be seen by comparing the numbers of informative genes corresponding to the smallest and 2nd smallest cross-validation errors as shown at the beginning of Chapter 3. To overcome this problem we take into consideration how likely the smallest cross-validation error approximates the true error. In particular, we consider the 2nd smallest cross-validation error in our second approach and compare it to the small-

est cross-validation error. In addition, refining the neighborhood of the initially selected thresholding value might result in a better thresholding parameter estimate. These ideas are implemented in our deep search algorithm.

Additionally, in this dissertation we compared the data driven thresholding parameter estimation method via cross-validation to three different thresholding parameter estimates that were suggested in literature. The overall comparison using a real data analysis was in favor of the thresholding parameter estimates obtained from cross-validation. However, the result of this analysis shows that none of the four thresholding parameter estimates are absolutely the best in every single dataset. Therefore, we derived an approximation for the probability of misclassification for the hard thresholding version algorithm in the two classes problem. Such approximation can be considered as a firm foundation upon which future research for estimating the optimal thresholding parameter can be based.

The PAM algorithm assumes homogeneity over different classes for the same variable in both variable selection and classification. However,in this research, we have shown that this assumption is not reasonable for many data sets. Therefore, we derived the test statistics and discriminant functions based on the heteroscedastic models. The test statistics are derived to compare the difference between the class centroid and the overall centroid. The discriminant function was derived from the posterior probability of the class label given the new sample under the assumption of heterogeneity similar to the Naive Bayes classifier. As a result of this work we provided heteroscedastic version algorithm for each of the three previously mentioned thresholding methods. It is very hard to validate the assumption of homogeneity for all variables in the high dimensional data because of the large number of variables involved. Thus, the devolvement of heteroscedastic version algorithms is a major contribution for the field of high dimensional classification.

Even though it has been shown in many studies that it is unexpected for a single method to outperform all other methods in all cases, our deep search algorithm results in some significant decrease in the number of selected genes. At the same time, the algorithms with deep search resulted in similar test errors to their counterpart without deep search

in that the difference in the test errors is no more than 2%. Moreover, the additional two thresholding methods we considered, hard and order thresholding, not only resulted in much more parsimonious models with significantly smaller number of genes, but also achieved better or at least comparable cancer status prediction accuracy for several data sets. From the results presented in this dissertation, it is very clear that the choice of thresholding method is of great importance for cancer classification. Also, our simulation results show the importance of acknowledging the heteroscedasticity in the classification process. Therefore, the algorithms in this dissertation provide important improvement for the PAM algorithm and they are very useful algorithms to be used in high dimensional multi-class classification problems.

6.2 Future research

First of all, since the choice of the thresholding method is of great importance for the classification, we would be interested in applying our proposed algorithms for real high dimensional data from other fields than the cancer classification using microarray data. This might lead us to identify the thresholding method that works best for each field or each specific type of data. Also, when using different data sets the new algorithms will be vetted for their usefulness and validity.

Second, the derivation of the PAM algorithm and the new algorithms proposed in this dissertation assume that the predictors are independent from each other. Thus, future research of interest would be studying the performance of the proposed algorithms when there is violation for this assumption.

In Chapter 4, for computational simplicity, we considered the two classes case (binary classification) to derive the probability of misclassification for the hard thresholding algorithm. The extension of this work would be considering the multi-class case. Furthermore, deriving approximations for the other two thersholding methods, soft and order threshold-

ing, is of great interest. By obtaining these approximations, in additional to the one derived in this dissertation for the hard thresholding case will provide another approach for comparing the theoretical performance of the different thresholding methods.

One more topic of interest comes from the fact that PAM and the algorithms in this dissertation assume that the predictors have a continues Gaussian distribution for deriving the classifier discriminant function. It would be interesting to study the performance of the proposed algorithms in this dissertation when the predictors follow different distributions. In particular, how robust is the performance of these algorithms when some predictors have discrete distributions. Also, it would be of interest to derive different discriminant functions for the different distributions if necessary.

