

# HIGH-DIMENSIONAL CLASSIFICATION AND ATTRIBUTE-BASED FORECASTING

A Thesis  
Presented to  
The Academic Faculty

by

Shin-Lian Lo

In Partial Fulfillment  
of the Requirements for the Degree  
Doctor of Philosophy in the  
School of Industrial and Systems Engineering

Georgia Institute of Technology  
December 2010

Copyright © 2010 by Shin-Lian Lo

# HIGH-DIMENSIONAL CLASSIFICATION AND ATTRIBUTE-BASED FORECASTING

Approved by:

Professor Kwok-Leung Tsui, Advisor  
School of Industrial and Systems  
Engineering  
*Georgia Institute of Technology*

Professor Ying Hung, Co-advisor  
Department of Statistics  
*Rutgers, The State University of New  
Jersey*

Professor David M. Goldsman  
School of Industrial and Systems  
Engineering  
*Georgia Institute of Technology*

Professor Ming Yuan  
School of Industrial and Systems  
Engineering  
*Georgia Institute of Technology*

Professor Kobi A. Abayomi  
School of Industrial and Systems  
Engineering  
*Georgia Institute of Technology*

Date Approved: 25 August 2010

*To my parents,  
for their love, support, and encouragement  
in this challenging journey.*

## ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor, Dr. Kwok-Leung Tsui, for his guidance, encouragement, and support in all aspects through my Ph.D. study. Every discussion with him inspired and helped me toward the achievement of this milestone. I would also like to express my appreciation to my co-advisor, Dr. Ying Hung, for her constant assistance and deep discussion through my research. Without her inspiration and guidance, the accomplishment of this dissertation would not be possible.

I am also thankful to Dr. David Goldsman, Dr. Ming Yuan, and Dr. Kobi Abayomi for serving on my dissertation committee. Their valuable suggestions and comments make this dissertation more complete. My special thanks go to Dr. Pelin Pekgun for her extended assistance during my last year of the doctoral program. It was my great fortunate to get the opportunity and work on several interesting research problems.

I would like to extend my gratitude to all my friends at the Georgia Institute of Technology for their continued care and help in my doctoral student life. I also thank to my friends, Dr. Chien-Yu Peng and Dr. Chih-Chun Tsai, for their encouragement and companion that make my foreign life more delightful. Last but not least, I would like to express my deepest appreciation to my parents for their endless love and support. They give me strength and help me through difficult times in this challenging journey.

# TABLE OF CONTENTS

DEDICATION . . . . .	iii
ACKNOWLEDGEMENTS . . . . .	iv
LIST OF TABLES . . . . .	viii
LIST OF FIGURES . . . . .	x
SUMMARY . . . . .	xi
I INTRODUCTION OF MICROARRAY EXPERIMENTS AND CLASSIFI- CATION PROBLEMS . . . . .	1
1.1 Bioinformatics and Microarray Experiments . . . . .	1
1.2 Statistical Analysis Issues in Microarray Experiments . . . . .	4
1.3 Classification Problems on Gene Expression Data . . . . .	5
II SOME EXISTING VARIABLE SELECTION AND CLASSIFICATION METH- ODS . . . . .	11
2.1 Variable Selection Methods . . . . .	11
2.1.1 Fold Change . . . . .	12
2.1.2 Individual T-test . . . . .	12
2.1.3 Multiple Test . . . . .	13
2.1.4 False Discovery Rate . . . . .	14
2.1.5 Other Univariate Ranking . . . . .	15
2.1.6 Correlation-based Ranking . . . . .	15
2.2 Classification Methods . . . . .	16
2.2.1 Linear Discriminant Analysis . . . . .	16
2.2.2 Logistic Regression . . . . .	17
2.2.3 Classification Tree . . . . .	18
2.2.4 $k$ -Nearest Neighbor . . . . .	18
2.3 Example: Leukemia Data . . . . .	19
2.4 Summary . . . . .	21

III	A NEW CLASSIFICATION APPROACH: ITERATIVE RESELECTION PENALIZED LOGISTIC REGRESSION	23
	3.1 Review of Embedded Approaches . . . . .	23
	3.2 Motivation and Features of New Approach . . . . .	24
	3.3 Iterative Reselection Penalized Logistic Regression . . . . .	26
	3.3.1 Penalized Logistic Regression . . . . .	26
	3.3.2 Iterative Reselection Algorithm . . . . .	28
	3.4 Performance Assessment . . . . .	33
	3.4.1 Estrogen and Lymph Data . . . . .	33
	3.4.2 Leukemia Data . . . . .	36
	3.4.3 Breast Cancer Data . . . . .	38
	3.5 Simulation Study . . . . .	41
	3.6 Conclusion and Discussion . . . . .	45
IV	A NEW MODELING METHOD: PENALIZED LOGISTIC MIXED MODEL . . . . .	48
	4.1 Replication of Microarray Experiments . . . . .	48
	4.2 Motivation and Literature Review . . . . .	49
	4.3 Penalized Logistic Mixed Model . . . . .	52
	4.4 MCEM Algorithm . . . . .	55
	4.5 Selection Consistency . . . . .	58
	4.6 Simulation Study . . . . .	60
	4.7 Application: Breast Cancer Study . . . . .	66
	4.8 Concluding Remarks . . . . .	68
V	HIERARCHICAL ATTRIBUTE-BASED FORECASTING . . . . .	70
	5.1 Introduction . . . . .	70
	5.2 Review of Existing Tree-based Methods . . . . .	74
	5.2.1 CART . . . . .	75
	5.2.2 CHAID . . . . .	77
	5.2.3 GUIDE . . . . .	78

5.2.4	Summary . . . . .	80
5.3	Hierarchical Attribute-based Forecasting . . . . .	81
5.3.1	Hierarchical Splitting . . . . .	81
5.3.2	Stopping and Pruning . . . . .	83
5.3.3	Interpretability . . . . .	85
5.3.4	Comparison of Tree-based methods . . . . .	86
5.4	Application . . . . .	88
5.4.1	Capacity Forecasting in the Air Cargo Industry . . . . .	88
5.4.2	Sensitivity Analyses . . . . .	92
5.5	Simulation Study . . . . .	95
5.5.1	Settings . . . . .	96
5.5.2	Results . . . . .	97
5.6	Conclusion and Discussion . . . . .	101
VI	FUTURE WORK . . . . .	103
APPENDIX A	PROOF OF THEOREM 1 . . . . .	105
REFERENCES	. . . . .	110

## LIST OF TABLES

1	The diagrams of the filter, wrapper, and embedded approaches . . . . .	10
2	Outcomes from $p$ hypothesis tests . . . . .	14
3	Comparison of selected genes by T-test, Bonferroni test, and FDR . . .	20
4	Comparison of classification methods with different ranking criteria . .	21
5	Recent related work using embedded approaches for microarray data	24
6	Comparison of classification methods for breast tumors based on ER status . . . . .	34
7	Comparison of classification methods for breast tumors based on LN status . . . . .	35
8	Selected genes for classifying leukemia data . . . . .	37
9	Comparison of classification methods for leukemia data . . . . .	38
10	Comparison of classification methods for leukemia data through 50 new splits . . . . .	39
11	Comparison of classification methods for breast tumors with NNN / non-NNN subtypes . . . . .	41
12	Description of top 10 genes for breast tumors with NNN / non-NNN subtypes . . . . .	42
13	Comparison of PLRL <sub>1</sub> and IRPLRL <sub>1</sub> ( $c = 300$ ) with simulated data . . .	44
14	Related theoretical work of penalized regression models . . . . .	51
15	Comparison of the Lasso and adaptive Lasso penalties in PLMM . . . .	63
16	Comparison of the Lasso and adaptive Lasso penalties in PLMM and PLR in cancer study . . . . .	67
17	Comparison of tree-based methods . . . . .	86
18	Differences between CHAID and HABF . . . . .	87
19	Data attributes and categories . . . . .	88
20	ANOVA table . . . . .	89
21	Forecasting performance of different tree-based methods . . . . .	91
22	Simulation settings . . . . .	97
23	Simulation results ( $\sigma^2 = 1$ ) . . . . .	99

24	Simulation results ( $\sigma^2 = 25$ ) . . . . .	100
----	--	-----

## LIST OF FIGURES

1	A typical process of a microarray experiment (Wong, 2005) . . . . .	3
2	Iterative reselection algorithm . . . . .	31
3	The diagrams of the new approach with three existing approaches . .	32
4	Comparison of convergence in SA and a descent search. . . . .	35
5	Top 10 active genes in terms of the frequency identified by IRPLRL <sub>1</sub>	41
6	Comparison of PLRL <sub>1</sub> and IRPLRL <sub>1</sub> ( $p = 3000$ ) . . . . .	45
7	Comparison of PLRL <sub>1</sub> and IRPLRL <sub>1</sub> ( $p = 5000$ ) . . . . .	46
8	Comparison of PLMM with Lasso and adaptive Lasso ( $p = 200$ ) . . .	64
9	Comparison of PLMM with Lasso and adaptive Lasso ( $p = 400$ ) . . .	65
10	Comparison of PLMM with Lasso and adaptive Lasso ( $p = 3000$ ) . .	65
11	Comparison of PLMM with Lasso and adaptive Lasso ( $p = 5000$ ) . .	66
12	Forecasting errors generated by different methods . . . . .	93
13	Forecasting errors vs. length of history . . . . .	94
14	Percentage of testing samples forecasted by all significant predictors vs. length of history . . . . .	94
15	Forecasting errors vs. sample size and time effects . . . . .	95

## SUMMARY

This thesis consists of two parts. The first part focuses on high-dimensional classification problems in microarray experiments. The second part deals with forecasting problems with a large number of categories in predictors.

The first part of this thesis contains four chapters. The first chapter provides an overall introduction of microarray experiments and associated classification issues. The second chapter reviews some existing variable selection and classification methods. The third chapter develops a new classification approach to maintain variable selection consistency and classification accuracy in high dimensionality. The fourth chapter proposes a new classification method in the consideration of different variability among experimental observations. The second part of this thesis is included in chapter five, where a new forecasting approach that deals with a large number of categories in predictors and takes into account predictor structures is developed.

Classification problems in microarray experiments refer to discriminating subjects with different biologic phenotypes or known tumor subtypes as well as to predicting the clinical outcomes or the prognostic stages of subjects. A typical microarray experiment monitors the expression levels of thousands of genes taken from tens of subjects. Due to the large number of genes with a relatively small sample size, most traditional classification methods require preliminary variable selection before being employed for classification. As a result, the classification accuracy of such methods strongly relies on the choice of the pre-selected variables. Different from traditional classification methods, the penalized logistic regression method is known for simultaneous variable selection and classification. However, the performance of this method declines as the number of variables increases. With this concern, in chapter three, we

propose a new classification approach that employs the penalized logistic regression method iteratively with a controlled size of gene subsets to maintain variable selection consistency and classification accuracy. Moreover, we incorporate a randomized heuristic algorithm that efficiently searches for the optimal gene subset without an exhaustive search. The performance of the new classification approach is evaluated and compared with existing methods through four real-world microarray datasets and a simulation study. The results show that the new approach outperforms the existing methods in terms of gene selection and classification accuracy.

The research described in chapter four is motivated by a modern microarray experiment that includes two layers of replicates. This new experimental setting causes most existing classification methods, including penalized logistic regression, not appropriate to be directly applied because the correlations among replicates violate the assumption of independent observations in penalized logistic regression. To solve this problem, we propose a new classification method by incorporating random effects into penalized logistic regression such that the heterogeneity among different experimental subjects and the correlations from repeated measurements can be taken into account. The proposed method, however, poses computational challenges because the high-dimensional integrals over the distribution of random effects can not be expressed in a closed form. Therefore, an efficient hybrid algorithm is introduced to tackle the difficulties in estimation and integration over random effect distributions. The theoretical results of variable selection consistency is also presented, and the finite sample performance is examined via a simulation study. Applications to a modern microarray experiment in breast cancer study show that the proposed classification method obtains smaller models with higher prediction accuracy than the method based on the assumption of independent observations.

In chapter five, we propose a new forecasting approach for large-scale datasets associated with a large number of predictor categories and with observed predictor

structures. The new approach is similar to tree-based methods that grow a number of nodes through splitting and adopt piecewise constant prediction at terminal nodes. However, conventional tree-based methods do not accommodate intrinsic predictor structures, and they are not generally considered efficient to deal with a large number of categorical values in predictors. Beyond the conventional tree-based methods, the new approach incorporates observed predictor structures by a general linear model and multi-way hierarchical splits to make the grown trees more comprehensive, efficient, and interpretable. Through an empirical study of a capacity forecasting problem in the air cargo industry, we show that the new approach has higher forecasting accuracy and higher computational efficiency than existing tree-based methods consistently over time. Furthermore, we investigate the performance of the new approach under different circumstances via a simulation study. The simulation results show that the forecasting accuracy and the computational efficiency of the new approach is less influenced by the number of predictor categories and the irrelevant predictors than existing tree-based methods.

# CHAPTER I

## INTRODUCTION OF MICROARRAY EXPERIMENTS AND CLASSIFICATION PROBLEMS

The first part of this thesis deals with classification problems in microarray experiments. In this chapter, we provide background knowledge and discuss the statistical analysis issues in microarray experiments in the first two sections. Thereafter, we examine the classification problems and the challenges in gene expression microarray data. We also review the existing approaches of variable selection and classification in the third section.

### *1.1 Bioinformatics and Microarray Experiments*

*Bioinformatics* is an interdisciplinary research field in which biology, statistics, and computer science interact to manage, analyze, and understand large amounts of biological data using databases, computational and statistical techniques, and theories. The primary goal of bioinformatics is to increase our understanding of biological processes, particularly in response to rapid advancements in molecular biology and genomics. Related applications have also become popular and important nowadays, and thus they can be rightly singled out into separate fields, in which many opportunities have been emerging for research work. An comprehensive overview of current research topics in bioinformatics can be referred to Rzhetsky (2008).

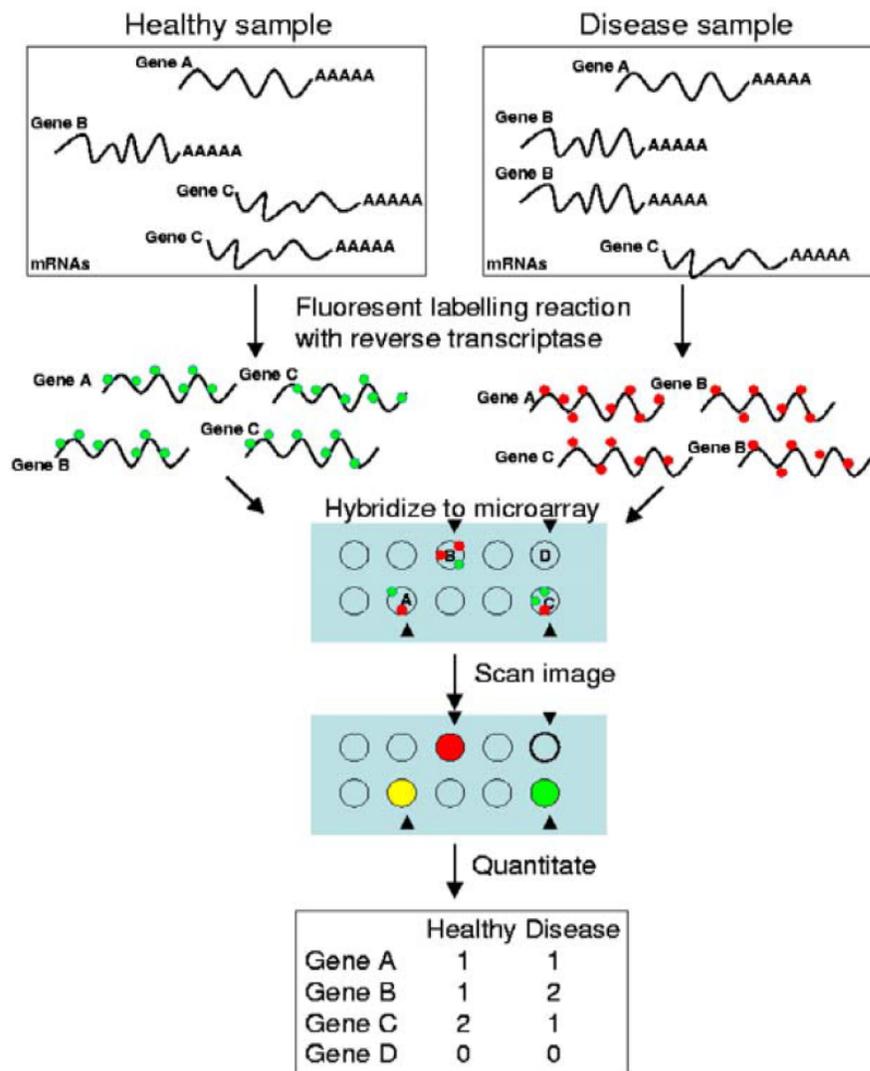
Scientists use a technique to monitor and analyze information contained in a genome called *microarrays*. The type of microarray depends on the material, such as DNA (deoxyribonucleic acid), RNA (ribonucleic acid), protein, or tissue, spotted on the microscope slides. For example, DNA microarrays are part of a promising class

of biotechnologies (Speed, 2003) that allow simultaneous monitoring of expression levels in cells for thousands of genes (the units of the DNA sequence that control the identifiable heredity traits of an organism). DNA microarray technology has been applied to a number of investigations, particularly in the study of genomics and cancer. For instance, high-throughput microarrays can be used as a screen for early detection of disseminated breast tumor cells in peripheral blood (Martin et al., 2001). The use of DNA microarray can also benefit many other fields, such as pharmacology, specifically drug discovery and toxicological research (Shi, 2002).

*Microarray experiments* consist of multiple steps. A typical process of a microarray experiment is exhibited in Figure 1. RNA (the transcription of DNA) samples are first extracted from tissues or cells by common organic extraction procedures used in molecular biology experiments. Once RNA samples are extracted, direct-labeling of the RNA samples can be done by producing complementary DNA (cDNA) from RNA with enzyme reverse transcriptase and by incorporating fluorescent labels for hybridization. Hybridizing fluorescently-labeled DNAs onto microarrays is similar to hybridizations in other molecular biology applications. After hybridization, microarrays are washed for several minutes in decreasing salt buffers and finally dried. The fluorescently-labeled microarrays can then be read by scanners, which give a relative expression amount of fluorescent emission from different represented transcripts (Wong, 2005). The more detailed description of microarray experiments and technology can be referred to Lee (2004).

The inherent nature of *microarray data* is that fewer samples or replicates compared to a large number of genes are involved in microarray experiments. In the past, a typical microarray experiment included thousands of genes but only tens of biological samples (i.e., subjects) due to the cost of microarray experiments and sample availability (Golub, 1999; West et al., 2001). With the advancements in microarray technology, recent experiments also include a few technical replicates in addition to

biological replicates (Lee et al., 2002; Yang et al., 2003) for the reason that it is inevitable to encounter technical problems or variability in any step of microarray experiments. For example, various systematic errors in microarray measurements may exist during the preparation of arrays and in the procedure of analyzing images. The common sources of variation in microarray experiments can be referred to Lee (2004). From the analytical perspective, technical replicates can offer the benefits of improving statistical precision and diagnostic checking.



**Figure 1:** A typical process of a microarray experiment (Wong, 2005)

## 1.2 *Statistical Analysis Issues in Microarray Experiments*

The high cost, high volume, and complex experimental artifacts associated with microarray data collection have emphasized the need for statistical analyses and techniques at all stages of experiments (Parmigiani et al., 2003). The issues of statistical analysis in microarray experiments can be classified into six components: design, pre-processing, comparison, clustering, classification, and trend analysis. Below we briefly describe the general purposes or important considerations for each component.

- *Experimental design* affects the efficiency and the internal validity of experiments (Kerr, 2003). Wong (2005) discussed several factors that one must account for when conducting experimental design and controls. One is to plan for sufficient replicates for the purpose of decreasing experimental error and providing statistical power. Another is to recognize the importance of experimental parameters. That is, regardless of the treatment, the time, the dosage, the individual, or the tissue location, the results should be interpretable with a minimum number of confounders. An additional consideration is to select the most optimal statistical practices and design procedures after considerable forethought and consultation.
- The inherent characteristics of measured intensities may affect data analysis results. In order to reduce systematic variation, one should conduct data processing prior to data analysis. The *preprocessing* steps usually include image analysis, normalization across microarrays, data transformation, and background subtraction. These steps allow data to be more consistent with the assumptions of the underlying follow-up studies.
- The *identification of differentially expressed genes* is of fundamental and practical interest. Research in biology and medicine may benefit from the examination of the identified genes to confirm recent discoveries in cancer research or suggest

new avenues to be explored. For instance, medical diagnostic tests that measure the abundance of a given protein in serum may be derived from a small subset of differentially expressed genes.

- *Clustering* techniques are often used to support visualization and as methods for generating hypotheses about the existence of gene groups or samples with similar behavior by exploring gene expression data. Some successful applications of clustering analysis include identifying novel cancer subtypes, discovering new gene classes in gene ontology, and generating heatmaps (the most commonly used visualization tool).
- *Classification* refers to discriminating samples with different biologic phenotypes (characteristics outward displayed) or known tumor subtypes as well as to predicting the clinical outcome or the prognostic stage of a patient using gene expression intensities as predictors. A closely related issue is to find a small group of genes that reliably generalizes beyond the sample analyzed. The accuracy of class prediction can then be assessed using a validation set or by cross-validation.
- *Time series analysis* can be used to identify genes that show similar trends over time within the same organism or sample type as well as to identify samples that are differentiated by such patterns. These analyses are often performed using regression, by which time is a primary predictor variable and gene expression is the outcome.

### ***1.3 Classification Problems on Gene Expression Data***

Among many statistical analysis issues described in the previous section, in this thesis, we focus on classification problems on microarray gene expression data, especially with binary outcomes. We first describe a typical classification problem with a real-life

example. Suppose we want to detect cancer cells by means of their genetic properties. As the human DNA has millions of genes, the following questions are arisen: Which of these genes are really useful for classifying a cell as “cancerous” or “normal”? Do we need 10, 50, 100, 5,000, 10,000, or more genes to solve this task? These questions lead to two fundamental problems for classification: How to detect useful genes and how to utilize these useful genes to construct a classifier.

Although classification is not a new subject in statistical literature, different from conventional classification problems in other fields, classification based on DNA microarray data raises more challenges. The major *challenges* are rooted in the huge number of genes with relatively small sample size taken in microarray experiments. Other challenges include many genes are not useful predictors for tissue types; on the contrary, they introduce noise in the classification process and thus potentially drown out the contributions of other useful genes. Moreover, for diagnostic purposes, it is important to find *small* subsets of genes that are sufficiently informative to distinguish samples between different cell types. Even though many genes are co-regulated (with the high degree of expression similarity) as they are mutually involved in disease pathways or have common upstream regulatory sequence patterns. From the statistical and computational perspective, these challenges induce the following problems:

- *Limitation of classification methods*: Most traditional classification methods, such as discriminant analysis and logistic regression, are not designed to cope with high-dimensional predictors with only a small number of samples and thus can not be directly applied to microarray data.
- *Overfitting*: When the number of variables is much larger than the number of samples, one can easily find a classifier that produces good prediction in training data but poor prediction in testing data. In such a situation, a model overfitting problem arises. In particular, when a classifier contains many irrelevant

variables, an overfitting problem could bring more risk.

- *Multi-collinearity*: A multi-collinearity problem occurs when highly-correlated variables are used in constructing a classifier. As a result, the classifier lacks robustness.

Two frequently used techniques for tackling these statistical problems are *dimension reduction* and *variable selection*. *Dimension reduction* techniques, such as principal component analysis and partial least squares, were employed in the literature and satisfactory performance was also reported (Ghosh, 2003). However, one disadvantage is that none of the original variables can be completely discarded when a classifier is constructed unless a preliminary variable selection step is performed (Nguyen and Rocke, 2002). Another drawback is that the super-composed variables do not necessarily have easy and clear biological interpretation. On the contrary, *variable selection* techniques select a subset of the original variables, instead of utilizing all variables. Compared with dimension reduction, variable selection techniques have potential benefits of (i) facilitating data visualization and data understanding, especially the interaction between genes and the response class; (ii) helping biologists discover unrevealed genes; (iii) reducing gene expression measurements and storage requirements; and (iv) providing more cost-effective predictors for further study. With these potential advantages, variable selection is usually considered more favorable than dimension reduction in microarray classification studies.

In the context of classification, variable selection techniques can be categorized according to how they combine or integrate with classification models into three approaches: the filter, wrapper, and embedded approaches (Blum and Langley, 1997; Saeys et al., 2007). The first, the *filter approach*, is named by Kohavi and John (1997) in that variable selection is independent of classification. Many variable selection approaches proposed in the past were based on variable ranking techniques, either

univariate or multivariate, depending on whether the interdependence of variables is considered. Some common ranking techniques include the fold change, the T-test, the F-test, the false discovery rate (Benjamini and Hochberg, 1995), correlation-based feature selection (Hall et al., 1999), and variants. A complete review of ranking techniques can be found in Saeys et al. (2007). In practice, selection of the top  $g$ -ranked variables for some arbitrary  $g$  is a common way to build a classifier. Alternatively, a score threshold is set and only variables whose scores exceed the threshold are selected. Although classifiers are then built based on the pre-selected variables that are independent of classification, the filter variable selection approach prevails in practice for it is easily scalable to high dimension, easy to understand, and less computationally intensive. Thus, the filter variable selection approach is commonly used as a baseline method for classification problems.

In contrast to the *filter approach*, another class of variable selection approach, the *wrapper approach* (Blum and Langley, 1997; Kohavi and John, 1997), takes into account the interaction with classification models and evaluates variables until certain classification accuracy is satisfied. As variables are selected around classification, this approach usually incorporates a search algorithm that finds an optimal variable subset for classification such that it can achieve better classification performance than the filter approach but with added cost for computational efforts in searching. Since the number of variable subsets is extremely large in microarray data and the space of variable subsets exponentially grows with the number of variables, it is suggested that a greedy or randomized search is a better choice than an exhaustive search. A greedy search, such as forward selection, backward elimination, and hill-climbing, favors fast computation but risks at getting stuck in a local optimum. In contrast, a randomized heuristic search, such as the genetic algorithm (Holland, 1975), simulated annealing (Kirkpatrick et al., 1983), and the tabu search (Glover, 1986), involves some probabilistic scheme and prevents from getting trapped in a local optimum,

while it needs more computation time than a greedy search. Some existing wrapper approaches include recursive feature elimination (Guyon et al., 2002) combined with the support vector machine (Vapnik, 1998) and the parallel genetic algorithm (Liu et al., 2001) combined with weighted voting (Golub et al., 1999).

In addition to the filter and wrapper approaches introduced above, the third and more advanced approach, the *embedded approach* (Blum and Langley, 1997), can accomplish variable selection and classification simultaneously. The *embedded approach* also interacts with a classification model as the wrapper approach, but it is far less computationally intensive than the wrapper approach (Saeys et al., 2007). Compared with the filter approach, the embedded approach accounts for the correlations between variables better, and thus they are expected to achieve better classification performance. A typical embedded approach can be referred to a classification model with some penalty functions (Ma and Huang, 2008). Among various penalty functions, the  $L_1$ -norm (Lasso) penalty (Tibshirani, 1996) is especially popular because of its sparse estimation. That is, only variables with non-zero estimated coefficients would affect the classifier and constitute the variable subset. Several previous research had showed the satisfactory classification performance of the embedded approach in high-dimensional applications (Segal et al., 2003; Shevade and Keerthi, 2003; Ghosh and Chinnaiyan, 2005).

The diagrams of the filter, wrapper, and embedded approaches as well as the comparisons of their major attributes are presented in Table 1. The filter and embedded approaches will be expanded upon in Chapter 2 and Chapters 3 – 4, respectively.

**Table 1:** The diagrams of the filter, wrapper, and embedded approaches

Approach	Filter approach – univariate	Filter approach – multivariate	Wrapper approach	Embedded approach
Diagram	<p><b>P</b></p>		<p><b>P</b></p>	<p><b>P</b></p>
Interdependence of variables	✗	✓	✓	✓
Interaction of variable selection and classification	✗	✗	✓	✓
Computation complexity	Low	Medium	High	Medium

**P:**  $p$ -dimensional variable set

## CHAPTER II

### SOME EXISTING VARIABLE SELECTION AND CLASSIFICATION METHODS

This chapter focuses on the *filter approaches*. We introduce some commonly used variable ranking techniques in the first section and well-known classification methods in the second section. Then we compare the performance of these variable selection techniques and classification methods through a popular microarray dataset in the third section. In the last section, we make a summary and comments on the filter approaches.

We define following notations for convenience. Let  $x_{ijk}$  be the expression level for the  $j^{\text{th}}$  gene of the  $i^{\text{th}}$  sample within the  $k^{\text{th}}$  group. Suppose there are  $p$  genes and  $n$  samples, of which  $n_k$  samples are from the group  $k$ . For example, in the case of two groups of patients ( $K = 2$ ), for each gene  $j$ ,  $(x_{1,j,1}, x_{2,j,1}, \dots, x_{n_1,j,1})$  denote the  $n_1$  gene expressions from the group 1, and  $(x_{1,j,2}, x_{2,j,2}, \dots, x_{n_2,j,2})$  denote the  $n_2$  gene expressions from the group 2. When emphasis on the gene is unnecessary, the second subscript will be omitted, and the gene expression level denotes as  $x_{ik}$ . Also, let  $\bar{x}_k$  and  $s_k$  denote the mean and the standard deviation of the gene expressions in the  $k^{\text{th}}$  group, respectively.

#### ***2.1 Variable Selection Methods***

In this section, we introduce *filter variable selection* methods. To date, many variable selection methods have been proposed; most of which are based on *variable ranking* techniques. It is recognized that some variable ranking techniques were derived from hypothesis test statistics, which were originally used for comparing gene expression

levels across groups. Based on test statistics or corresponding p-values, genes that are differentially expressed can be identified. The selected genes are then used for classifying samples into groups. Below we describe six commonly used variable selection methods and limit our discussion to the case of two groups (classes) and independent samples.

### 2.1.1 Fold Change

The first and the easiest method to identify differentially expressed genes is the *fold change*. It compares the difference of the expression levels of an individual gene between groups while it essentially assumes a constant variance across all transcripts for the reason that all transcripts go through the same process and therefore have similar variances. A gene is declared  $k$ -fold or greater differentially expressed if  $|\bar{x}_1 - \bar{x}_2| > \log(k)$ . The popularity of the fold change method among practitioners primarily comes from its simplicity for ranking genes. However, from a statistical standpoint, it is considered less valid as an inferential statistic because it does not incorporate the variance and the sample size. This makes a simple rule that eliminates genes with less than two- or three-fold expression changes easily miss biologically important genes that have a small fold change but high statistical significance due to the low variability from replicates (Rosa et al., 2005). Taking variability into account leads to the following T-test.

### 2.1.2 Individual T-test

A basic statistical test for comparing two groups without the equal-variance assumption is the *two-sample Welch's test* (1947). This test statistic is defined as

$$T = \frac{|\bar{x}_1 - \bar{x}_2| - \Delta}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \quad (1)$$

with  $\Delta = 0$  when it tries to detect any differences. The null distribution of  $T$  is approximately a t-distribution with the degrees of freedom  $v$ :

$$v = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{1}{n_1-1}\left(\frac{s_1^2}{n_1}\right)^2 + \frac{1}{n_2-1}\left(\frac{s_2^2}{n_2}\right)^2}.$$

A gene is declared differentially expressed at the level of significance  $\alpha$  if  $T > t_{\alpha/2,v}$ .

In addition to utilizing information from an individual gene, it is possible to borrow information across multiple genes. Tusher et al. (2001) developed the *significance analysis of microarrays (SAM)* statistic by adding a penalty to the sample standard deviation in the denominator of T-statistic (1) to account for a very small standard deviation that results in a large T value. The modified non-parametric T-test is given by

$$T = \frac{|\bar{x}_1 - \bar{x}_2| - \Delta}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2} + s_0}}.$$

Specifically,  $s_0$  is chosen as the percentile of the  $\sqrt{s_1^2/n_1 + s_2^2/n_2}$  values, which makes the coefficient of variation of SAM approximately constant as a function of

$$\sqrt{s_1^2/n_1 + s_2^2/n_2}.$$

### 2.1.3 Multiple Test

In the case of  $p$  statistical tests ( $p$  is usually in thousands) being performed at the significance level  $\alpha$ , if all tests are independent, the probability of at least one false positive (type I error) is  $1 - (1 - \alpha)^p$ , which is very close to unity when  $p$  is large. The expected number of false positives is  $\alpha \times p$ , which is also a large number. Thus, the number of false positives can be so high as to overwhelm and easily obscure actual effects. It is possible to mitigate this problem by a *family-wise multiple test*, such as the Bonferroni multiple test to adjust individual tests. In the Bonferroni multiple test, followed by the individual T-test, a gene is identified differentially expressed if  $T > t_{\alpha/(2p),v}$ .

Note that the Bonferroni multiple test tends to be conservative and may produce

a very large critical value, which makes it difficult to reject null hypotheses, and consequently the adjusted tests yield lower power. In microarray experiments, since the number of genes is very large while the number of samples is limited, the power of a multiple test is likely to be very small. This is clearly undesirable, especially when one needs to make a large number of inferences (Amaratunga and Cabrera, 2004).

#### 2.1.4 False Discovery Rate

Different from the multiple hypothesis tests discussed in the previous section, where the family-wise error rate is controlled, Benjamini and Hochberg (1995) proposed to control the *false discovery rate (FDR)*. The FDR is defined as the expected proportion of the number of false positives (type I error) among the number of rejected null hypotheses. If not every null hypothesis ( $H_0$ ) is true, the FDR method, in fact, maintains some control over the number of false positives, in the sense that the more hypotheses are truly false, the smaller the FDR. Hence, procedures that control the FDR (e.g.  $FDR \leq \alpha$ ) tend to be more powerful than procedures that control the family-wise error rate at the same significance level (Amaratunga and Cabrera, 2004). The FDR is defined as

$$FDR = E \left[ \frac{V}{R} | R > 0 \right] \cdot \Pr(R > 0),$$

where V and R are outlined in Table 2.

**Table 2:** Outcomes from  $p$  hypothesis tests

	Not reject $H_0$	Reject $H_0$	Total
$H_0$ True	$U$ (True negative)	$V$ (False positive; Type I error)	$J_0$
$H_1$ True	$T$ (False negative; Type II error)	$S$ (True positive)	$J_1$
Total	$W$	$R$	$J$

### 2.1.5 Other Univariate Ranking

Fisher, Golub et al. (1999), and Dudoit et al. (2002) developed various univariate ranking criteria. The general idea of these criteria is to select  $d$  genes with the largest ranking scores, where  $d$  is a pre-specified number.

*Fisher criterion score (FCS)* is closely related to the T-test statistic. It is defined as

$$FCS = \frac{(\bar{x}_1 - \bar{x}_2)^2}{s_1^2 + s_2^2}.$$

This criterion is known to require a nearly normal distribution (Duda and Hart, 1973).

Golub et al. (1999) considered a ranking criterion  $GS$  that emphasizes the signal-to-noise ratio for each gene. The  $GS$  criterion is given in (2). Large positive values indicate high expression in group 1 while large negative values indicate high expression in group 2. With ranked  $GS$ 's, an equal number ( $d/2$ ) of genes with most positive and with most negative  $GS$ 's are selected.

$$GS = \frac{\bar{x}_1 - \bar{x}_2}{s_1^2 + s_2^2} \quad (2)$$

Later, Dudoit et al. (2002) selected genes based on the ratio of between-group to within-group sums of squares. For each gene, this ratio is calculated as

$$BW = \frac{\sum_i \sum_k I(y_i = k)(\bar{x}_k - \bar{x})}{\sum_i \sum_k I(y_i = k)(\bar{x}_{ik} - \bar{x}_k)},$$

where  $\bar{x}$  denotes the average expression level of genes across all groups.

### 2.1.6 Correlation-based Ranking

Hall (1999) proposed the *correlation-based variable selection* method. It is a multivariate filter variable selection technique that uses a search algorithm along with a function to evaluate each variable subset. The logic behind this technique is that a good variable subset should contain variables that are highly correlated with the class but uncorrelated with each other. The correlation-based ranking score is defined as

$$H = \frac{d\bar{r}_{cf}}{\sqrt{d + d(d-1)\bar{r}_{ff}}},$$

where  $H$  is the score of a variable subset containing  $d$  variables,  $\bar{r}_{cf}$  is the average correlation between  $d$  variables, and the class  $\bar{r}_{ff}$  is the average pair-wise correlation between  $d$  variables. The numerator of  $H$  represents how predictive of the class a group of variables is while the denominator represents how much redundancy exists in a variable subset. More details can be found in Hall's dissertation (1999).

## 2.2 Classification Methods

The main use of classification methods is to derive effective classification rules (i.e., classifiers) with the data in training sets. The classifiers are then applied to an independent dataset that is usually referred to as a testing set to evaluate the performance of classifiers. Various classification methods differ in the assumptions regarding the structure and the distribution of the data, the form of the classification rules, and the availability of prior information (Lee, 2004). Below we introduce four well-known classification methods.

### 2.2.1 Linear Discriminant Analysis

Fisher (1936) proposed a method, *linear discriminant analysis (LDA)*, that finds the linear projections of the data that most effectively separates out the  $k$  classes. In the case of two classes, classification can be based on the projection  $w'x$ : the projection is made in the direction  $w$ , where the classes are most widely separated in the training set. Let  $n_s$ , the generic  $G$ -vector  $\bar{x}_s$ , and the  $G \times G$  matrix  $S_s$  denote the sample size, the mean, and the variance-covariance matrix of the  $s^{th}$  class in the training set, respectively. Also, let  $S = [(n_1 - 1)S_1 + (n_2 - 1)S_2]/(n_1 + n_2 - 2)$  denote the pooled variance-covariance matrix. A standardized measure of separation between the two samples in the training set in the direction  $w$  can be written as

$$\lambda = \frac{(w'\bar{x}_1 - w'\bar{x}_2)^2}{w'Sw},$$

which equals to the squared distance between the linear combinations of means divided by the variance of the linear combination. The direction  $w$  that maximizes  $\lambda$  is given by

$$w = S^-(\bar{x}_1 - \bar{x}_2),$$

where  $S^-$  denotes the generalized inverse of  $S$  as  $S$  is usually singular in microarray data.

The classification rule is based on the linear classifier:

$$w'x = (\bar{x}_1 - \bar{x}_2)'S^-x.$$

If  $w'x > w'(\bar{x}_1 + \bar{x}_2)/2$ , then  $x$  is classified as in class 1; otherwise,  $x$  is classified as in class 2. Some extensions of the linear discriminant analysis can be referred to Hastie et al. (2008); Amaratunga and Cabrera (2004).

### 2.2.2 Logistic Regression

The *logistic regression* model arises from the desire to model the posterior probabilities of the  $K$  classes via linear functions in  $x$ . The model is specified in terms of  $K-1$  log-odds (Hastie et al., 2008). This model is widely used in biostatistical applications, where binary responses occur quite frequently.

Let  $y_i \in \{0, 1\}$  be the binary outcome for subject  $i$ ,  $i = 1, \dots, n$  and  $\mathbf{x}_i$  be a  $p \times 1$  vector of predictors (genes). The generic logistic regression model has the form

$$\log \frac{\Pr(y_i = 1|\mathbf{x}_i)}{\Pr(y_i = 0|\mathbf{x}_i)} = \beta_0 + \mathbf{x}_i^T \boldsymbol{\beta},$$

where  $\beta_0$  and  $\boldsymbol{\beta}$  are unknown parameters. The maximum likelihood estimates of  $\beta_0$  and  $\boldsymbol{\beta}$  are obtained by minimizing the negative log-likelihood function

$$\begin{aligned} l(\beta_0, \boldsymbol{\beta}) &= -\sum_{i=1}^n [y_i \log \pi_i + (1 - y_i) \log(1 - \pi_i)] \\ &= -\sum_{i=1}^n [y_i(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta}) + \log(1 + \exp(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta}))], \end{aligned}$$

where  $\pi_i$  is the probability of observing  $y_i = 1$ .