Bibliography

Alizadeh, A., Eisen, M., Davis, R., Ma, C., Lossos, I., Rosenwald, A., Boldrick, J., Sabet, H., Tran, T., Yu, X., Powell, J., Yang, L., Marti, G., Moore, T., Hudson, J. J., Lu, L., Lewis, D., Tibshirani, R., Sherlock, G., Chan, W., Greiner, T., Weisenburger, D., Armitage, J., Warnke, R., Levy, R., Wilson, W., Grever, M., Byrd, J., Botstein, D., Brown, P., and Staudt, L. (2000). Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769):503–511.

Armstrong, S. A., Staunton, J. E., Silverman, L. B., Pieters, R., den Boer, M. L., Minden, M. D., Sallan, S. E., Lander, E. S., Golub, T. R., and Korsmeyer, S. J. (2002). MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia. *Nature Genetics*, 30:41–47.

Bair, E., Hastie, T., Paul, D., and Tibshirani, R. (2006). Prediction by supervised principal components. *Journal of the American Statistical Association*, 101:119–137.

Beer, D. G., Kardia, S. L., Huang, C.-C., Giordano, T. J., Levin, A. M., Misek, D. E., Lin, L., Chen, G., Gharib, T. G., Thomas, D. G., Lizyness, M. L., Kuick, R., Hayasaka, S., Taylor, J. M., Iannettoni, M. D., Orringer, M. B., and Hanash, S. (2002). Gene-expression profiles predict survival of patients with lung adenocarcinoma. *Nature Medicine*, 8(8):816–824.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300.

Bhattacharjee, A., Richards, W. G., Staunton, J., Li, C., Monti, S., Vasa, P., Ladd, C., Beheshti, J., Bueno, R., Gillette, M., Loda, M., Weber, G., Mark, E. J., Lander, E. S., Wong, W., Johnson, B. E., Golub, T. R., Sugarbaker, D. J., and Meyerson, M. (2001).

Classification of human lung carcinomas by mrna expression profiling reveals distinct adenocarcinoma subclasses. *Proceedings of the National Academy of Sciences*, 98(24):13790–13795.

Bickel, P. (1983). Minimax estimation of the mean of a normal distribution subject to doing well at a point. In *Recent Advances in Statistics*, Pap. in Honor of H. Chernoff, Academic Press, New York, 511-528.

Bishop, Y. M., Fienberg, S. E., and Holland, P. W. (2007). *Discrete multivariate analysis*. Springer, New York.

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

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

Cai, T. T. (2002). On block thresholding in wavelet regression: adaptivity, block size, and threshold level. *Statistica Sinica*, 12(4):1241–1273.

Cai, T. T. and Silverman, B. W. (2001). Incorporating information on neighbouring coefficients into wavelet estimation. *Sankhya Series B*, 63:127–148.

Chang, C.-C. and Lin, C.-J. (2011). Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):1–27.

Crammer, K. and Singer, Y. (2001). On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292.

Donoho, D. L. (2000). High-dimensional data analysis: the curses and blessings of dimensionality. In *American Mathematical Society Conf. Math Challenges of the 21st Century*.

Donoho, D. L. and Johnstone, I. M. (1995). Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, 90(432):1200–1224.

Donoho, D. L. and Johnstone, J. M. (1994). Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455.

Dumais, S. and Chen, H. (2000). Hierarchical classification of web content. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '00, pages 256–263, New York, NY, USA. ACM.

Fan, J. (1996). Test of significance based on wavelet thresholding and Neyman's truncation. *Journal of the American Statistical Association*, 91(434):674–688.

Fan, J. and Fan, Y. (2008). High-dimensional classification using features annealed independence rules. *Annals of Statistics*, 36:2605–2637.

Fan, J., Li, R., Sanz-Sole, M., Soria, J., Varona, J. L., and Verdera, J. (2006). Statistical challenges with high-dimensionality: Feature selection in knowledge discovery. In *International congress of mathematicians*, volume III, pages 595–622. American Mathematical Society, Providence, RI. Book, Section.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119 – 139.

Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloomfield, C., and Lander, E. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537.

Hall, P., Pittelkow, Y., and Ghosh, M. (2008). Theoretical measures of relative performance of classifiers for high dimensional data with small sample sizes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1):159–173.

Hastie, T. and Tibshirani, R. (1998). Classification by pairwise coupling. In *Proceedings of the 1997 conference on Advances in neural information processing systems 10*, NIPS '97, pages 507–513, Cambridge, MA, USA. MIT Press.

Héberger, K. (2010). Sum of ranking differences compares methods or models fairly. *Trends in Analytical Chemistry*, 29(1):101–109.

Héberger, K. and Kollár-Hunek, L. (2011). Sum of ranking differences for method discrimination and its validation: comparison of ranks with random numbers. *Journal of Chemometrics*, 25:151–158.

Hsu, C.-W. and Lin, C.-J. (2002). A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on*, 13(2):415–425.

Johnstone, I. M. and Silverman, B. W. (2004). Needles and straw in haystacks: empirical Bayes estimates of possibly sparse sequences. *Annals of Statistics*, 32:1594–1649.

Khan, J., Wei, J. S., Ringner, M., Saal, L. H., Ladanyi, M., Westermann, F., Berthold, F., Schwab, Manfred Antonescu, C. R., Peterson, C., and Meltzer, P. S. (2001). Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, 7(6):673–679.

Kim, M. and Akritas, M. (2010). Order thresholding. *Annals of Statistics*, 38(4):2314–2350.

Kim, M. and Akritas, M. (2012). Goodness-of-fit testing: the thresholding approach. *Journal of Nonparametric Statistics*, 24:119–138.

Knerr, S., Personnaz, L., and Dreyfus, G. (1990). Single-layer learning revisited: a stepwise procedure for building and training a neural network. In Souli, F. and Hrault, J., editors, *Neurocomputing*, volume 68 of *NATO ASI Series*, pages 41–50. Springer Berlin Heidelberg.

Kohavi, R. and John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97(1):273–324.

Lee, Y., Lin, Y., and Wahba, G. (2004). Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association*, 99:67–81.

Neyman, J. (1937). Smooth test for goodness of fit. *Skandinavisk Aktuarietidskrift*, 20:149–199.

Perou, C. M., Sorlie, T., Eisen, M. B., van de Rijn, M., Jeffrey, S. S., Rees, C. A., Pollack, Jonathan R. Ross, D. T., Johnsen, H., Akslen, L. A., Fluge, O., Pergamenschikov, A., Williams, C., Zhu, S. X., Lonning, P. E., Borresen-Dale, A.-L., Brown, P. O., and Botstein, D. (2000). Molecular portraits of human breast tumours. *Nature*, 406:747–752.

Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S., Yeang, C.-H., Angelo, M., Ladd, C., Reich, M., Latulippe, E., Mesirov, J. P., Poggio, T., Gerald, W., Loda, M., Lander, E. S., and Golub, T. R. (2001). Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Academy of Sciences*, 98(26):15149–15154.

Rifkin, R. and Klautau, A. (2004). In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141.

Statnikov, A. R., Tsamardinos, I., Dosbayev, Y., and Aliferis, C. F. (2005). Gems: A system for automated cancer diagnosis and biomarker discovery from microarray gene expression data. *International Journal of Medical Informatics*, 74(7-8):491–503.

Su, A., Welsh, J., Sapinoso, L., Kern, S., Dimitrov, P., Lapp, H., Schultz, P., Powell, S., Moskaluk, C., Frierson, H. J., and Hampton, G. (2001). Molecular classification of human carcinomas by use of gene expression signatures. *Cancer Research*, 61(20):7388–7393.

Tan, A. C., Naiman, D. Q., Xu, L., Winslow, R. L., and Geman, D. (2005). Simple decision rules for classifying human cancers from gene expression profiles. *Bioinformatics*, 21(20):3896–3904.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, B 58:267–288.

Tibshirani, R., Hastie, T., Narasimhan, B., and Chu, G. (2002). Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences*, 99(10):6567–6572.

Vapnik, V. N. (1998). *Statistical learning theory*. Wiley-Interscience, Canada, 1st edition.