Logistic regression offers the advantage of simultaneously estimating the probabilities  $\pi_i$  and  $1-\pi_i$  for each class and classifying subjects. The probabilities of classifying the  $i^{\text{th}}$  sample in class 1 is estimated by  $\hat{\pi}_i(\mathbf{x}) = \frac{\exp(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta})}{1 + \exp(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta})}$ . The predicted class is then obtained by  $I\{\hat{\pi}_i(\mathbf{x}) > \frac{1}{2}\}$ , where  $I(\cdot)$  is an indicator function. However, for high-dimensional applications with the dimension  $p + 1$  greater than the samples size  $n$ , the second-derivatives of  $l(\beta_0, \boldsymbol{\beta})$  are not of full rank. In this case, logistic regression fails to produce reliable estimation and classification results. More discussion can be found in McCullagh and Nelder (1998).

### 2.2.3 Classification Tree

A *classification tree* features a visual display of recursive partitioning, which generates partitions from the training samples with the goal of achieving a partition that generates a good prediction rule. One of the nice tree properties is that trees resemble decision rules in an easy to understand way compared to most of other methods (Amaratunga and Cabrera, 2004).

For a binary tree, it begins at a root node where data are split into two buckets using one of the classification variables from the set. One of the commonly used node-splitting criteria is the *deviance*, which is defined as  $l \min(l_L, r_L) + r \min(l_R, r_R)$ , where  $l$  and  $r$  are the proportions of observations going to the left and right buckets;  $l_L$  ( $l_R$ ) and  $r_L$  ( $r_R$ ) are the proportions of class 1's and class 2's in the left-side (right-side) bucket, respectively. In order to prevent an overfitted tree with small buckets at terminal nodes or an oversized tree that is hard to interpret, a cross-validation method is usually involved when a tree is constructed.

### 2.2.4 $k$ -Nearest Neighbor

The *k-nearest neighbor* method does not build a classifier on the training data as do the foregoing methods. Instead, when a testing subject arrives, it searches for the  $k$  neighboring points closest to the testing subject and uses their labels to label the

new subject (Amaratunga and Cabrera, 2004). Let  $x_i$  represent the  $i^{th}$  sample, and  $y_i$  gives the class number of the  $i^{th}$  sample. Also, let  $x$  be the candidate sample for classification and  $S_{k,x}$  be the set of the  $k$ -nearest neighbors of  $x$  in the training set. The simple  $k$ -nearest neighbor ( $k$ NN, for short) method consists of estimating the probability that  $x$  belongs to the  $i^{th}$  class  $p(l|x)$  by the proportion of the  $k$ -nearest neighbors that belong to the  $i^{th}$  class:

$$\hat{p}(l|x) = \frac{\#\{g_i = l \mid x_i \in S_{k,x}\}}{k}.$$

The classification rule is based on a majority vote. That is,  $x$  is assigned to the  $i^{th}$  class if  $l$  maximizes the probability  $\hat{p}(l|x)$ . The  $k$  value is usually chosen by cross-validation with the training set. The one with the smallest cross-validation error is then selected and applied to the testing set.

### ***2.3 Example: Leukemia Data***

In this section, we use an example to demonstrate how well filter approaches perform in the classification problem of microarray experiments. Among numerous filter variable selection and classification methods introduced in the previous two sections, we chose three typical methods for each and compared their performance in this study with a popular gene expression dataset, published by Golub et al. (1999). They monitored gene expression on Affymetrix high-density oligonucleotide DNA microarrays that contains 7129 probes to classify human acute leukemias. The initial leukemia dataset consisted of 38 bone marrow samples, including 27 acute lymphoblastic leukemia (ALL) and 11 acute myeloid leukemia (AML), from acute leukemia patients. This initial dataset was used to create a class predictor, and then an independent collection of 34 testing samples (20 ALL and 14 AML) was used to assess the validity of the class predictors.

Since there are too many genes compared to samples in the initial dataset (i.e.,

training dataset), we apply three popular gene ranking methods here, namely, the T-test, the Bonferroni test, and the false discovery rate. The number of selected genes with different significance levels  $\alpha$  is summarized in Table 3. The individual T-test tends to identify much more genes than the other two methods, while multiplicity adjustments are so strong that they identify fewer differentially expressed genes. The false discovery rate is less conservative and becomes an intermedium between the individual T-test and the Bonferroni multiple test. In this comparison, FDR seems to be a favorable way to identify genes that are differentially expressed across ALL and AML patients. However, it is worthy to note that large numbers of selected genes may still be problematic for classification.

**Table 3:** Comparison of selected genes by T-test, Bonferroni test, and FDR

Significance level $\alpha$	Number of selected genes		
	T-test	Bonferroni Test	FDR
0.2	3174	43	3025
0.1	2325	<b>26</b>	2091
0.05	1694	16	1395
0.01	806	7	501
0.005	606	5	299
0.001	305	1	65
0.0005	226	1	<b>25</b>
0.0001	101	0	2
0.00005	77	0	2
0.00001	<b>31</b>	0	0
0.000005	22	0	0

To compare different classification methods, we fixed the number of genes when constructing a classifier. Two doable numbers of top genes,  $q = 20$  and  $q = 30$ , were evaluated, and the rankings of genes were based on two criteria, |T-test| and FDR. Three classification methods, linear discriminant analysis (LDA), logistic regression (LR), and  $k$ -nearest neighbors ( $k$ NN) were studied in which classifiers were built with all pre-selected genes without further model selection. For the  $k$ NN method, the

number of  $k$ -nearest neighbors was chosen between 1 to 5 by cross validation with the training dataset. The classification performance was then measured by the number of misclassified samples in the testing dataset.

Table 4 shows the comparison of three classification methods with different numbers of variables selected via two variable selection methods. The numbers shown in the last three columns are the number of misclassified testing samples. Below are our findings. First, with both ranking criteria, increasing the number of selected genes ( $q$ ) beyond 20 does not help to reduce classification errors. This implies that for the diagnostic purposes, *small* subsets of genes are sufficiently informative to distinguish acute leukemia subtypes. Second, in terms of the classification performance with the same number of pre-selected genes, FDR is preferable to the T-test in identifying differentially expressed genes. Third,  $k$ NN is outperformed among the three classification methods in this example. These findings, however, may need more investigations on different datasets for more supports.

**Table 4:** Comparison of classification methods with different ranking criteria

Ranking	$q$	LDA	LR	$k$ NN
T-test	30	18	13	3
	20	12	12	2
FDR	30	7	11	3
	20	8	5	2

## 2.4 Summary

In the previous section, we examined the performance of filter approaches through a real example. Notwithstanding the filter variable selection techniques are easily applied and commonly used as baseline methods for classification, filter approaches have some overall drawbacks: (i) A threshold of selecting variables is an essential ingredient for classification. That is, the accuracy of prediction heavily relies on the pre-selected

variables; (ii) Variable selection is independent of classification such that the pre-selected variable subset, no matter how significant in differentiating groups, may not be optimal for classification; (iii) For most of classification methods, the number of the pre-selected variables is not allowed to exceed the number of samples; otherwise traditional classification methods are still not applicable to microarray experiments; and (iv) Most filter variable selection approaches do not consider correlated variables. On the grounds of the above drawbacks of filter approaches and the features of microarray data, we comment that filter approaches are not considered as desirable as others that have capability against (i) – (iv). As the comparison we made in Section 1.3, both the wrapper and embedded approaches are desirable while the latter is relatively more efficient in computation. Thus, our attention will be directed to the *embedded approach* in the next two chapters.

## CHAPTER III

# A NEW CLASSIFICATION APPROACH: ITERATIVE RESELECTION PENALIZED LOGISTIC REGRESSION

In this chapter, we focus on the embedded approach and propose a new algorithm that not only performs variable selection and classification simultaneously but also improves the consistency of variable selection in penalized logistic regression. We start this chapter with the literature review of recent embedded approaches that have been applied to microarray studies. Then we elaborate the motivation and the features of the new algorithm in Section 3.2 and develop the whole algorithm in Section 3.3. The performance of the proposed algorithm is evaluated and compared through four real-world microarray studies in Section 3.4. A simulation study is also carried out to evaluate its finite sample performance in Section 3.5. Some discussion and extended work are remarked in the last section of this chapter.

### *3.1 Review of Embedded Approaches*

The general introduction of *embedded approaches* can be referred to Section 1.3. Here, we concentrate on recent work that adopted embedded approaches to microarray classification problems. Table 5 summarizes recent related work in this field. Among them, *regression models with the Lasso penalty* are indicated as one of the most popular embedded approaches (Roth, 2002; Shevade and Keerthi, 2003; Segal et al., 2003; Wu, 2006; Huang et al., 2008). Although some studies adopted linear regression as an alternative to logistic regression for classification problems, logistic regression was more common and also proved to excel linear regression for binary responses

(Press and Wilson, 1978).

**Table 5:** Recent related work using embedded approaches for microarray data

Author	Classification Method
Roth (2002)	Logistic regression with lasso penalty
Shevade et al. (2003)	Logistic regression with lasso penalty
Segal et al. (2003)	Linear regression with lasso penalty
Ghosh (2005)	Linear discriminant analysis with lasso penalty
Wu (2006)	Linear regression with lasso penalty
Pan et al. (2006)	Mixture model with adaptive lasso penalty
Liu et al. (2007)	Logistic regression with bridge / elastic net penalty
Huang et al. (2008)	Linear regression with lasso / adaptive lasso penalty
Guo et al. (2008)	Shrunken centroids regularized discriminant analysis

The Lasso penalty (Tibshirani, 1996) is a popular regularization technique, which imposes a  $L_1$ -norm penalty on regression coefficients. An important feature of the Lasso penalty is that it shrinks all regression coefficients towards zero and many of which are exactly set to zero. The sparse solutions, thus, can be used for variable selection. Since penalized logistic regression with the Lasso penalty (PLRL<sub>1</sub>) is able to perform variable selection and classification (estimation) simultaneously even for high-dimensional binary data, it has been widely and successfully used in practice.

### ***3.2 Motivation and Features of New Approach***

From the previous review, we noted that the Lasso penalty is widely used in practice because it can lead to sparse estimation. On the top of this, to serve as an embedded approach, it is necessary to examine its performance in variable selection, in particular the ability of identifying a true model. Some literature (Leng et al., 2006; Meinshausen and Bühlmann, 2006; Zhao and Yu, 2006) studied the consistency of variable selection. Here, consistency refers to the correct selection of non-zero coefficients (i.e., the true model) with probability converging to one. Zhao and Yu (2006)

provided some regularity conditions of variable selection consistency for Lasso under linear models when the number of variables ( $p$ ) is larger than the number of samples ( $n$ ). One important conclusion from their paper is that Lasso can be consistent in variable selection when  $p$  grows with  $n$  not faster than exponentially. Although there is a lack in developing the regularity conditions of variable selection consistency under generalized linear models, we conjecture that the performance of variable selection would also be affected by the relative size between  $p$  and  $n$ , as the case of linear models. This expectation motivates us to reduce the size of a variable set used in PLRL<sub>1</sub> to some extent to achieve better variable selection consistency.

The above motivation, however, raises a further question: How to search the space of all possible variable subsets with size  $c$ , where  $c$  is much less than  $p$  but larger than  $n$ ? A basic approach is through an exhaustive search, but it becomes computationally intractable for large  $p$  because the number of possible subsets ( $C_c^p$ ) is huge. In such situation, other procedures, for example, forward selection and backward elimination, would be more applicable; however, these selection procedures are at risk of being trapped in a local optimum. In fact, searching the best variable subset is recognized as an *NP-hard combinatorial optimization problem*. To solve this type of problem, a *randomized heuristic search* is commonly suggested, especially when the number of combinations grows exponentially with the number of variables (Lundy, 1985; Murty, 1995; Saeys, 2007). Popular heuristic optimization algorithms for searching a global optimum include the genetic algorithm (Holland, 1975), simulated annealing (Kirkpatrick et al., 1983), and the tabu search (Glover, 1986). In this study, a simulated annealing (SA) algorithm is utilized for its simplicity to iteratively search the space of all possible variable subsets for PLRL<sub>1</sub>. With the controlled size of variable sets used in PLRL<sub>1</sub>, a heuristic-based *iterative reselection penalized logistic regression (IRPLRL<sub>1</sub>)* is then developed for binary class prediction in this study.

According to the taxonomy of the variable selection and classification approaches

(as discussed in Section 1.3), the proposed IRPLRL<sub>1</sub> can be regarded as an iteratively embedded approach, which has three main features: (i) It performs variable selection and classification simultaneously; (ii) The consistency of variable selection with the Lasso penalty is expected to be improved by reducing the number of variables used in PLRL<sub>1</sub>; and (iii) It has the ability to efficiently find the best variable subset for classification. Generally speaking, IRPLRL<sub>1</sub> shares most advantages of embedded approaches, except that it requires more computational efforts than a non-iterative PLRL<sub>1</sub>. The details will be discussed in the next section.

### 3.3 Iterative Reselection Penalized Logistic Regression

The new classification approach is developed based on the standard penalized logistic regression (PLR) model. In this section, we first review the PLR model and then introduce the proposed new classification approach.

#### 3.3.1 Penalized Logistic Regression

Let  $y_i \in \{0, 1\}$  be the binary outcome for subject  $i$ ,  $i = 1, \dots, n$  and  $\mathbf{x}_i$  be a  $p \times 1$  vector of predictors (genes). The generic logistic regression model has the form

$$\log \frac{\Pr(y_i = 1 | \mathbf{x}_i)}{\Pr(y_i = 0 | \mathbf{x}_i)} = \beta_0 + \mathbf{x}_i^T \boldsymbol{\beta}, \quad (3)$$

where  $\beta_0$  and  $\boldsymbol{\beta}$  are unknown parameters. The maximum likelihood estimates of  $\beta_0$  and  $\boldsymbol{\beta}$  can be obtained by minimizing the negative log-likelihood function

$$\begin{aligned} l(\beta_0, \boldsymbol{\beta}) &= - \sum_{i=1}^n [y_i \log \pi_i + (1 - y_i) \log(1 - \pi_i)] \\ &= - \sum_{i=1}^n [y_i (\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta}) - \log(1 + \exp(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta}))], \end{aligned} \quad (4)$$

where  $\pi_i$  is the probability of observing  $y_i = 1$ .

Logistic regression offers the advantage of simultaneously estimating the probabilities  $\pi_i$  and  $1 - \pi_i$  for each class and classifying subjects. The probabilities of classifying the  $i^{th}$  sample in class 1 is estimated by  $\hat{\pi}_i(\mathbf{x}) = \frac{\exp(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta})}{1 + \exp(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta})}$ . The predicted class is then obtained by  $I\{\hat{\pi}_i(\mathbf{x}) > \frac{1}{2}\}$ , where  $I(\cdot)$  is an indicator function.

Penalized logistic regression (PLR) adds a nonnegative penalty function to (4) such that the size of coefficients in high-dimension can be controlled. Various penalty functions have been discussed in the literature; the details can be referred to Fan and Li (2001), Zou and Hastie (2005), Zou (2006), and Friedman et al. (2007). The  $L_1$ -norm penalty, proposed by Donoho and Johnstone (1994) and Tibshirani (1996), is one of the popular penalty functions. The  $L_1$ -norm penalty performs variable selection and estimation simultaneously by constraining the sum of the absolute values of coefficients, i.e.,  $\sum_{j=1}^p |\beta_j| \leq t$ , where the bound  $t$  is a user-specified parameter and often chosen by a model selection procedure. This constraint is equivalent to minimizing the negative log-likelihood function plus an  $L_1$ -norm penalty on the coefficients, which can be written in Lagrange's form as

$$\min_{\beta_0, \boldsymbol{\beta}} l_1(\beta_0, \boldsymbol{\beta}) = - \sum_{i=1}^n \{y_i(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta}) - \log[1 + \exp(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta})]\} + \lambda \sum_{j=1}^p |\beta_j|. \quad (5)$$

The nonnegative shrinkage parameter  $\lambda$  in (5) needs to be determined when a PLR model is applied. In practice, it is often chosen by a cross validation procedure. If there are more than one  $\lambda$  giving the smallest cross validation error, we prefer to choose the largest  $\lambda$  among them such that the number of selected genes is the smallest.

To solve a penalized logistic regression model, the traditional numerical methods are through maximum likelihood estimation or the Newton-Raphson algorithm. However, the computation of these methods is prohibitive when the number of variables is large (Zhu and Hastie, 2004). In this study, we adopt the coordinate descent algorithm, recently developed by Friedman et al. (2010), to solve PLRL<sub>1</sub> for  $p \gg n$  problems. The coordinate descent algorithm is favorable for its simplicity, efficiency, and stability (Wu and Lange, 2008). The idea of the coordinate descent algorithm is to solve the problem along a regularization path for each value of coefficients, using the current estimates as warm starts. Let  $x_{ij}$  indicate the  $j^{\text{th}}$  gene expression of the  $i^{\text{th}}$  sample. For simplicity, we assume that  $x_{ij}$  are standardized such that  $\frac{1}{n} \sum_{i=1}^n x_{ij} = 0$

and  $\frac{1}{n} \sum_{i=1}^n x_{ij}^2 = 1$ , for  $j = 1, \dots, p$ . It is well-known that the Newton-Raphson algorithm for minimizing (4) amounts to iteratively reweighted least squares (IRLS). Therefore, the coordinate-wise updates can be obtained by minimizing the quadratic approximation to the negative log-likelihood on current estimates  $(\tilde{\beta}_0, \tilde{\boldsymbol{\beta}})$ :

$$\beta_j = \frac{T(\sum_{i=1}^n w_i x_{ij} (v_i - \tilde{\beta}_0 - \mathbf{x}_i^T \tilde{\boldsymbol{\beta}}), \lambda n)_+}{\sum_{i=1}^n w_i x_{ij}^2}, \quad (6)$$

where

$$v_i = \tilde{\beta}_0 + \mathbf{x}_i^T \tilde{\boldsymbol{\beta}} + \frac{y_i - \tilde{\pi}_i}{\tilde{\pi}_i(1 - \tilde{\pi}_i)}$$

is the working response,  $w_i = \tilde{\pi}_i(1 - \tilde{\pi}_i)$  is the weight,  $\tilde{\pi}_i$  is evaluated at the current parameters  $(\tilde{\beta}_0, \tilde{\boldsymbol{\beta}})$ , and  $T(a, b)$  is the soft-thresholding operator with the value

$$T(a, b) \equiv \text{sign}(a)(|a| - b)_+ = \begin{cases} a - b & \text{if } a > 0 \text{ and } b < |a|, \\ a + b & \text{if } a < 0 \text{ and } b < |a|, \\ 0 & \text{if } b \geq |a|. \end{cases}$$

As a result, the  $L_1$ -norm (Lasso) penalty can achieve variable selection as variables with zero coefficients are effectively omitted from the model. In the aspect of computation, the coordinate descent algorithm is generally competitive with the well-known LARS algorithm (Efron et al., 2004) and others in the context of large Lasso-linear and Lasso-logistic regression models (Friedman et al., 2007; 2008).