Weston, J. and Watkins, C. (1999). Support vector machines for multi-class pattern recognition. in proceedings of *the 7th european symposium on artificial neural networks*. pages 219–224.

Yeoh, E., Ross, M. E., Shurtleff, S. A., Williams, W. K., Patel, D., Mahfouz, R., Behm, F. G., Raimondi, S. C., Relling, M. V., Patel, A., Cheng, C., Campana, D., Wilkins, D., Zhou, X., Li, J., Liu, H., Pui, C., Evans, W. E., Naeve, C., Wong, L., and Downing, J. R. (2002). Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. *Cancer Cell*, 1(2):133 – 143.

Zhang, H., Wang, H., Dai, Z., Chen, M.-s., and Yuan, Z. (2012). Improving accuracy for cancer classification with a new algorithm for genes selection. *BMC Bioinformatics*, 13:1–20.

Zhang, H. H., Ahn, J., and Lin, X. (2006). Gene selection using support vector machines with nonconvex penalty. *Bioinformatics*, 22:88–95.

Appendix A

Heatmaps for heterogeneity among different classes

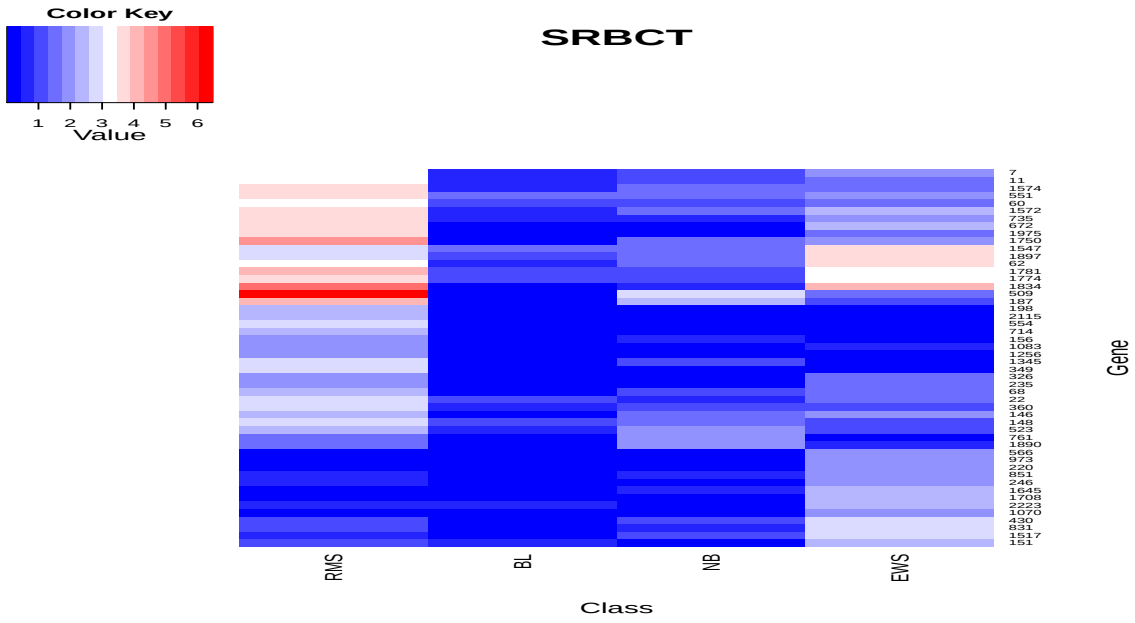


Figure A.1: *Heatmap of the sample standard deviation for 50 genes from the SRBCT cancer dataset. These are for the top 50 genes that have the highest range of the standard deviations for different classes.*

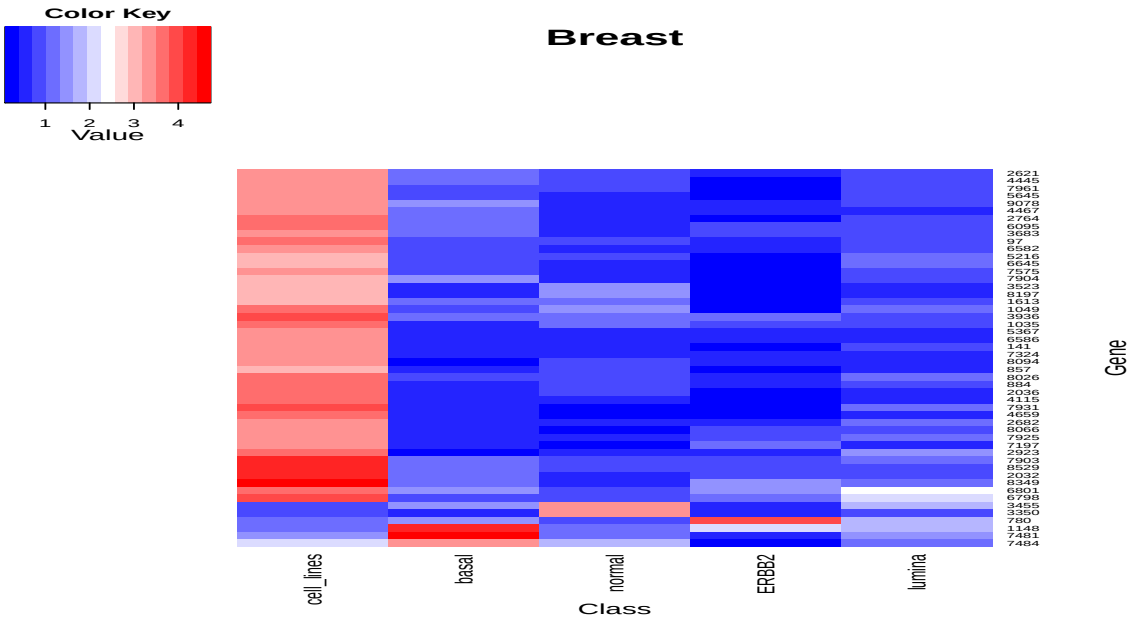


Figure A.2: *Heatmap of the sample standard deviation for 50 genes from the Breast cancer dataset. These are for the top 50 genes that have the highest range of the standard deviations for different classes.*

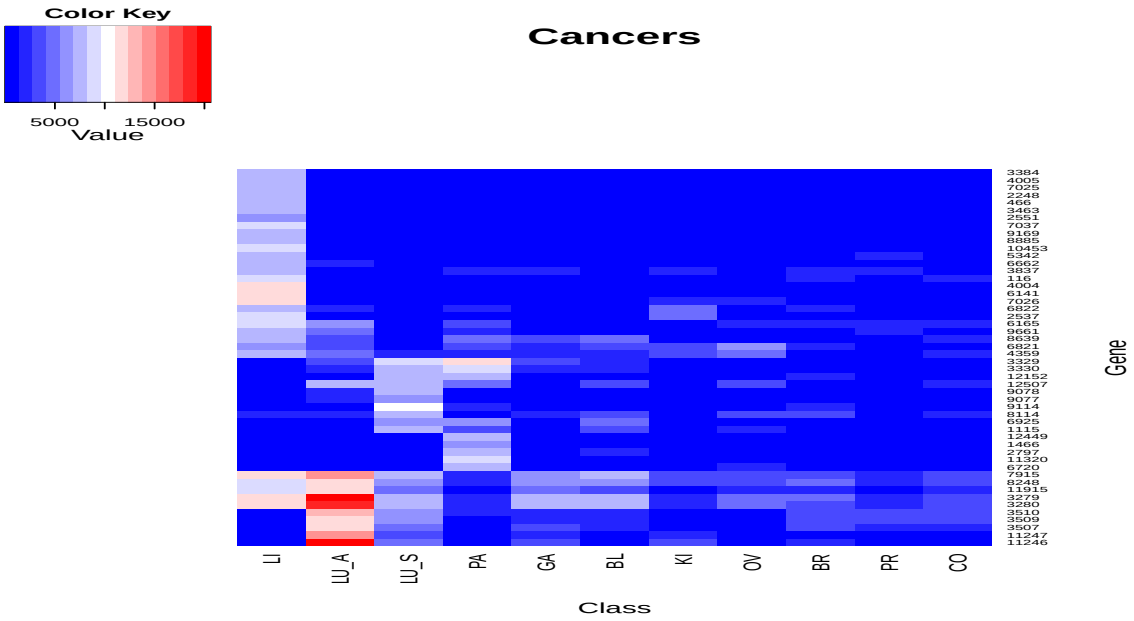


Figure A.3: *Heatmap of the sample standard deviation for 50 genes from the Cancers dataset. These are for the top 50 genes that have the highest range of the standard deviations for different classes.*