### 3.3.2 Iterative Reselection Algorithm

An important component of the proposed classification approach is the *iterative reselection* algorithm. The general design concept of this algorithm is from the principle behind *simulated annealing (SA)*, which was first proposed by Kirkpatrick et al. (1983). SA is a type of local search heuristic involving some random elements in the process. SA avoids getting trapped in a local optimum by accepting a feasible but unfavorable solution with some probability. This makes SA possible to move away from a local optimum and explore the feasible region in its entirety to find the

global optimum. In the proposed classification approach, SA provides iterative improvements on classification through three major activities: (i) regenerating variable subsets in a controlled size by partial selection, (ii) updating active variables with non-zero coefficients identified by PLRL<sub>1</sub>, and (iii) accepting new active variables in a probabilistic scheme as the standard way in SA. The purpose of (i) is to allow for more variable selection consistency in (ii) and to take advantage of the last iteration, while (iii) avoids a propensity to stick at local optimal variable subsets (i.e., attempts to achieve the global optimal gene subset). In theory, the SA algorithm should continue until the best solution is found; however, in practice, other stopping criteria are usually applied (Aarts and Lenstra, 1997). For example, the value of the objective function stays unchanged for a large number of consecutive iterations, or the algorithm reaches the maximal number of iterations. Below we describe the proposed iterative reselection algorithm step by step with the flowchart outlined in Figure 2.

Step 0: As all regression coefficients will be penalized with a global shrinkage parameter  $\lambda$  in PLRL<sub>1</sub>, predictors need to be standardized in the pre-process step such that the expression intensity of each gene across patients is centered to zero and has variance of one.

Step 1: Set  $m = 0$  and  $g = 0$ . Estimate  $p$  marginal least squares coefficients (i.e., apply individual T-tests for each gene) based on a training set and rank genes by the absolute values of the coefficients.

Step 2: Set  $m = 1$ . Fit a PLRL<sub>1</sub> model using a training set with top  $c$  genes ( $c < p$ ) from the ordered gene list. The shrinkage parameter  $\lambda$  is chosen by  $k$ -fold cross validation. Identify the active set  $A = \{j : \hat{\beta}_j \neq 0\}$  and calculate the cross validation error  $cv_0$  based on  $A$ . Set  $A^* \leftarrow A$ ,  $cv^* \leftarrow cv_0$ , and  $cv \leftarrow cv_0$ . ( $A^*$  and  $cv^*$  stand for the best-so-far active set and cross validation error.)

Step 3: Set  $m = m + 1$ . Randomly choose  $c - |A^*|$  genes from  $p - |A^*|$  genes that are not

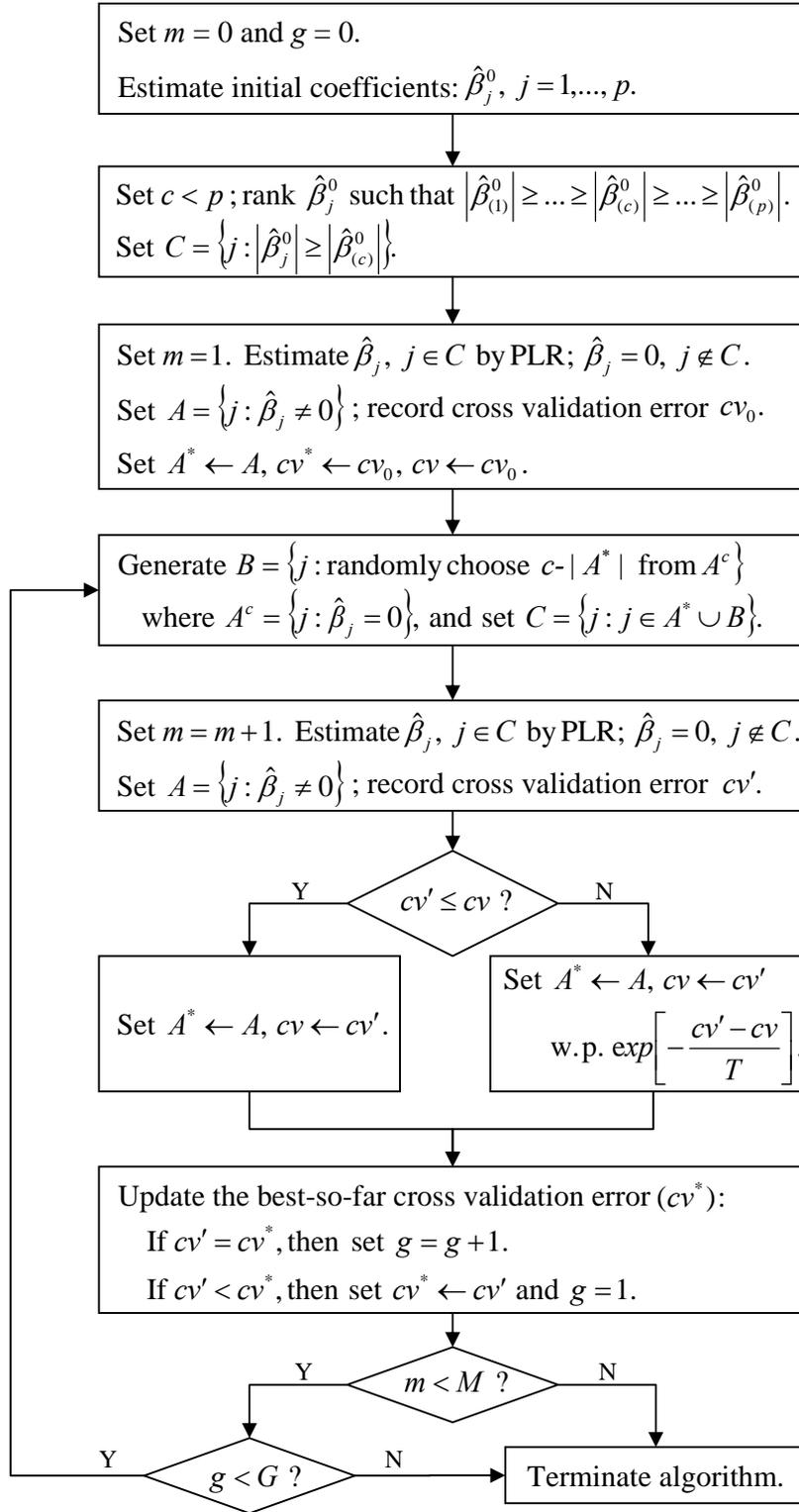
in the active set  $A^*$  to form a new candidate subset  $C$  that satisfies  $|C| = c$ , where  $|\cdot|$  is the cardinality of  $\cdot$ . Fit a PLRL<sub>1</sub> model based on  $C$ . Update  $A$  and calculate the cross validation error  $cv'$  as in Step 1.

Step 4: Update  $A^*$ ,  $cv$ , and  $cv^*$ . If  $cv' \leq cv$ , set  $A^* \leftarrow A$  and  $cv \leftarrow cv'$ . If  $cv' > cv$ , set  $A^* \leftarrow A$  and  $cv \leftarrow cv'$  with probability  $\exp[-(cv' - cv)/T]$ , where  $T$  is a preset parameter known as “temperature” in simulated annealing. If  $cv' < cv^*$ , set  $cv^* \leftarrow cv'$  and  $g = 1$ ; if  $cv' = cv^*$ , set  $g = g + 1$ .

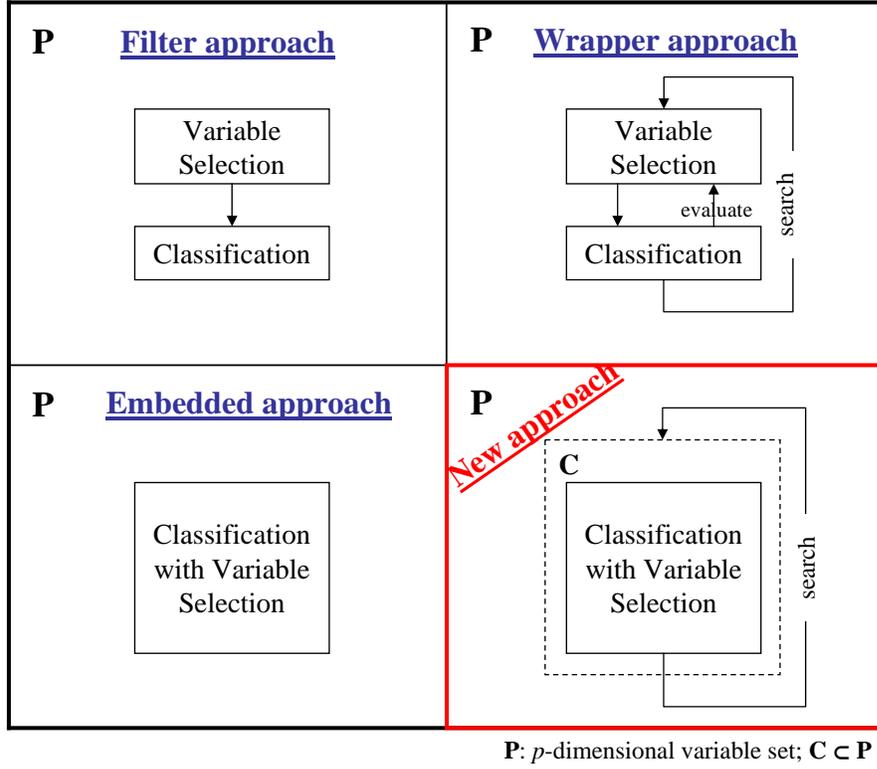
Step 5: Return to Step 3 until  $cv^*$  has not changed for a given number of times (i.e.,  $g = G$ ) or this algorithm reaches a predetermined number of iterations (i.e.,  $m = M$ ). The final classification is based on  $A^*$ .

Figure 3 presents the diagrams of IRPLRL<sub>1</sub> with three existing classification approaches. The proposed IRPLRL<sub>1</sub> can be regarded as an iteratively embedded approach. Note that this iterative reselection algorithm is not limited to either the number of variables or the penalty function used in logistic regression. The Lasso penalty can also be replaced with other penalty functions, such as adaptive Lasso (Zou, 2006), elastic net (Zou and Hastie, 2005), or variants.

Numerous researchers attempted to examine the convergence and finite-time behavior of the SA algorithm by assuming a mathematical model, of which the most popular one is a Markov chain since the next state depends only on the current state. The convergence of SA was investigated in the mid 80s by a number of research, including Gidas (1985), Lundy and Mees (1986), Mitra et al. (1986), and Hajek (1988). They showed that SA can converge in the limit to the globally optimal solution with probability one for an irreducible and aperiodic Markov chain. The irreducible Markov chain is one in which all states are reachable from all other states. From a theoretical point of view, this conclusion is important and useful since it provides an explanation for why SA works in practice. Thereafter, Rajasekaran (2000) studied



**Figure 2:** Iterative reselection algorithm



**Figure 3:** The diagrams of the new approach with three existing approaches

the worst-case convergence time of SA from a computational point of view. Assuming that each state not in the current solution is equally likely to be generated next, the expected number of iterations before the global optimal solution being visited is no more than  $[d \times \exp(\Delta/T)]^D$ , where  $T$  is the minimal temperature that SA ever went through,  $\Delta$  is the maximal difference of the objective function, and  $d$  and  $D$  are the degree and the diameter of the underlying Markov chain. This result holds regardless of the initial solutions and the annealing schedules. This implies that even if the temperature  $T$  is assumed to be constant throughout the process, as long as enough time is given, SA may still converge. The proposed iterative reselection algorithm essentially does not violate the above conditions of convergence (i.e. irreducibility and aperiodicity) because all possible subsets are reachable from all of the other subsets.

### 3.4 Performance Assessment

The proposed iterative reselection algorithm was implemented in MATLAB 7.6 and evaluated by three well-known, publicly available microarray datasets as well as one new dataset. For illustration and assessment, following parameters were set in the iterative reselection algorithm:  $c = 300$ ,  $k = 5$ ,  $T = 0.03/\log 2$ ,  $G = 5$ , and  $M = 1000$ , where  $T$  is corresponding to a tolerance of 50% in probability toward accepting an unfavorable subset with a 3% larger  $cv$  error.

#### 3.4.1 Estrogen and Lymph Data

The estrogen and lymph datasets were first presented by West et al. (2001), in which two statuses, the estrogen receptor (ER) and the lymph node (LN), of breast tumor samples were studied. With Affymetrix gene chip technology, 7129 genes on 49 breast tumor samples were obtained. These datasets can be downloaded from <http://data.cgt.duke.edu/west.php>.

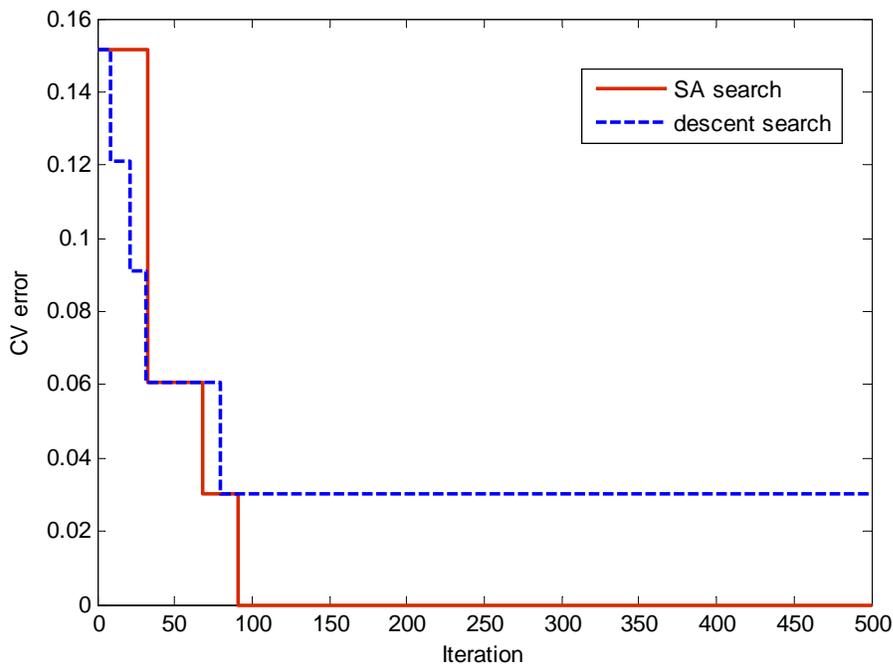
First, we analyzed the estrogen receptor status of 25 ER+ and 24 ER- tumors. Since an independent testing dataset was not available, we randomly chose two-thirds of the samples as a training set and kept the remaining one-third of the samples as a testing set. In each training set, 5-fold cross validation was used to select the shrinkage parameter  $\lambda$  with the smallest  $cv$  error, which was then used to establish a classifier with the training set. Then the classification performance was evaluated based on the testing set. The above procedure was repeated 50 times to obtain the average misclassification rate. The proposed IRPLRL<sub>1</sub> identified an average of 40 active genes in establishing classifiers and yielded an average testing error as 9.38% (= 1.5/16). For comparison, we applied the identical 50 training-testing splits to some existing methods. Table 6 summarizes the classification performance of various methods. It is interesting to observe that the original PLRL<sub>1</sub> also identified an average of 40 active genes, but it generated a slightly larger average testing error (2.0/16) than IRPLR

did. In addition, we included a filter approach, the T-test combined with logistic regression (LR), and the other two recently-developed embedded approaches: pseudo logistic regression (PsLR; Zhang et al., 2007) and shrunken centroids regularized discriminant analysis (SCRDA; Guo et al., 2008). The listed testing error of T-test + LR is the smallest error that can be achieved. In PsLR, all genes were utilized in building the classifiers, which produced an average testing error of 14.375% (= 2.3/16). This result is similar to the average testing error (14.6%) obtained from 100 random splits by Zhang et al. (2007). Overall, IRPLRL<sub>1</sub> is favorable because of its higher classification accuracy and the modest gene subsets in the analysis of the estrogen receptor status.

**Table 6:** Comparison of classification methods for breast tumors based on ER status

Method	Number of genes	Testing error
T-test + LR	15	4.6/16
PLRL <sub>1</sub>	40	2.0/16
<b>IRPLRL<sub>1</sub></b>	<b>40</b>	<b>1.5/16</b>
SCRDA	50	2.3/16
PsLR (LR + SVM)	7129	2.3/16

To demonstrate the advantage of the SA algorithm in IRPLRL<sub>1</sub>, we investigated its convergence and compared with a naive descent search (NDS) algorithm. The NDS algorithm accepts a new variable subset only when the *cv* error is smaller than the current one; otherwise, the current variable subset would be retained (Murty, 1995). As the NDS algorithm is easily trapped in local optima, its performance highly depend on the initial settings. Using the estrogen receptor dataset, we ran 500 iterations in which all the parameters were set the same in both search methods. Figure 4 exhibits that the NDS algorithm got trapped in a local optimum with a 3% error after 80 iterations while SA further decreased *cv* errors to 0 within 100 iterations. These results show that SA plays an important role in searching variable subsets so that IRPLRL<sub>1</sub> can achieve better classification results.



**Figure 4:** Comparison of convergence in SA and a descent search.

The second example concerns an important clinical issue of metastatic tumor spread. In the dataset of lymph, 25 and 24 tumor samples were reported as LN+ and LN-, respectively. (Note that the tumor samples with LN+ are not identical to the samples with ER+.) Using the same procedure above for the ER status, we analyzed the LN status and compared the classification results from different methods. Although the classes of the LN were less well separated (i.e., have higher prediction error) than the classes of the ER, numerical results continued to show that IRPLRL<sub>1</sub> produced lower classification errors than the other methods, as shown in Table 7.

**Table 7:** Comparison of classification methods for breast tumors based on LN status

Method	Number of genes	Testing error
T-test + LR	16	6.0/16
PLRL <sub>1</sub>	45	3.9/16
<b>IRPLRL<sub>1</sub></b>	<b>63</b>	<b>3.7/16</b>
SCRDA	68	6.3/16
PsLR (LR + SVM)	7129	6.2/16

In the aspect of computational efficiency, the average number of iterations in  $\text{IRPLRL}_1$  was 101 for the ER status and 357 for the LN status. The computation time of 100 iterations was about 50 seconds on the computer equipped with the Intel Core2 Duo 2.66 GHz CPU and 3GB RAM.

### 3.4.2 Leukemia Data

Golub et al. (1999) published the leukemia dataset, in which two classes of acute leukemias were studied: acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML). The gene expression intensities of 7129 probes on human genes were obtained from Affymetrix high-density oligonucleotide microarrays. In this experiment, a training set of 38 patients (27 ALL and 11 AML) and an independent testing set of 34 patients (20 ALL and 14 AML) are available at [http://www.broadinstitute.org/cgi-bin/cancer/publications/pub\\_paper.cgi?mode=view&paper\\_id=43](http://www.broadinstitute.org/cgi-bin/cancer/publications/pub_paper.cgi?mode=view&paper_id=43).

We built a classifier based on the training set and then evaluated its performance using the independent testing set. The results show that the proposed  $\text{IRPLRL}_1$  algorithm converged after 111 iterations. The achieved minimal *cv* error was 1/38, with which the smallest number of active genes was 11. The detailed descriptions of active genes are listed in Table 8. Based on these 11 genes,  $\text{IRPLRL}_1$  yielded one misclassified AML testing sample, which was also misclassified by Golub et al. (1999) using a weighted voting scheme on 50 genes. This dataset was also analyzed by other researchers using various classification methods. We list the results of some popular in Table 9, in which the proposed  $\text{IRPLRL}_1$  was compared with six filter approaches, three wrapper approaches, and three embedded approaches. In general, the filter approaches were not as good as the wrapper and embedded approaches, except when FDR was utilized for variable selection. From the study by Zhu and Hastie (2004), SVM used more genes than  $\text{PLRL}_2$  in building a classifier. Of these compared methods, five of them plus  $\text{IRPLRL}_1$  produced the best performance of

only one misclassified testing sample. Among them, IRPLRL<sub>1</sub> appeared to be the most parsimonious for it utilized the smallest number of genes in building classifiers and reached the same classification error. In this analysis, although we are able to classify two subtypes of acute leukemias well based on a smaller number of genes, the biological meanings of the identified genes may need further investigation.

**Table 8:** Selected genes for classifying leukemia data

Gene No.	Description
461	Liver mRNA for interferon-gamma inducing factor(IGIF)
1745	LYN V-yes-1 Yamaguchi sarcoma viral related oncogene homolog
1834	CD33 antigen (differentiation antigen)
2020	FAH Fumarylacetoacetate
2242	Peptidyl-prolyl cis-trans isomerase, mitochondrial precursor
2402	Azurocidin gene
3320	Leukotriene C4 synthase (LTC4S) gene
4847	Zyxin
5039	LEPR Leptin receptor
6041	APLP2 Amyloid beta (A4) precursor-like protein 2
6378	NF-IL6-beta protein mRNA

To better distinguish the performance of the proposed approach from others, we combined the original training and testing sets and randomly split the entire leukemia dataset into 38 training samples and 34 testing samples 50 times. The average number of selected genes and the testing errors were compared with three embedded classification methods (PLRL<sub>1</sub>, PsLR, and SCRDA) for the reason that these methods are not subject to the pre-determined gene subset as the proposed approach. In addition, we included the T-test combined with logistic regression as a representative for the filter approach. Table 10 shows that IRPLRL<sub>1</sub> not only led to small gene subsets but also outperformed other methods in terms of the average testing error. We noted that IRPLRL<sub>1</sub> and PLRL<sub>1</sub> yielded the same average number of significant genes while the selected genes were not exactly identical. Based on the lower misclassification

**Table 9:** Comparison of classification methods for leukemia data

Approach	Method	Author	Number of genes	Testing error
Filter	T-test + LR	This study	19	8/34
	GS + WV	Golub (1999)	50	4/34
	FCS + SVM	Weston (2000)	20	3/34
	BW + PLRL <sub>2</sub>	Zhu (2004)	16	3/34
	BW + SVM	Zhu (2004)	22	3/34
	FDR + PLRL <sub>2</sub>	Liao (2007)	20	1/34
Wrapper	PGA + WV	Liu (2001)	29	4/34
	RFE + PLRL <sub>2</sub>	Zhu (2004)	26	1/34
	RFE + SVM	Zhu (2004)	31	1/34
Embedded	PLRL <sub>1</sub>	This study	33	3/34
	SCRDA	Guo (2008)	46	1/34
	PsLR (LR + SVM)	Zhang (2007)	7129	1/34
<b>New</b>	<b>IRPLRL<sub>1</sub></b>	<b>This study</b>	<b>11</b>	<b>1/34</b>

Note: LR = logistic regression; GS = Golub’s ranking (see Section 2.1.5); WV = weighted voting; FCS = Fisher’s criterion score (see Section 2.1.5); SVM = support vector machine; BW = Dudoit’s ranking (see Section 2.1.5); PLRL<sub>2</sub> = penalized logistic regression with  $L_2$ -norm penalty; FDR = false discovery rate (see Section 2.1.4); PGA = parallel genetic algorithm; RFE = recursive feature elimination; PLRL<sub>1</sub> = penalized logistic regression with  $L_1$ -norm penalty; SCRDA = shrunken centroids regularized discriminant analysis; PsLR = pseudo logistic regression.

rate produced by IRPLRL<sub>1</sub>, it seems that IRPLRL<sub>1</sub> can identify globally important genes more accurately. This further investigation demonstrated that using a smaller variable subset in PLRL<sub>1</sub> enables more variable selection consistency when  $p \gg n$ .

### 3.4.3 Breast Cancer Data

This new microarray dataset of breast cancer was provided by Emory Winship Cancer Institute (<http://www.cancer.emory.edu/>). In this microarray experiment, tissue samples were extracted from the collection of breast cancer patients’ raw samples and placed in two separated Sentrix Array Matrix (SAMs) panels. Each panel, containing 96 (12×8) samples, then went through the cDNA (complementary DNA) Annealing Selection extension and Ligation (DASL) experiment. After running the

**Table 10:** Comparison of classification methods for leukemia data through 50 new splits

Method	Number of genes	Testing error
T-test + LR	10	3.9/34
PLRL <sub>1</sub>	37	1.6/34
<b>IRPLRL<sub>1</sub></b>	<b>37</b>	<b>1.3/34</b>
SCRDA	137	1.6/34
PsLR (LR + SVM)	7129	1.5/34

DASL experiment, the image fluorescent intensities of 1488 gene probes were interpreted in BeadStudio and raw signal intensities were exported for the meta-analysis: The average signal intensity, the detected genes (p-values less than 0.01), the background, and the noise (the standard deviation of background signals) were analyzed for trends by plate, row, column, and immunohistochemical (IHC) receptor status. Although stochastic variability existed in all aforementioned categories, no alarming trends were observed. To further investigate this data, some data cleansing rules were applied to both SAMs utilized for this experiment. As breast cancer subtypes are traditionally defined by three immunohistochemical (IHC) receptors (i.e. estrogen receptor, ER; progesterone receptor, PGR; and human epidermal growth factor receptor 2, ERBB2), we removed controls and samples with no IHC receptor status on ESR1, PGR, or ERBB2, which resulted that the sample size was reduced to 165. Of these 165 samples, 13 had an average signal intensity of less than 3,000 which were determined failed as well as one sample with a background and noise signal over 2000, also determined failed. A subsequent meta-analysis was conducted over the remaining 151 samples, and it revealed that removing controls and failed samples did, in fact, equalize the average signal intensity between the two DASL experiments. Of the 151 cleaned samples, 21 patients were tested negative on ESR1, PGR, and ERBB2 via IHC (denoted as NNN), and 70 patients were tested positive for at least one receptor (denoted as non-NNN). From clinical experience, NNN carcinomas are

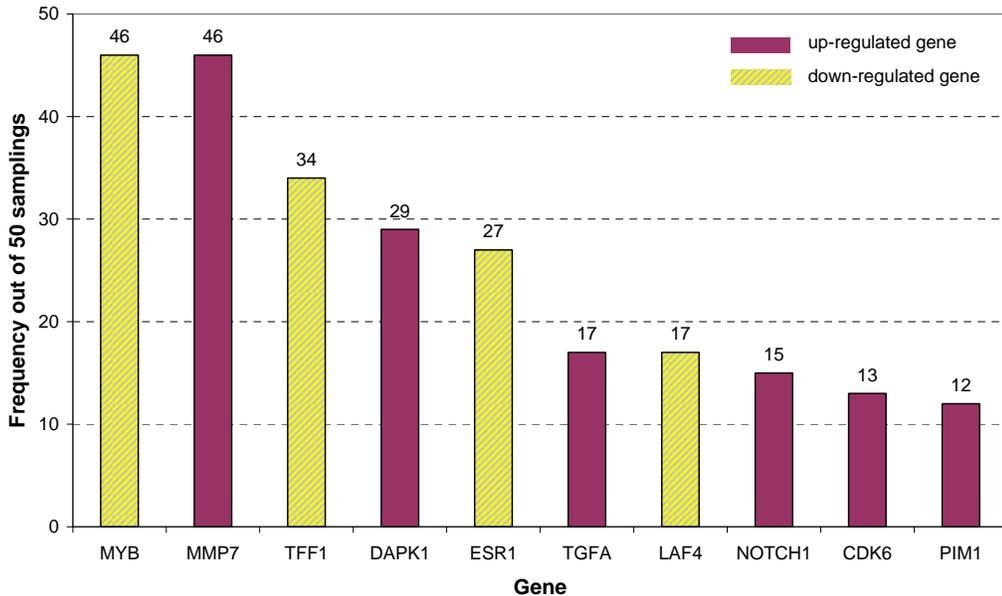
extremely aggressive, and cancer patients with this type of breast cancer tend to have poor outcome. Accordingly, the objective of this microarray experiment is to classify patients with the NNN versus the non-NNN subtypes of breast tumors. In this experiment, most patients had technical replicates (i.e., RNA from that patient was measured via DASL more than once) while the number of technical replicates was not the same for each patient due to the limited availability of RNA derived from formalin-fixed paraffin-embedded (FFPE) tissues. Overall, there are 32 NNN samples and 119 non-NNN samples.

To assess the performance of  $\text{IRPLRL}_1$  and other classification methods, we randomly chose 21 NNN samples and 79 non-NNN samples to form a training set while the remaining 11 NNN samples and 40 non-NNN samples were reserved to construct a testing set. We repeated the above splitting procedure 50 times and treated all samples were independent to patients (i.e., the correlations between samples from the same patients were ignored). For each split, five classification methods were applied. Apart from  $\text{IRPLRL}_1$ , the classification performance of one filter approach (T-test combined with logistic regression) and three embedded approaches were compared in Table 11. The average testing errors were similar among three embedded approaches and  $\text{IRPLRL}_1$ , but the proposed  $\text{IRPLRL}_1$  obtained a smaller gene subset than the other methods. In addition, we counted the frequency of active genes over 50 splittings. Figure 5 exhibits the top 10 active genes in terms of the frequency identified by  $\text{IRPLRL}_1$ . Of which, six genes are up-regulated while others are down-regulated. Up-regulated (down-regulated) genes refer to situations that patients with the NNN breast cancer subtype have higher (lower) gene expression intensities than patients with the non-NNN breast cancer subtype. The pairwise Pearson correlations of these 10 genes are between -0.49 and 0.66. Although  $\text{IRPLRL}_1$  was not directly developed for tackling multi-collinearity problems, it does not seem to have a serious multi-collinearity problem on the selected genes. The descriptions of the top 10 genes

are listed in Table 12. Overall, the proposed iterative reselection penalized logistic regression yielded reasonable and satisfied results in this new microarray experiment.

**Table 11:** Comparison of classification methods for breast tumors with NNN / non-NNN subtypes

Method	Number of gene probes	Testing error
T-test + LR	15	14.35%
PLRL <sub>1</sub>	19	7.1%
<b>IRPLRL<sub>1</sub></b>	<b>10</b>	<b>6.5%</b>
SCRDA	51	6.6%
PsLR (LR + SVM)	1488	7.6%



**Figure 5:** Top 10 active genes in terms of the frequency identified by IRPLRL<sub>1</sub>

### 3.5 Simulation Study

In this section, we study the finite sample performance of the iterative reselection penalized logistic regression (IRPLRL<sub>1</sub>) in a more controlled manner. The performance is going to be evaluated and compared with non-iterative PLRL<sub>1</sub> in two aspects: the accuracy of variable selection and the misclassification rate. The accuracy of variable selection is measured by two scores: (i) the average number of the correctly identified

**Table 12:** Description of top 10 genes for breast tumors with NNN / non-NNN subtypes

Gene	Description
MYB	Homo sapiens v-myb myeloblastosis viral oncogene homolog (avian)
MMP7	Homo sapiens matrix metalloproteinase 7 (matrilysin; uterine)
TFF1	Homo sapiens trefoil factor 1 (breast cancer; estrogen-inducible sequence expressed in)
DAPK1	Homo sapiens death-associated protein kinase 1
ESR1	Homo sapiens estrogen receptor 1 (ESR1)
TGFA	Homo sapiens transforming growth factor
LAF4	Homo sapiens lymphoid nuclear protein related to AF4
NOTCH1	Homo sapiens Notch homolog 1; translocation-associated ( <i>Drosophila</i> )
CDK6	Homo sapiens cyclin-dependent kinase 6
PIM1	Homo sapiens pim-1 oncogene

non-zero coefficients; and (ii) the average number of misspecified zero coefficients. We also examine the frequency of correctly identified zero and non-zero coefficients in repeated simulations. The misclassification rate is calculated by the percentage of misclassified samples in the testing dataset. The simulations are conducted with different  $p$  for a fixed sample size to demonstrate the performance of variable selection and classification when the number of predictors increases in penalized logistic regression.

The simulation setup of the logistic regression model (3) is as follows. Let  $\mathbf{x}_i$  be a  $p \times 1$  covariate vector, generated from a multi-normal distribution with mean 0 and covariance matrix  $\Sigma$ . The first 25 covariates are assumed to be relevant predictors with non-zero coefficients which are independent of the remaining  $p-25$  irrelevant predictors with zero coefficients. The pairwise correlation of the first 25 and the remaining  $p-25$  covariates is specified as a function of  $\gamma$  ( $= 0.8$ ), decreasing in the

power of the index distance between two covariates, as shown in (7).

$$\Sigma = [\sigma_{jj'}^2] = \begin{cases} 1, & \text{for all } j = j' \\ 0.8^{|j-j'|}, & \text{for } j \neq j' \text{ and } (j, j') \in \{1, \dots, 25\} \text{ or } (j, j') \in \{26, \dots, p\} \\ 0, & \text{otherwise} \end{cases} \cdot \quad (7)$$

We also differentiate the impact of the first 25 relevant predictors on a binary outcome  $y_i$  by setting different coefficients ( $\beta_j$ ). Let the first five predictors be the most important with the coefficient 2, followed by two relatively less important sets of ten predictors with the coefficients 1 and 0.5, respectively; other coefficients are set to zero. This setting implies that only the first 25 relevant predictors are in use of generating  $\pi_i = \text{Prob}(y_i = 1)$ .

For different numbers of covariates ( $p = 3000$  and  $5000$ ), 200 independent observations were simulated by

$$\pi_i = \frac{\exp(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta})}{1 + \exp(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta})}$$

and

$$y_i \sim \text{Bernoulli}(\pi_i),$$

where

$$\boldsymbol{\beta} = [\beta_1, \dots, \beta_p]^T \text{ with } \beta_j = \begin{cases} 2, & j = 1, \dots, 5 \\ 1, & j = 6, \dots, 15 \\ 0.5, & j = 16, \dots, 25 \\ 0, & j = 26, \dots, p \end{cases},$$

and they were fairly split into a training set ( $n = 100$ ) and a testing set. The coefficients  $\boldsymbol{\beta}$  were iteratively estimated by (6) with 100 training data and with starting values  $\tilde{\beta}_0 = \log \bar{y}/(1 - \bar{y})$ , where  $\bar{y} = \sum_{i=1}^n y_i/n$  and  $\tilde{\beta}_j = 0$ ,  $j = 1, \dots, p$ . In which, 5-fold cross validation based on the training set was performed to choose a shrinkage parameter  $\lambda$  in PLR. Then, the number of estimated non-zero coefficients from the training set and the misclassification rate in the testing set were reported.

In this simulation study, the above procedure was repeated 500 times for both PLRL<sub>1</sub> and IRPLRL<sub>1</sub> for which the pre-set parameters  $(c, T, G, M)$  are the same as those used in Section 3.4. Table 13 summarizes the aforementioned two scores regarding the accuracy of variable selection and the misclassification rate from the two models with different  $p$ . It can be seen that both PLRL<sub>1</sub> and IRPLRL<sub>1</sub> were comparative in identifying relevant predictors with non-zero coefficients  $(\beta_1, \dots, \beta_{25})$  as indicated in the 4<sup>th</sup> column of Table 13; whereas, the proposed IRPLRL<sub>1</sub> misspecified fewer zero coefficients  $(\beta_{26}, \dots, \beta_p)$  than did the PLRL<sub>1</sub>, as in the 5<sup>th</sup> column. On the other hand, IRPLRL<sub>1</sub> produced smaller averaged misclassification rates than PLRL<sub>1</sub> in the testing set.

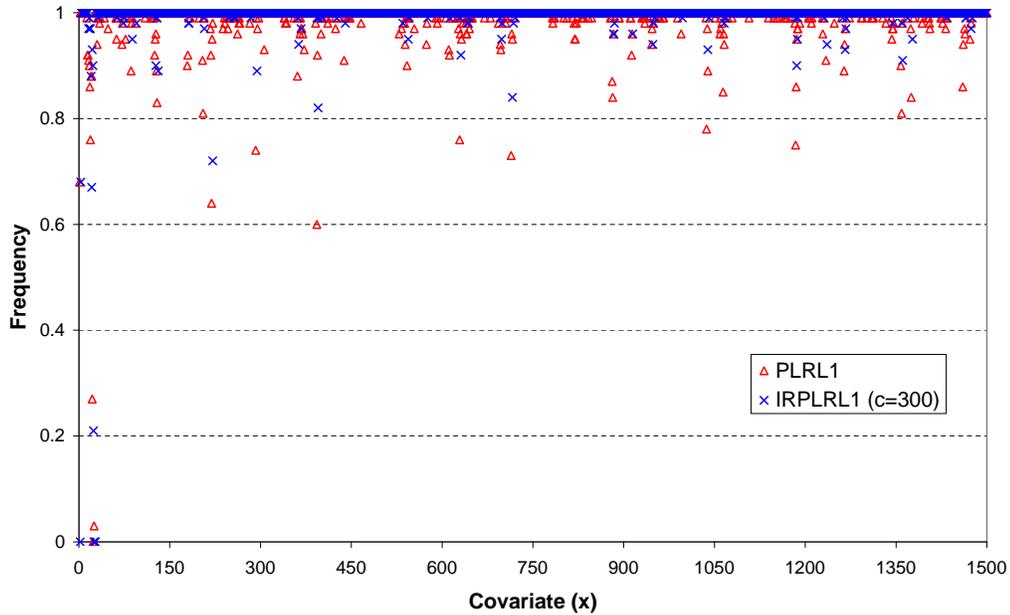
**Table 13:** Comparison of PLRL<sub>1</sub> and IRPLRL<sub>1</sub>( $c = 300$ ) with simulated data

Model	$p$	$\lambda$	Ave. number of estimated non-zero's		Ave. testing error (standard deviation)
			$\hat{\beta}_1 \sim \hat{\beta}_{25}$	$\hat{\beta}_{26} \sim \hat{\beta}_p$	
PLRL <sub>1</sub>	3000	0.14	20.0	28.5	8.18% (2.92%)
IRPLRL <sub>1</sub>	3000	0.13	20.2	9.1	7.46% (2.03%)
PLRL <sub>1</sub>	5000	0.16	16.3	28.9	8.22% (3.21%)
IRPLRL <sub>1</sub>	5000	0.14	16.3	7.0	7.72% (1.80%)

To better study the performance of PLRL<sub>1</sub> and IRPLRL<sub>1</sub> in variable selection, *the frequencies of the covariates being correctly identified* from the 500 repeated simulations are plotted in Figures 5 and 6. For a clear view, only half of the covariates are displayed, among which the first 25 covariates are the relevant predictors and the rest are randomly chosen from the irrelevant predictors. The red triangle represents the frequency of the covariates being correctly identified by PLRL<sub>1</sub> while the blue cross represents that by IRPLRL<sub>1</sub>. From these figures, we notice that both models can identify the relevant covariates successfully except some of  $(x_{16}, \dots, x_{25})$  with comparably smaller coefficients. This explains the results that the average numbers of the correctly identified non-zero coefficients ranges from 16 to 20 in Table 13, rather

than being close to 25. In addition, more triangles than crosses are lying away from 1 on the y-axis, which implies that  $\text{PLRL}_1$  is generally not as accurate in variable selection as the proposed  $\text{IRPLRL}_1$ .

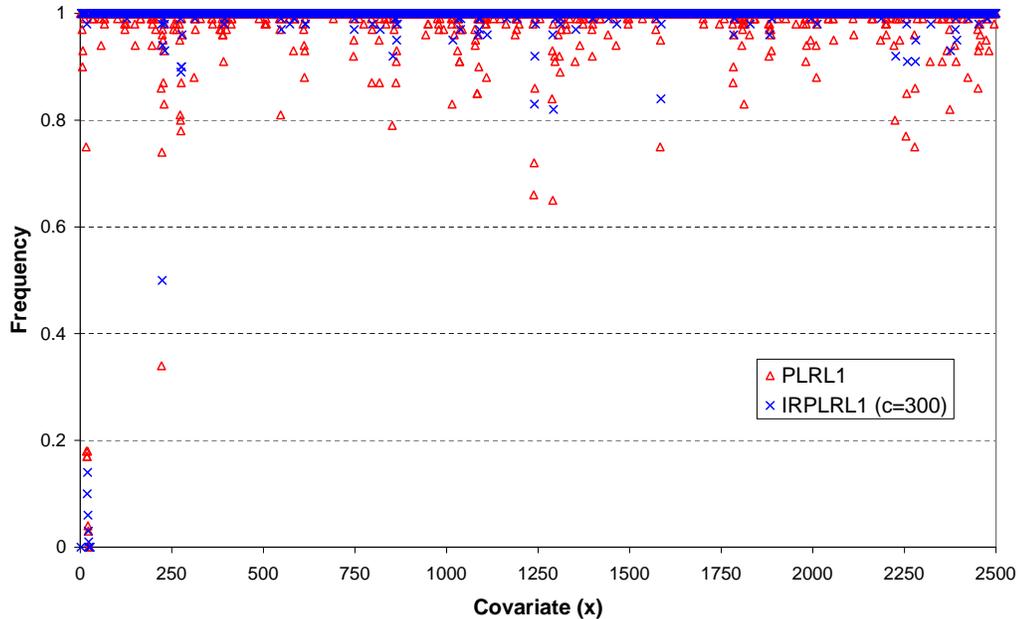
All in all, this simulation study shows that the proposed iterative reselection algorithm improved the variable selection consistency of  $\text{PLRL}_1$  and further yielded better classification performance in terms of testing errors. In other words, with the controlled size of variable sets used in  $\text{PLRL}_1$ , it broadens the application of  $\text{PLRL}_1$  to higher-dimensional problems.