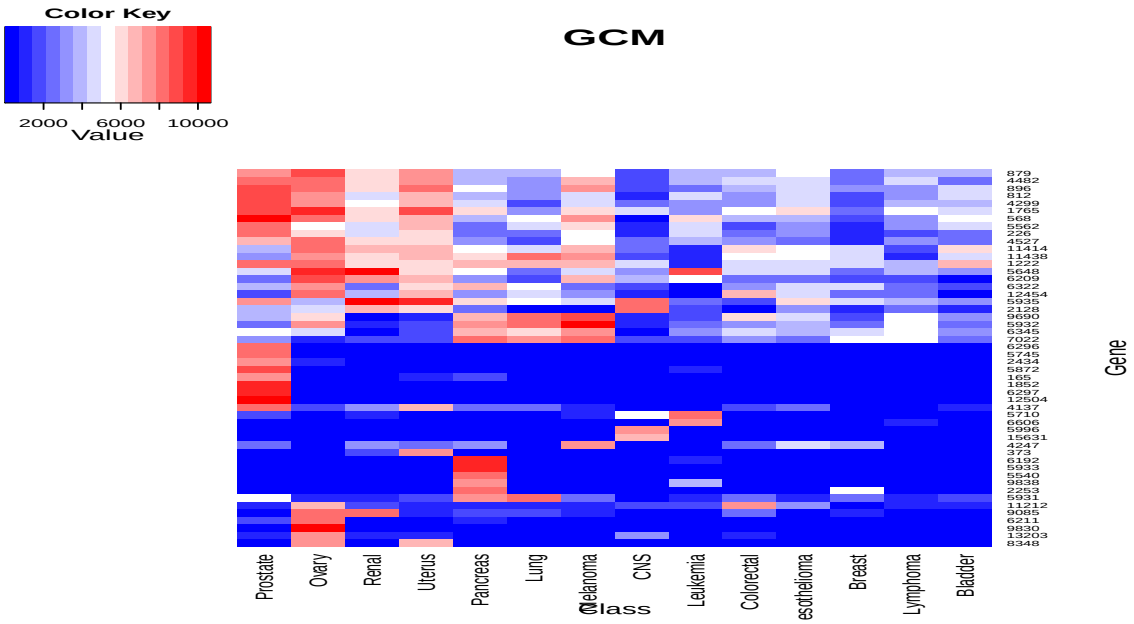


Figure A.4: *Heatmap of the sample standard deviation for 50 genes from the GCM cancer dataset. These are for the top 50 genes that have the highest range of the standard deviations for different classes.*