**Figure 6:** Comparison of  $\text{PLRL}_1$  and  $\text{IRPLRL}_1$  ( $p = 3000$ )

### 3.6 Conclusion and Discussion

Due to high dimensionality and small sample size, it is challenging to solve classification problems with microarray experiments. For such problems, penalized logistic regression is one of widely used classification methods. However, its performance on estimation convergence and prediction accuracy deteriorates as the number of genes



**Figure 7:** Comparison of  $\text{PLRL}_1$  and  $\text{IRPLRL}_1$  ( $p = 5000$ )

increases. To overcome these deficiencies, a new approach, iterative reselection penalized logistic regression, was proposed in this study.

The new approach proceeds through the examination of a sequence of PLR with smaller variable subsets and the integration of a heuristic search algorithm so that effective estimation and good prediction can be obtained. The proposed approach was evaluated by both real-world microarray datasets and a simulation study. From the classification results of four microarray datasets with the comparison to some existing methods, the proposed approach attractively generated smaller models with higher prediction accuracy. The proposed approach was also shown to improve the variable selection consistency of PLR and achieve better classification performance through the simulation study. These results illustrated the superiority of the new approach over some existing methods for high-dimensional classification problems.

In the proposed iterative reselection algorithm, we adopted penalized logistic regression with the Lasso penalty to perform variable selection and classification simultaneously while this algorithm is not limited to a specific model. Other classification

methods, as long as they are applicable to the  $p > n$  situation or with certain variable selection scheme can take the place. In addition, other heuristic algorithms than simulated annealing can be used to search the best subset for combinatorial optimization problems. To assess the performance of the selected gene subset in each iteration, a cross validation procedure is used in the proposed algorithm while the cross validation error is not necessarily a robust criterion in all situations; some random errors may exist in itself. Thus, a better scoring method for evaluating variable selection performance is worth future study. Some parameters, such as  $c$  and  $T$ , used in the proposed algorithm may also need further evaluation for the effects on variable selection consistency and on convergence rate.

In this study, we focus on two-class gene expression in microarray applications. For the multi-class prediction problems, Friedman et al. (2010) proposed a coordinate decent algorithm for penalized multi-logit model in high-dimensional situations. Thus, this study can be extended to multi-class problems as future work. In addition, under the general framework of penalized logistic regression, it is possible to use other penalty functions (Fan and Li, 2001). Antoniadis and Fan (2001) also provided some insights into choosing a penalty function. Although we did not pursue other penalty functions in this study, we believe that the new approach would also be applicable to other penalties and similar conclusions could be drawn.

## CHAPTER IV

### A NEW MODELING METHOD: PENALIZED LOGISTIC MIXED MODEL

In this chapter, we propose a new embedded classification method in the consideration of different variability existing in experimental observations. This study is motivated by microarray experiments with two types of replicates. We first introduce the replication of microarray experiments in Section 4.1, followed by a review section of recent modeling and theory development in the framework of penalized regression models. The proposed new classification method along with an estimation algorithm and an asymptotic property are described in Sections 4.3 – 4.5. The performance of different penalty functions is also compared by a simulation study in Section 4.6. The application of the new method to a breast cancer microarray experiment is illustrated in Section 4.7. Some discussion and extended work are remarked in the end of this chapter.

#### *4.1 Replication of Microarray Experiments*

A microarray experiment is a multi-step process in which multiple sources of variability exist. In order to increase the overall precision of an experiment, *replication* is an important consideration (Fisher, 1951). There are two types of replicates in microarray experiments: biological replicates and technical replicates. *Biological replicates* refer to collecting several mRNA samples from a number of different but similar subjects. These replicates reflect genetic differences among experimental subjects and are of most interest for researchers to make inferences from samples to populations. *Technical replicates* refer to multiple measurements on the same experimental subject,

which are useful to assess platform reproducibility and to deal with technical variation arising from mRNA extraction, labeling, hybridization, scanning, and imaging (Amaratunga and Cabrera, 2004).

In the past, most gene expression microarray experiments focused on biological replicates only (Alon et al., 1999; Golub et al., 1999; West et al., 2001). However, it is important to realize that experiments with *technical replicates* are able to provide more reliable analyses. Lee et al. (2000) provided a nice illustration. They studied technical replication of expression measurements for 288 gene probes which were obtained under the same experimental conditions from the same human tissue sample. In their controlled experiment, only 32 out of the 288 genes contained Alu messages, and they were expected to show a high level of signals and be classified as expressed. The consistency of three replicates was checked and found that the numbers of genes classified as expressed are 55, 36, and 58, respectively, which resulted in a large number of false positives. However, based on the combined data from all replicates, the classification results produced only two false positives and no false negatives. Their study showed that technical replication in microarray experiments is neither equivalent to duplication nor a waste of scientific resources. In fact, experimental replication is essential to reliable scientific discovery in genetic research.

## ***4.2 Motivation and Literature Review***

Nowadays, microarray experiments with both biological and technical replicates are commonly seen in practice. However, an appropriate classification method for this type of microarray gene expression data is in short supply. Although penalized logistic regression (PLR) is widely used for classification in microarray studies (Roth, 2002; Shevade et al., 2003; Liu et al., 2007) as reviewed in Section 3.1, it is not suitable to be directly applied to experiments with both biological and technical replicates. The main reason is that PLR relies on the assumption that the observed binary

responses are mutually independent. This assumption holds if experiments involve biological replicates only and experiments are conducted on independent subjects. In this situation, PLR has been well-known for simultaneous variable selection and classification. However, when some mRNA samples are taken from the same subject, these technical replicates are no longer independent due to certain unobserved shared factors. This violates the essential assumption of PLR and makes PLR inappropriate. Therefore, a more advanced classification method for high-dimensional predictors with not only small but also *correlated samples* is needed.

In concept, correlated samples can be easily taken into account by incorporating *random effects* in PLR for the heterogeneity among samples. However, this is more than a simple extension for PLR in both computational estimation and theoretical derivation. In the estimation aspect, besides an efficient algorithm for the estimation of fixed effects in high dimension with small sample size as introduced in Section 3.3.1, numerical or Monte Carlo integration techniques are required because analytical solutions of high-dimensional integrals over the distribution of random effects are not available for the logistic regression models. This certainly makes computational work more intractable.

Regarding the theoretical issue of the selection consistency, a number of prior work investigated the asymptotic properties of penalized logistic models (as in Table 14). Knight and Fu (2000) first showed that the Lasso penalty is root- $n$  consistent. Fan and Li (2001) proposed a new penalty function, smoothly clipped absolute deviation (SCAD) and showed its variable selection consistency with different models under low dimensionality ( $p < N$ ). Zou (2006) proposed adaptive weights for penalizing coefficients in Lasso, named adaptive Lasso (AdaLasso). In linear models with  $p < N$ , it was proved that AdaLasso is variable selection consistent under some general conditions. Recently, Huang et al. (2008) showed that AdaLasso can still be

**Table 14:** Related theoretical work of penalized regression models

Author	Penalized Model	Penalty	$p > N$ ?	Dependent Samples ?
Knigh and Fu (2000)	linear / logistic	Lasso	×	×
Fan and Li (2001)	linear / logistic	SCAD	×	×
Zou (2006)	linear / logistic	AdaLasso	×	×
Zhao and Yu (2006)	linear	Lasso	✓	×
Huang et al. (2008)	linear	AdaLasso	✓	×

variable selection consistent in high dimensionality if certain conditions of orthogonality are satisfied. Zhao and Yu (2006) also discussed the conditions of the variable selection consistency for the Lasso penalty in high-dimensionality. Even though the last two articles developed asymptotic theories under  $p > N$ , their findings are restricted to linear models and independent samples. For binary classification problems in high dimension with small and correlated samples, it remains to be investigated the asymptotic property of variable selection in the framework of penalized logistic regression. The development of asymptotic theories is crucial without the independence assumption.

The aim of this study are three-fold: (i) propose a new classification method for high-dimensional predictors with small and correlated samples; (ii) introduce a new estimation algorithm for the new modeling method with both fixed and random effects; (iii) pursue the asymptotic property of variable selection for the new model. These will be deployed in the next three sections.

### 4.3 Penalized Logistic Mixed Model

Assuming independent binary realization  $y_i$ 's are taking values 0 or 1. A logistic regression model can be written as

$$\text{logit}(\pi_i) = \log \frac{\Pr(y_i = 1)}{1 - \Pr(y_i = 1)} = \mathbf{x}_i^T \boldsymbol{\alpha},$$

where  $\pi_i = E(y_i) = \Pr(y_i = 1)$ , the  $p \times 1$  vector  $\mathbf{x}_i = (x_{i,1}, \dots, x_{i,p})^T$  denotes the covariates, and  $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_p)^T$  denotes the corresponding parameters. The log-likelihood function for  $\boldsymbol{\alpha}$  is

$$L(\boldsymbol{\alpha}) = \sum_i \left( y_i \log \frac{\pi_i(\boldsymbol{\alpha})}{1 - \pi_i(\boldsymbol{\alpha})} + \log(1 - \pi_i(\boldsymbol{\alpha})) \right). \quad (8)$$

Penalized logistic regression (PLR) has been widely used for model fitting in high-dimensional classification. It is a regularization technique for simultaneous estimation and variable selection. The idea is to add a penalty function into the logistic regression likelihood (8). The penalized logistic regression parameters  $\boldsymbol{\alpha}$  can be estimated as follows

$$\hat{\boldsymbol{\alpha}} = \operatorname{argmax}_{\boldsymbol{\alpha}} \left\{ \left[ \sum_i y_i \log \frac{\pi_i(\boldsymbol{\alpha})}{1 - \pi_i(\boldsymbol{\alpha})} + \log(1 - \pi_i(\boldsymbol{\alpha})) \right] - \sum_{k=1}^p P_\lambda(\alpha_k) \right\},$$

where  $P_\lambda(\alpha_j)$  is a penalty function with parameter  $\lambda$ . More discussions about the estimation and algorithm can be found in Friedman et al.(2007), Goeman (2008), and Friedman et al. (2010).

PLR is generally used in many applications; however, it is limited to the assumption that all observations are independent. To take into account heterogeneity among experimental subjects as well as correlations among observations from the same experimental subject, we propose a new model, a *penalized logistic mixed model (PLMM)*. The idea is to assume that experimental subjects are sampled from a population; and random effects  $\beta_i$ 's are used to represent the heterogeneity among experimental subjects. That is,  $\beta_i$ 's are independent from a distribution with parameters  $D$ , denoted

by  $f_{\beta}(\beta|D)$ . In particular,  $\beta = (\beta_1, \dots, \beta_n)^T$  is assumed to be normally distributed with mean  $\mathbf{b}$  and variance  $\Sigma_b$ , where  $\mathbf{b}$  is a column of  $b$ 's with length  $m$ ,  $\Sigma_b = \sigma_b^2 \mathbf{I}_m$ ,  $\mathbf{I}_m$  is the  $m \times m$  identity matrix, and  $D = (\mathbf{b}, \sigma_b^2)$ . Correlations among observations on the same experimental subject arise from their shared variables,  $\beta_i$ . Assuming  $y_{ij}$  is the  $j^{\text{th}}$  binary observation from subject  $i$ , where  $j = 1, \dots, m$ ,  $i = 1, \dots, n$ , and  $N = nm$  denotes the total number of observations. The PLMM can be written as

$$\text{logit}(\pi_{ij}^{\beta}) = \eta_{ij}^{\beta} = \mathbf{z}_{ij}^T \beta + \mathbf{x}_{ij}^T \alpha, \quad (9)$$

where  $\pi_{ij}^{\beta} = P(y_{ij} = 1|\beta)$ ,  $\beta \sim N(\mathbf{b}, \Sigma_b)$ . The vector  $\mathbf{x}_{ij} = \{x_{ij,1}, \dots, x_{ij,p}\}^T$  denotes the covariates associated with the  $p$ -dimensional fixed effects  $\alpha = (\alpha_1, \dots, \alpha_p)^T$  and  $\mathbf{z}_{ij} = \{z_{ij,1}, \dots, z_{ij,n}\}^T$  denotes the design matrix for the random effects  $\beta$  such that  $\mathbf{z}_{ij}^T \beta = \beta_i$ . That is,  $z_{ij,i} = 1$  and  $z_{ij,t} = 0$  for all  $t \neq i$ . Note that the heterogeneity is directly modeled through subject-specific parameters. If a random intercept alone is not sufficient to capture the variation exhibited in the data, this model can be easily extended to a general form by incorporating more complicated random effects.

The likelihood function for (9) is given by

$$L(\alpha, D) = \int \left[ \prod_{i=1}^n \prod_{j=1}^m \exp \left( y_{ij} \log \frac{\pi_{ij}^{\beta}}{1 - \pi_{ij}^{\beta}} + \log(1 - \pi_{ij}^{\beta}) \right) \right] \exp(-\beta^T \Sigma_b^{-1} \beta) d\beta.$$

Since the number of covariates is much larger than the number of observations ( $p \gg N$ ), traditional estimation methods that maximizes the likelihood, such as estimating fixed effects  $\alpha$  in a generalized linear mixed model (GLMM), cannot be employed. Similar to PLR, a regularization technique is applied to achieve simultaneous variable selection and estimation in PLMM. The penalized log-likelihood function can be written as

$$PL(\alpha, D) = \log(L(\alpha, D)) - \sum_{k=1}^p P_{\lambda}(|\alpha_k|). \quad (10)$$

For the penalty function  $P_{\lambda}(|\alpha_k|)$ , there are many discussions in the literature (Fan and Li, 2001; Friedman et al., 2007). In this paper, we mainly focus on two

widely used penalty functions. The first one is Lasso (Tibshirani, 1996 and 1997; Donoho and Johnstone, 1994), which can be written as

$$P_\lambda(|\alpha_k|) = \lambda|\alpha_k|. \quad (11)$$

The second one is adaptive Lasso (Zou, 2006), which can be written as

$$P_\lambda(|\alpha_k|) = \lambda v_k |\alpha_k|, \quad (12)$$

where  $\mathbf{v} = (v_1, \dots, v_p)$  is a known weights vector. Zou (2006) suggested a weights vector as a function of the ordinary least squares estimators. However, the ordinary least squares estimator is no longer feasible as  $p \gg N$ . Therefore, the marginal regression estimators are suggested by Huang et al. (2008), i.e.,  $v_k = |\tilde{\alpha}_k|^{-\gamma}$  and  $\gamma > 0$ . Though two penalty functions are pursued here, similar results can be extended to other penalty functions, such as the elastic net (Zou and Hastie, 2005) and the smoothly clipped absolute deviation penalty (Fan and Li, 2001). More discussion can be found in Section 4.4.

In general, PLMM includes two important elements: random effects and a penalty function. By incorporating random effects, PLMM can be used to model dependent observations and provides higher prediction accuracy. On the other hand, the utilization of penalized likelihood can achieve simultaneous variable selection and estimation in high-dimensional problems efficiently. Hence, PLMM can be used in high-dimensional ( $p \gg N$ ) binary classification without assuming that observations are independent.

Despite the flexibility of PLMM, this new classification method poses some challenges. First, incorporating random effects makes the estimation more complicated than PLR. Mathematically, the parameters can be estimated by

$$(\hat{\boldsymbol{\alpha}}, \hat{D}) = \operatorname{argmax}_{(\boldsymbol{\alpha}, D)} PL(\boldsymbol{\alpha}, D). \quad (13)$$

However, due to the need of the numerical evaluation of high-dimensional integration, this standard estimation (13) is limited to simple models. To avoid computational

problems, an efficient estimation approach is needed. Second, theoretical study regarding selection consistency is challenging in PLMM because of high dimensionality ( $p \gg N$ ) and random effects.

#### **4.4 MCEM Algorithm**

In this section, an efficient algorithm is introduced to estimate parameters in PLMM with the Lasso and the adaptive Lasso penalties. Difficulties in parameter estimation are mainly from two aspects. One is the estimation of fixed effects in PLMM due to high dimensionality ( $p \gg N$ ); the other is the estimation of variance components due to the random effects involved in PLMM. Note that without random effects, estimation in PLMM is the same as that in PLR with high dimension and low sample size. Below we first review some work in this regard.

A number of authors proposed algorithms to solve PLR in high dimension with small sample size (Fu, 1998; Efron et al., 2004). Recently, Friedman et al. (2010) developed an efficient algorithm by applying the coordinate descent method in PLR. It is well-known that the Newton algorithm for maximizing the unpenalized log-likelihood amounts to iteratively reweighted least squares (IRLS). Therefore, Friedman et al. (2010) proposed to estimate the penalized logistic regression parameters via the coordinate descent method with iterative re-weights.

The coordinate descent algorithm is an effective approach to handle PLR when  $p \gg N$ . This algorithm, however, cannot be directly applied to PLMM estimation because it does not consider any random effects. With the random effects in the PLMM model, integration over the distribution of random effects must be performed. As a result, estimation is much more complicated because the integration cannot be expressed in a closed form. Instead of direct calculation, a Metropolis algorithm (Tanner, 1993) is applied to overcome this computation difficulty, which

leads to a hybrid algorithm, named *Monte Carlo Expectation-Maximization via coordinate descent method (MCEM-CD)*. This is a modified version of the Monte Carlo Expectation-Maximization (MCEM) algorithm (Chan and Ledolter, 1995), which is generally used in GLMM estimation (McCulloch, 1997).

The main idea of the MCEM-CD algorithm is to construct the EM algorithm by regarding the random effects  $\boldsymbol{\beta}$  as missing data. Thus, the complete-data log-likelihood for  $(y_{ij}, \boldsymbol{\beta})$  is given by

$$CL = \sum_{i=1}^n \sum_{j=1}^m \left( y_{ij} \log \frac{\pi_{ij}^{\boldsymbol{\beta}}}{1 - \pi_{ij}^{\boldsymbol{\beta}}} + \log(1 - \pi_{ij}^{\boldsymbol{\beta}}) \right) - \lambda \sum_{k=1}^p |\alpha_k| + \log f_{\boldsymbol{\beta}}(\boldsymbol{\beta}|D), \quad (14)$$

where  $\pi_{ij}^{\boldsymbol{\beta}}$  is based on (9), the penalty is based on (11), and  $f_{\boldsymbol{\beta}}$  is assumed to be normally distributed). In this EM algorithm, the M-step is to maximize (14) with respect to  $\boldsymbol{\alpha}$  and  $D$ . Because the fixed effects  $\boldsymbol{\alpha}$  enter only the first two terms, the M-step with respect to  $\boldsymbol{\alpha}$  uses only the first two terms that can be formulated as maximizing the likelihood in PLR with  $p \gg N$ . Therefore, the coordinate descent method (Friedman et al., 2010) can be applied in the M-step for estimating fixed effects.

In the E-step of the EM algorithm, the conditional distribution of  $\boldsymbol{\beta}|y$  that involves the distribution of  $y$  is difficult to calculate directly. Therefore, a Metropolis algorithm is applied to produce random draws from the conditional distribution  $\boldsymbol{\beta}|y$ . This can be specified as follows. Assuming the candidate distribution,  $g_{\boldsymbol{\beta}}(\boldsymbol{\beta})$  is from a normal distribution with mean  $\mathbf{b}$  and variance  $\Sigma_{\mathbf{b}}$ . Let  $\boldsymbol{\beta}$  denote the previous draw from the conditional distribution. Denote  $\boldsymbol{\beta}^* = (\beta_1, \beta_2, \dots, \beta_{r-1}, \beta_r^*, \beta_{r+1}, \dots, \beta_n)^T$ , where  $\beta_r^*$  is a new value generated from the candidate distribution of  $\boldsymbol{\beta}|y$ . The Metropolis algorithm accepts  $\boldsymbol{\beta}^*$  as the new value with probability  $P_r(\boldsymbol{\beta}, \boldsymbol{\beta}^*)$  is given by

$$P_r(\boldsymbol{\beta}, \boldsymbol{\beta}^*) = \min \left\{ 1, \frac{f_{\boldsymbol{\beta}|y}(\boldsymbol{\beta}^*|y, \boldsymbol{\alpha}, \Sigma_{\mathbf{b}})g_{\boldsymbol{\beta}}(\boldsymbol{\beta})}{f_{\boldsymbol{\beta}|y}(\boldsymbol{\beta}|y, \boldsymbol{\alpha}, \Sigma_{\mathbf{b}})g_{\boldsymbol{\beta}}(\boldsymbol{\beta}^*)} \right\}, \quad (15)$$

where  $g_{\boldsymbol{\beta}}(\boldsymbol{\beta})$  is a candidate distribution in the Metropolis algorithm. On choosing  $g_{\boldsymbol{\beta}}(\boldsymbol{\beta}) = f_{\boldsymbol{\beta}}(\boldsymbol{\beta}|D)$  and  $\boldsymbol{\beta}$  is normally distributed, the second term in braces in (15)

can be written as

$$\frac{f_{\beta|y}(\boldsymbol{\beta}^*|\mathbf{y}, \boldsymbol{\alpha}, \Sigma_{\mathbf{b}})g_{\beta}(\boldsymbol{\beta})}{f_{\beta|y}(\boldsymbol{\beta}|\mathbf{y}, \boldsymbol{\alpha}, \Sigma_{\mathbf{b}})g_{\beta}(\boldsymbol{\beta}^*)} = \exp\left(\sum_{j=1}^m y_{r_j}(\beta_r^* - \beta_r)\right) \prod_{j=1}^m \frac{1 + \exp(\mathbf{x}_{r_j}^T \boldsymbol{\alpha} + \beta_r)}{1 + \exp(\mathbf{x}_{r_j}^T \boldsymbol{\alpha} + \beta_r^*)}.$$

Details of the MCEM-CD algorithm are described as follows. We first discuss this algorithm with the Lasso penalty.

1. Choose starting values  $\boldsymbol{\alpha}^{(0)}, \sigma_b^{(0)}$ . Set  $l = 0$ .
2. Generate  $U$  values,  $\boldsymbol{\beta}^{(1)}, \boldsymbol{\beta}^{(2)}, \dots, \boldsymbol{\beta}^{(U)}$ , from  $f_{\beta|y}(\boldsymbol{\beta}|\mathbf{y}, \boldsymbol{\alpha}^{(l)}, \sigma_b^{(l)})$  using the Metropolis algorithm in (15).

- (i) Choose  $\boldsymbol{\alpha}^{(l+1)} = (\alpha_1^{(l+1)}, \dots, \alpha_k^{(l+1)}, \dots, \alpha_p^{(l+1)})^T$  to maximize the Monte Carlo estimate

$$\frac{1}{U} \sum_{u=1}^U \left[ \sum_{ij} y_{ij} (\mathbf{x}_{ij}^T \boldsymbol{\alpha} + \mathbf{z}_{ij}^T \boldsymbol{\beta}^{(u)}) - \log(1 + \exp(\mathbf{x}_{ij}^T \boldsymbol{\alpha} + \mathbf{z}_{ij}^T \boldsymbol{\beta}^{(u)})) \right] - \lambda \sum_{k=1}^p |\alpha_k|.$$

This can be achieved by the coordinate descent method. Therefore, we have

$$\alpha_k^{(l+1)} = \frac{1}{U} \sum_{u=1}^U \frac{T(\sum_{i=1}^n \sum_{j=1}^m w_{ij}^{(u)} x_{ij,k} (v_{ij} - \mathbf{x}_{ij}^T \tilde{\boldsymbol{\alpha}} - \mathbf{z}_{ij}^T \boldsymbol{\beta}^{(u)}), \lambda N)_+}{\sum_{i=1}^n \sum_{j=1}^m w_{ij}^{(u)} x_{ij,k}^2}, \quad (16)$$

where

$$v_{ij} = \mathbf{x}_{ij}^T \tilde{\boldsymbol{\alpha}} + \mathbf{z}_{ij}^T \boldsymbol{\beta}^{(u)} + \frac{y_{ij} - \pi_{ij}(\tilde{\boldsymbol{\alpha}}, \boldsymbol{\beta}^{(u)})}{\pi_{ij}(\tilde{\boldsymbol{\alpha}}, \boldsymbol{\beta}^{(u)})(1 - \pi_{ij}(\tilde{\boldsymbol{\alpha}}, \boldsymbol{\beta}^{(u)}))},$$

is the working response, the weights  $w_{ij}$  are defined as

$$w_{ij}^{(u)} = \pi_{ij}(\tilde{\boldsymbol{\alpha}}, \boldsymbol{\beta}^{(u)})(1 - \pi_{ij}(\tilde{\boldsymbol{\alpha}}, \boldsymbol{\beta}^{(u)})),$$

and  $T(a, b)$  is the soft-thresholding operator

$$T(a, b) \equiv \text{sign}(a)(|a| - b)_+ = \begin{cases} a - b & \text{if } a > 0 \text{ and } b < |a|, \\ a + b & \text{if } a < 0 \text{ and } b < |a|, \\ 0 & \text{if } b \geq |a|. \end{cases}$$

(ii) Find  $\sigma_b^{2(l+1)} = \frac{1}{U} \sum_{u=1}^U (\sum_{j=1}^n \beta_j^{(u)2})/n$ .

(iii) Set  $l = l + 1$ .

3. If convergence is achieved, then declare  $\boldsymbol{\alpha}$  and  $\sigma_b$  to be estimates. Otherwise, return to Step 2.

For the adaptive Lasso penalty (12), estimates can be obtained by replacing Step 2 (i) with the following three steps (i-a, i-b, and i-c):

(i-a) Define  $x_{ij,k}^* = x_{ij,k}/v_k$ , where vector  $x_{ij,k}$  represents the  $k^{th}$  variable,  $k = 1, \dots, p$ , and  $v_k$  is assumed to be a function of the marginal regression estimates in  $p \gg N$  problems (Huang et al., 2008).

(i-b) Solve  $\alpha_k$  by replacing  $x_{ij,k}$  with  $x_{ij,k}^*$  in equation (16).

(i-c) Update  $\alpha_k^{*(l+1)} = \alpha_k/v_k$ , for all  $k = 1, \dots, p$ .

This algorithm can be generally used to estimate fixed effects and variance components in PLMM. Although only Lasso and adaptive Lasso are illustrated here, the algorithm can be further extended to other penalty functions by modifying Step 2 (i). Moreover, this algorithm is not restricted to normally distributed random effects. Other distributions of random effects can also be incorporated into the Metropolis procedure in (15).

## 4.5 Selection Consistency

The variable selection consistency of PLMM is pursued in this section. Existing results in the literature mainly focus on the cases where the sample size is larger than the number of covariates. For the high dimension with small sample size problem, theoretical study remain scare and limited to the linear models with independent observations. New theory is called for as PLMM tackles this problem without assuming that observations are mutually independent. Theoretical derivation for PLMM

is, however, more challenging because of the following two reasons. First, PLMM is a generalized linear model which is more complicated than a linear model. Second, random effects are involved to handle correlated binary observations, which would make the derivation more difficult.

As PLMM aims at performing variable selection and estimation simultaneously for  $p \gg N$  problems, an important issue is to study the selection consistency of the estimated parameters. A good variable selection procedure should be able to select the correct model consistently. Traditional selection consistency requires the zero coefficients to be matched but not the signs. A stronger version of the traditional variable selection consistency, sign consistency, was introduced by Zhao and Yu (2006). That is, an estimate  $\hat{\boldsymbol{\alpha}}$  is equal in sign with the true model  $\boldsymbol{\alpha}$  (denoted by  $\hat{\boldsymbol{\alpha}} =_S \boldsymbol{\alpha}$ ) if and only if  $\text{sign}(\hat{\boldsymbol{\alpha}}) = \text{sign}(\boldsymbol{\alpha})$ , where  $\text{sign}(\cdot)$  maps a positive entry to 1, a negative entry to -1, and zero to zero. The sign consistency for PLMM with the Lasso penalty will be discussed in Theorem 1. The assumptions and the proofs are given in Appendix A.

We first assume  $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_q, \alpha_{q+1}, \dots, \alpha_p)^T$ , where  $\alpha_k \neq 0$  for  $k = 1, \dots, q$  and  $\alpha_k = 0$  for  $k = q + 1, \dots, p$ . Let  $\alpha_{(1)} = (\alpha_1, \dots, \alpha_q)^T$  and  $\alpha_{(2)} = (\alpha_{q+1}, \dots, \alpha_p)^T$ . Let  $\mathbf{y}_i = (y_{i1}, \dots, y_{im})$  for  $i = 1, \dots, n$ ,  $\mathbf{y} = (\mathbf{y}_1, \dots, \mathbf{y}_n)'$  be a vector with length  $N$ ,  $\mathbf{X}$  be the corresponding  $N \times p$  matrix associated with the fixed effects, and  $Z$  be the corresponding  $N \times n$  matrix associated with the random effects. We write  $\mathbf{X}(1)$  and  $\mathbf{X}(2)$  as the first  $q$  and the last  $p - q$  columns of  $\mathbf{X}$  respectively and let  $\mathbf{W}$  be the  $N \times N$  diagonal matrix with diagonal terms  $w_{ij} = \pi_{ij}(\boldsymbol{\alpha}, \sigma_b)(1 - \pi_{ij}(\boldsymbol{\alpha}, \sigma_b))$ ,  $C^N = \frac{1}{N} \mathbf{X}^T \mathbf{W} \mathbf{X}$ ,  $C_{11}^N = \frac{1}{N} \mathbf{X}(1)^T \mathbf{W} \mathbf{X}(1)$ ,  $C_{22}^N = \frac{1}{N} \mathbf{X}(2)^T \mathbf{W} \mathbf{X}(2)$ ,  $C_{12}^N = \frac{1}{N} \mathbf{X}(1)^T \mathbf{W} \mathbf{X}(2)$ ,  $C_{21}^N = \frac{1}{N} \mathbf{X}(2)^T \mathbf{W} \mathbf{X}(1)$ . Hence,  $C^N$  can be expressed in a block-wise form as follows:

$$C^N = \begin{bmatrix} C_{11}^N & C_{12}^N \\ C_{21}^N & C_{22}^N \end{bmatrix}.$$

Assuming  $C_{11}^N \rightarrow C_{11}$ , where  $C_{11}$  is positive definite. The following result holds.

**THEOREM 1 (LASSO SIGN CONSISTENCY):** *Under assumptions A1 to A3, if there exists  $0 \leq c_3 < c_2$  for which  $p = O(e^{Nc_3})$ , then PLMM with the Lasso penalty has the sign consistency. In particular, for  $\lambda \propto N^{\frac{1+c_4}{2}}$  with  $c_3 < c_4 < c_2$ ,*

$$P(\hat{\boldsymbol{\alpha}} =_S \boldsymbol{\alpha}) \geq 1 - o(e^{-Nc_3}) \rightarrow 1 \text{ as } N \rightarrow \infty.$$

Theorem 1 shows that using the Lasso penalty in PLMM,  $p$  is allowed to grow much faster than  $N$  (up to exponentially fast) while the sign consistency is still maintained. A special case of this result is the performance of high dimensional PLR. That is, when there is no random effect involved in PLMM, this theorem implies that the sign consistency holds for PLR with high dimension and small sample size.

## ***4.6 Simulation Study***

In this section, we study the finite sample performance of PLMM with the Lasso and the adaptive Lasso penalties. The performance will be evaluated in two aspects: the accuracy of variable selection and the classification error. The accuracy of variable selection is measured by two scores. One is the average number of the relevant covariates that are correctly identified (i.e., covariates with non-zero coefficients) in the repeated simulations, and the other is the average number of the irrelevant covariates that are misspecified (i.e., covariates with zero coefficients). We also look at the frequency of correctly identified relevant and irrelevant covariates in these simulations. The classification error is calculated by the percentage of misclassified samples in the testing dataset. To demonstrate the performance of variable selection and classification with the increasing number of covariates in PLMM, simulations were conducted with different  $p$  given a fixed sample size.

The simulation setup is as follows. Denote  $y_{ij}$  the  $j^{\text{th}}$  binary observation from the experimental subject  $i$ . We consider 20 experimental subjects and 5 samples from each subject, namely,  $i = 1, \dots, 20, j = 1, \dots, 5$ , and the total number of observations

$N = 100$ . Assuming they are generated from

$$\text{logit}(\pi_{ij}) = \beta_i + \mathbf{x}_{ij}^T \boldsymbol{\alpha},$$

where  $\pi_{ij} = P(y_{ij} = 1)$ , the random effect  $\beta_i$  follows normal distribution with mean 0 and  $\sigma_b^2 = 1$ ;  $\mathbf{x}_{ij} = (x_{ij,1}, \dots, x_{ij,p})^T$  is the  $p$ -dimensional covariates. The first 25 covariates are assumed to be relevant covariates ( $q = 25$ ), which are weakly correlated with the remaining ( $p-25$ ) irrelevant covariates. The vector  $\mathbf{x}_{ij}$  is generated from a multi-normal distribution with mean 0 and covariance matrix

$$\Sigma = [\sigma_{kk'}^2] = \begin{cases} 1, & \text{for all } k = k' \\ 0.8^{|k-k'|}, & \text{for } k \neq k' \text{ and } (k, k') \in \{1, \dots, 25\} \text{ or } (k, k') \in \{26, \dots, p\} \\ 10^{-4}, & \text{otherwise} \end{cases} .$$

Note that the setup of covariance matrix highly influences variable selection consistency. To make PLMM able to achieve selection consistency, the covariates with zero coefficients need to be irrepresentable for the covariates with non-zero coefficients (Zhao and Yu, 2006). Thus, here we only consider weak correlations between the relevant and the irrelevant covariates. This covariance setup, in fact, had been discussed as reasonable in microarray data analysis in the sense that the genes that are correlated with the phenotype of interest and those that are not related to the phenotype may exist in different functional pathways (Bair et al., 2006; Huang et al., 2008). We denote the corresponding coefficients as  $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_p)^T$  and further differentiate the impact of the first 25 relevant covariates and the rest by

$$\alpha_k = \begin{cases} 2, & k = 1, \dots, 5 \\ 1, & k = 6, \dots, 15 \\ 0.5, & k = 16, \dots, 25 \\ 0, & k = 26, \dots, p \end{cases} .$$

Four different numbers of covariates,  $p = 200, 400, 3000$ , and  $5000$ , were considered in this simulation study while the total number of observations was fixed ( $N = 100$ ).

For each  $p$ , simulations were conducted based on 100 training samples and 100 testing samples. For both Lasso and adaptive Lasso (AdaLasso), tuning parameters ( $\lambda$ ) were determined based on 5-fold cross validation with the training dataset. After selecting the turning parameters, the PLMM models with both penalties were re-estimated for the training dataset. Then based on the PLMM estimates, the classification errors were calculated by the percentage of misclassified samples in the testing dataset.

Based on 500 simulations, comparisons of PLMM with Lasso and adaptive Lasso are summarized in Table 15. Here  $v_k = |\tilde{\alpha}_k|^{-1}$  is used in the adaptive Lasso penalty, and  $\tilde{\alpha}_k$  is the marginal estimate. “VC” in the third column lists the estimated variance component,  $\hat{\sigma}_b^2$ . The fifth column, ACI, represents the average number of relevant covariates (with non-zero coefficients  $\alpha_1, \dots, \alpha_{25}$ ) correctly identified by PLMM in 500 simulations, and  $\frac{\text{ACI}}{25}$  is the correct identification rate. Similarly,  $\text{ACI}_1$  in the sixth column is the average number of correct identification of the first fifteen covariates, where the coefficients  $\alpha_1, \dots, \alpha_{15}$  are comparably larger, and  $\frac{\text{ACI}_1}{15}$  is the correct identification rate. The seventh column, AMI, stands for the average number of the irrelevant covariates that are misspecified (with zero coefficients  $\alpha_{26}, \dots, \alpha_p$ ), and  $\frac{\text{AMI}}{p-25}$  is the variable misspecification rate. The last column represents the average classification error (ACE) with the standard deviation (SD) based on 500 simulations.

In general, both penalties performed well in the sense that the correct identification rate was high and the variable misspecification rate was low, as shown in Table 15. Even with a large number of covariates, PLMM still successfully identified more than 80% of the relevant covariates (non-zero coefficients). Furthermore, more than 98% of the first 15 covariates were correctly identified, and the average misspecification rate ( $\frac{\text{AMI}}{p-25}$ ) was less than 1%. Specifically, for each  $p$ , it can be seen from the fifth column that the Lasso penalty performed slightly better in identifying relevant covariates comparing to the adaptive Lasso penalty except for  $p = 3000$ . They worked almost equally well in identifying the first fifteen covarites (the sixth column,  $\text{ACI}_1$ ). Based on

the seventh column (AMI), however, the Lasso penalty resulted in more misspecified irrelevant covariates than that with the adaptive Lasso penalty. It appeared that adaptive Lasso tends to yield a smaller model and slightly higher prediction accuracy (i.e. smaller average classification error in the last column).

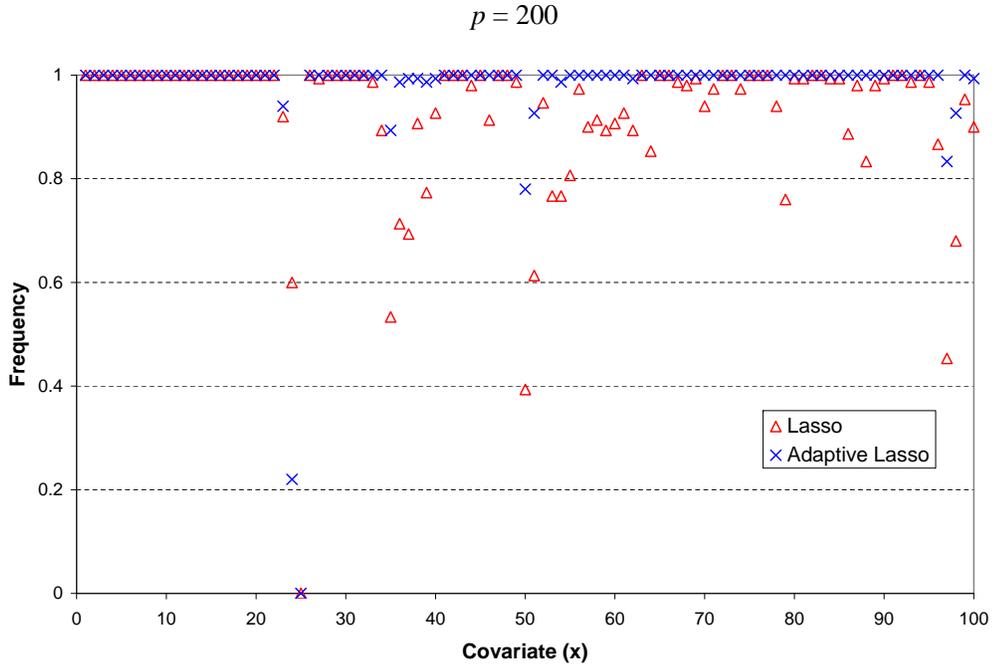
**Table 15:** Comparison of the Lasso and adaptive Lasso penalties in PLMM

PLMM	$p$	VC	$\lambda$	ACI ( $\frac{ACI}{25}$ )	ACI <sub>1</sub> ( $\frac{ACI_1}{15}$ )	AMI ( $\frac{AMI}{p-25}$ )	ACE (SD)
Lasso	200	0.4908	0.09	23.5 (0.940)	15 (1)	19.7 (0.113)	10.57% (2.25%)
AdaLasso	200	0.4884	0.05	23.2 (0.928)	15 (1)	4.9 (0.028)	9.55% (1.99%)
Lasso	400	0.4957	0.11	23.2 (0.928)	15 (1)	19.6 (0.052)	8.72% (2.47%)
AdaLasso	400	0.4818	0.06	23.1 (0.924)	15 (1)	11.5 (0.031)	7.47% (1.95%)
Lasso	3000	0.4901	0.13	20.0 (0.800)	14.7 (0.980)	29.7 (0.010)	12.77% (2.64%)
AdaLasso	3000	0.4882	0.07	20.0 (0.800)	14.9 (0.993)	13.5 (0.005)	11.17% (2.77%)
Lasso	5000	0.4868	0.14	23.2 (0.928)	14.8 (0.987)	29.2 (0.006)	7.32% (2.87%)
AdaLasso	5000	0.4973	0.10	22.9 (0.916)	14.8 (0.987)	9.2 (0.002)	6.00% (2.27%)

Another interesting observation from Table 15 is that PLMM with adaptive Lasso obtained a smaller increase in AMI than PLMM with Lasso when  $p$  increased. For example, AMI for adaptive Lasso was 4.9 when  $p = 200$  and 9.2 when  $p = 5000$ , while it increased from 19.7 to 29.2 for Lasso. It seems that the adaptive Lasso penalty can identify covariates more accurately when  $p$  increases than the Lasso penalty.