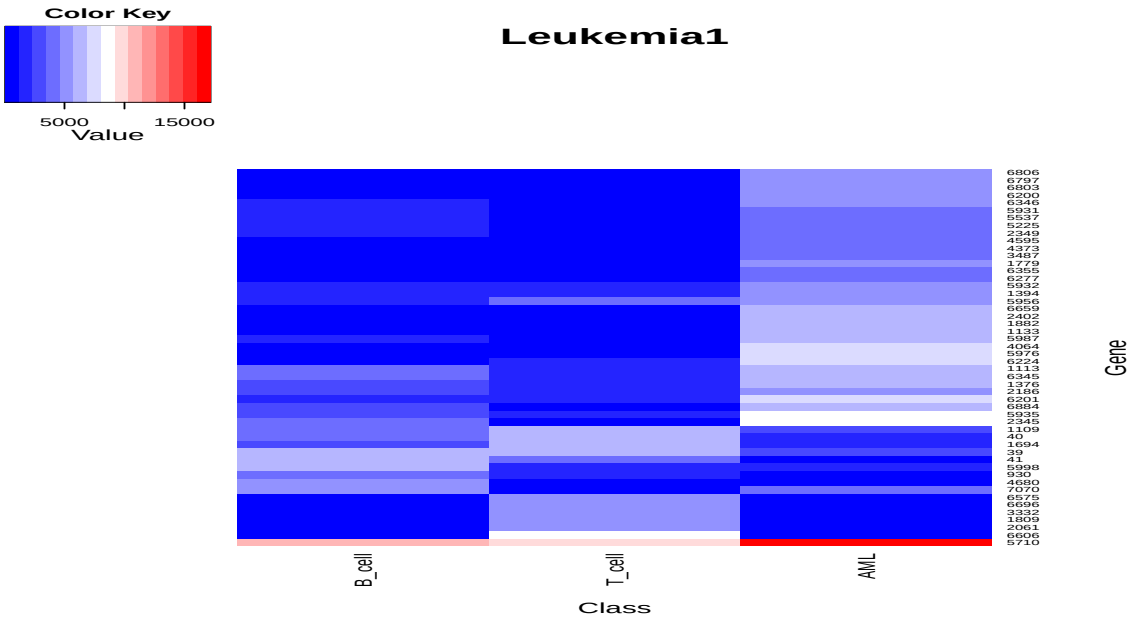


Figure A.5: *Heatmap of the sample standard deviation for 50 genes from the Leukemia1 cancer dataset. These are for the top 50 genes that have the highest range of the standard deviations for different classes.*

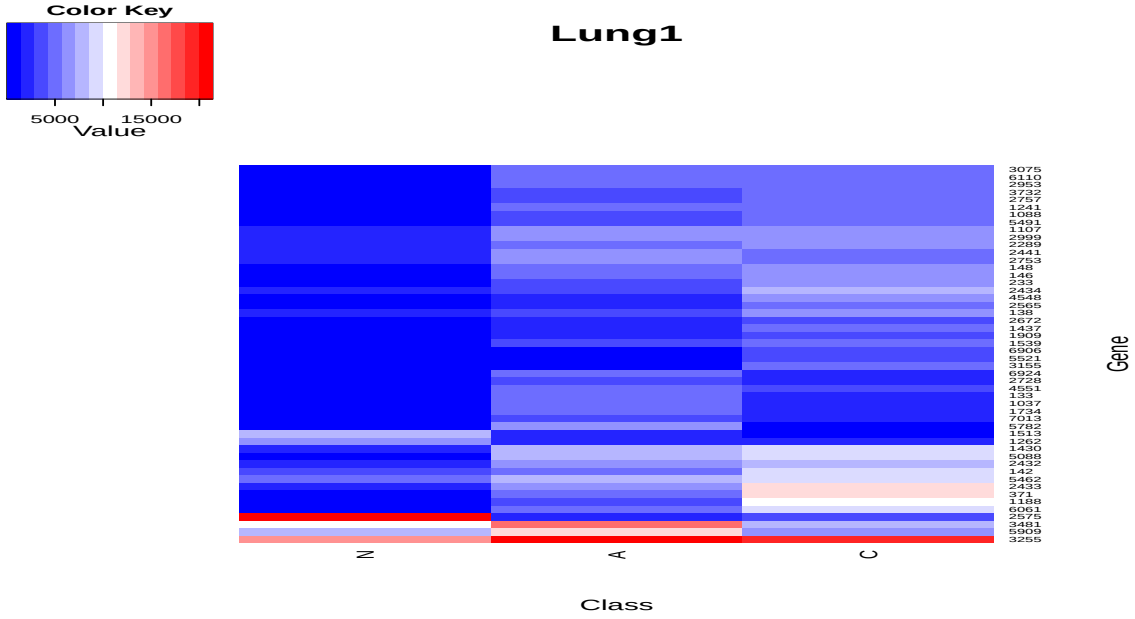


Figure A.6: *Heatmap of the sample standard deviation for 50 genes from the Lung1 cancer dataset. These are for the top 50 genes that have the highest range of the standard deviations for different classes.*