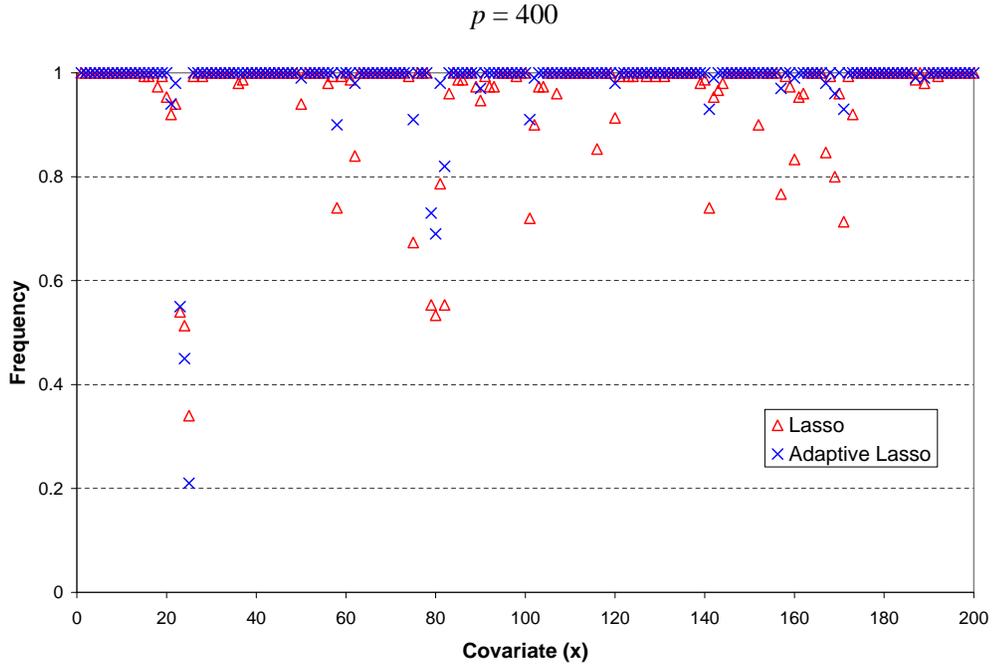
To better study the performance of PLMM, the frequencies of individual covariates being correctly identified from the 500 simulations are plotted in Figures 8 – 11 with different numbers of covariates ( $p = 200$ ,  $p = 400$ ,  $p = 3000$ ,  $p = 5000$ ). For a clear view, only half of the covariates are plotted; the first 25 of them are the relevant covariates and the rest are randomly chosen from the irrelevant covariates. The red triangle represents the frequency based on PLMM with the Lasso penalty while the blue cross represents that with the adaptive Lasso penalty. Both penalty functions can identify covariates successfully except those relevant covariates with comparably smaller coefficients ( $\alpha_{16}, \dots, \alpha_{25}$ ) when  $p$  goes large. Comparatively, the adaptive Lasso penalty has higher frequency of identifying the irrelevant covariates correctly

than the Lasso penalty.

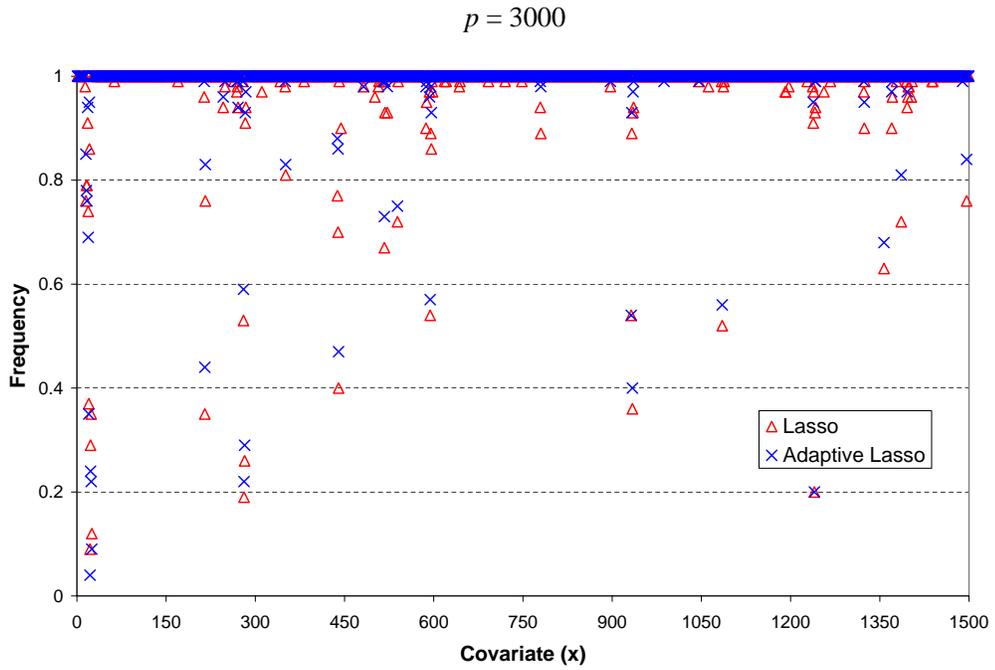


**Figure 8:** Comparison of PLMM with Lasso and adaptive Lasso ( $p = 200$ )

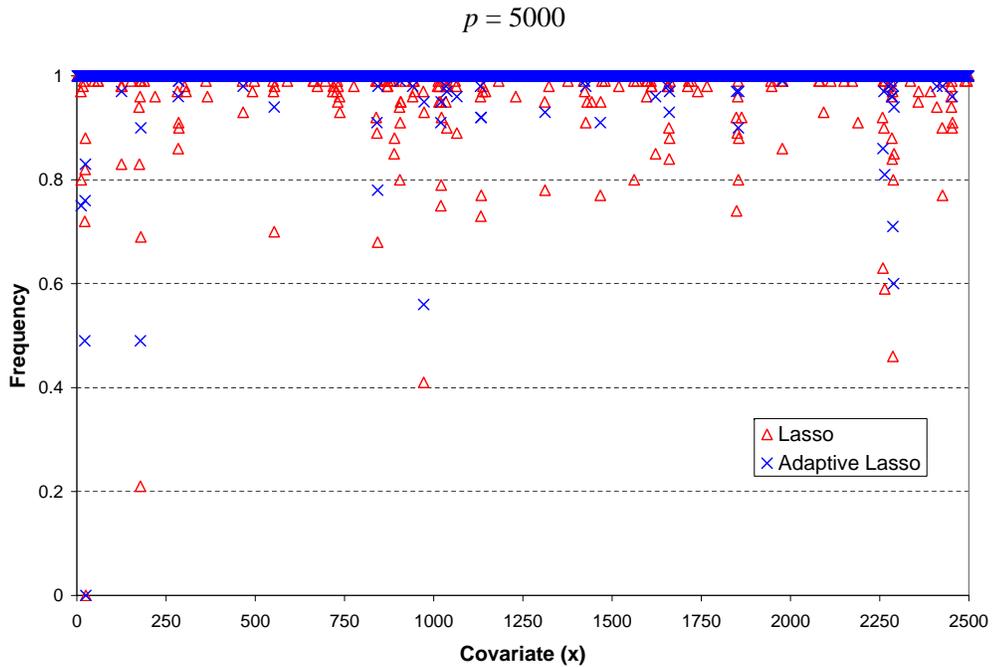
As for the computational speed of the proposed hybrid algorithm, we traced the computation time of each complete run, which includes a 5-fold cross validation procedure to choose the tuning parameter ( $\lambda$ ) from 11 different values and reports classification errors from 100 training and 100 testing samples. Tested on the computer equipped with the Intel Core2 Duo 2.66GHz CPU and 3GB RAM, the average computation time of PLMM with the Lasso penalty was 11 seconds when  $p = 200$  and 55 seconds when  $p = 5000$ . For PLMM with the adaptive Lasso penalty, the average computation time was 23 seconds when  $p = 200$  and 6.3 minutes when  $p = 5000$ .



**Figure 9:** Comparison of PLMM with Lasso and adaptive Lasso ( $p = 400$ )



**Figure 10:** Comparison of PLMM with Lasso and adaptive Lasso ( $p = 3000$ )



**Figure 11:** Comparison of PLMM with Lasso and adaptive Lasso ( $p = 5000$ )

#### 4.7 Application: Breast Cancer Study

The proposed method is applied to a microarray experiment conducted by Winship Cancer Institute (<http://www.cancer.emory.edu/>). In this experiment, tissue samples were extracted from breast cancer patients, and both biological and technical replicates were considered. There are 151 cleaned samples exported from 91 patients (biological replicates), and more than half of the patients have 2 to 4 technical replicates. Note that the number of technical replicates is not the same for each patient due to the limited availability of RNA derived from formalin-fixed paraffin-embedded (FFPE) tissues. Among the 91 patients, 21 of them were tested all negative (denoted by NNN) for three immunohistochemical (IHC) receptors (estrogen receptor, progesterone receptor, and human epidermal growth factor receptor 2) by which breast cancer subtypes are traditionally defined and 70 of them were tested positive (denoted by non-NNN) for at least one receptor. Note that NNN carcinomas are extremely

aggressive and cancer patients with this type of breast cancer tend to have poor outcome. The image fluorescent intensities of 1488 gene probes related to breast cancer were measured after running cDNA Annealing Selection extension and Ligation (DASL) experiment. The objective of this experiment is to classify the binary responses (NNN vs. non-NNN) based on the level of gene expression.

The penalized logistic mixed model (PLMM) was applied to analyze this experiment. To study its performance, we randomly selected 14 patients with NNN and 46 patients with non-NNN as a training set, while the remaining 7 patients with NNN and 24 patients with non-NNN were set aside as a blind testing set. This stratified sampling procedure was repeated 100 times. The average sample sizes of training and testing sets were 103 and 48, respectively.

The fitted PLMM based on the training set is

$$\text{logit}(\pi_{ij}) = \beta_i + \mathbf{x}_{ij}^T \boldsymbol{\alpha},$$

where  $i = 1, \dots, 60$ , for a given  $i$  the corresponding  $j$  ranges from 1 to 4,  $\pi_{ij} = P(y_{ij} = \text{NNN})$ ,  $\mathbf{x}_{ij} = (x_{ij,1}, \dots, x_{ij,p})^T$ , and  $p = 1488$ . The random effect is assumed to be normally distributed with mean 0 and variance  $\sigma_b^2$ . The estimated variance component ( $\hat{\sigma}_b^2$ ) is listed in Table 16.

**Table 16:** Comparison of the Lasso and adaptive Lasso penalties in PLMM and PLR in cancer study

Model	$\hat{\sigma}_b^2$	$\lambda$	Number of selected genes	Ave. classification error (standard deviation)
PLMM + Lasso	0.9667	0.22	12	15.92% (4.55%)
PLR + Lasso	–	0.23	12	16.48% (4.68%)
PLMM + AdaLasso	0.9763	1.14	5	14.86% (3.55%)
PLR + AdaLasso	–	1.16	5	16.01% (4.49%)

PLMM with two penalty functions, Lasso and adaptive Lasso (AdaLasso), were

compared in Table 16. The tuning parameters ( $\lambda$ ) were selected by 5-fold cross validation with the training sets. In this experiment, the average classification errors in PLMM with both penalties on the 100 testing datasets were smaller than 16%. With adaptive Lasso, PLMM even produced smaller models and slightly higher prediction accuracy in the testing sets than with Lasso. As a comparison, we applied a naive method, namely, the penalized logistic regression (PLR) without considering random effects. Similarly, both the Lasso and adaptive Lasso penalties were used for PLR. Table 16 shows that with the Lasso penalty, PLMM and PLR selected the same number of genes on the average, while PLMM yielded better prediction accuracy (i.e. the smaller classification error in the testing sets). With the adaptive Lasso penalty, PLMM also resulted in smaller classification errors than PLR with the same number of genes selected on the average. These results indicate that with the incorporation of the random effects, prediction can be more accurate and further inference can be made beyond the patients involved in this experiment.

#### ***4.8 Concluding Remarks***

Despite the prevalence of classification methods in the literature, there is no existing model readily applicable to classify correlated binary responses when the number of covariates is much larger than the sample size. To tackle this problem, a penalized logistic mixed model (PLMM) was proposed. By incorporating random effects, PLMM takes into account the correlations among the repeated observations from the same experimental subject and the heterogeneity among different experimental subjects. Thus, inferences can be made beyond the subjects involved in the experiment.

For PLMM, a new algorithm was introduced to estimate fixed effects and variance components. Theoretical properties regarding PLMM with the Lasso penalty were also addressed, which showed that PLMM can estimate the correct signs consistently. Finite sample performance was then examined via a simulation study.

The proposed method was applied to a gene expression microarray experiment for breast cancer study, where part of RNA samples were collected with correlations for they came from the same patient. This correlation structure can be captured by PLMM, and thus it resulted in better prediction accuracy than assuming independence. Motivated but not limited to microarray gene expression experiments, the proposed modeling method can be applied to other high-dimension low-sample-size classification problems with correlated samples.

## CHAPTER V

### HIERARCHICAL ATTRIBUTE-BASED FORECASTING

In this chapter, we propose a new forecasting approach that deals with a large number of categories in predictors and takes into account the predictor structure. The new approach is motivated by a capacity forecasting problem in the air cargo industry. We first introduce this forecasting problem and describe its characteristics in the first section.

#### *5.1 Introduction*

With the continuing globalization and regional specialization of industry, the world air cargo traffic is predicted to expand at an average annual rate of 5.8% for the next two decades, tripling current traffic levels (Boeing, 2008). As cargo traffic typically grows faster than passenger traffic (Airbus, 2009), many passenger airlines have converted from pure passenger carriers to combination carriers that carry both passengers and cargo. For the combination carriers, cargo is carried to the belly space in aircraft after passenger baggage. Due to passenger priority, the combination carriers do not know how much capacity they have available for free sale until shortly before flight departure. As the free-to-sell capacity is constrained by passenger baggage requirements, an accurate capacity forecast of *passenger baggage* before cargo capacity booking proceeds is crucial to combination carriers' cargo business and operations.

Below we describe the characteristics of this capacity forecasting problem in details:

- On any given day, a large U.S. commercial carrier typically operates more than a thousand of flights on hundreds of international and domestic routes.

A route is defined by a pair of origin and destination airports (O-D pair). Note that traffic flow is directional, which means route A-B is different from route B-A. Among hundreds of routes, some routes are more popular and have more flights than others during a period of time. For example, a popular domestic route may be served 12 flights per day while an international route departing from the same origin airport may be served only one flight per week. Hence, the daily airline operations data are commonly observed with *high volume, unbalanced frequency, and a large number of origin-destination pairs and flight numbers*.

- Cargo capacity is typically measured in two dimensions – weight and volume. But due to the nature of passenger data collected in airlines, only passenger baggage weight is available while volume is not. Thus, in this capacity forecasting problem, the dependent variable we consider is the *average baggage weight per passenger* on each flight leg. Here a flight leg comprises a pair of origin and destination airports as well as a flight number.
- In the consideration of passenger behaviors and baggage policies, passenger baggage weight is expected to be affected by several factors, such as geographical regions, airport cities, departure dates and time, and flight numbers. Note that these factors are not simply cross over to each other, but instead following relationships exist between these factors: Airports are nested in regions because each airport is conjunction with only one geographical region. For a similar reason, the flight number of a single-leg flight (e.g., route A-B) is nested in an origin-destination pair. However, for a multiple-leg flight (e.g., route A-B-C), the flight number is no longer nested in an O-D pair because every flight number is shared among more than one O-D pair. Thus, flight numbers are considered to be *partially nested* in O-D pairs.

- Flying routes and schedules are not always the same over time. In fact, the number of scheduled flights is growing. There could be tens to hundreds of *new flights* that were not shown in history on a weekly basis. A new flight may appear in several different ways: being assigned with a new flight number, serving a new origin or destination airport, or operating on a new route.

All in all, this forecasting problem has following data characteristics: a large sample size (in tens of thousands); a large number (thousand) of categorical values in predictors; a wide, varied number of samples in each predictor category; nested predictor structures; and in the presence of new forecasting units (i.e., new flights). The objective of this study is to forecast baggage weight for every single future flight, including new flights.

Despite a variety of methods that have been developed for predictive modeling (Hastie et al., 2008), there are no existing methods completely suitable for this forecasting problem. Given that the dataset is large and complex in terms of the number of observations and the number of categories in prediction variables, *computational considerations* will play an important role in this forecasting problem. This also makes computationally intensive algorithms impractical. In addition, this forecasting problem requires *interpretable* forecasting results for a practical use; simply producing forecasts is not enough. Thus, black-box methods, such as neural networks (McCulloch and Pitts, 1943), will become less useful in this problem even if it has high predictive power. Hastie et al. (2008, p.351) summarized the characteristics of different predictive methods, including neural nets, tree methods, and kernel methods. Among them, tree-based methods are considered to be the most favorable to this forecasting problem for it is relatively efficient and interpretable and also for it can easily handle interactions between predictors. However, conventional tree-based methods, such as CART (Breiman et al., 1984) and CHAID (Kass, 1980), are not generally

considered efficient in the presence of a large number of categorical values in predictors. Moreover, they are not able to take into account the intrinsic nested predictor structures and consequently not adequate to forecast new flights in this study.

In this study, we propose a new forecasting approach, named hierarchical attribute-based forecasting (HABF). The new approach is developed for the situation where a forecasting problem is associated with a large number of categories in predictors and with some observed predictor structures. Similar to conventional tree-based methods, HABF adopts piecewise constant prediction models (i.e., using the sample mean of a node), whereas HABF is different from conventional tree-based methods in two aspects. First, to incorporate the nested predictor structures, HABF first selects a set of significant predictors through statistical significance tests with structure settings and then orders the selected predictors by their importance. Based on the predictor order, a hierarchy of predictors is built and used for splitting. Second, in order to have HABF competent in efficiency in the presence of a large number of categorical values in predictors, HABF does not adopt recursive partitioning or an exhaustive search to find a splitting value. It simplifies this step by a series of complete multi-way hierarchical splits without merging categories. As a result, not only is the new approach more efficient than conventional tree-based methods, but also it makes forecasts fully interpretable and enhances new flight forecasting in this study.

Although the new forecasting approach is motivated by a capacity forecasting problem in the air cargo industry, the generic characteristics of this forecasting problem are also observed in other applications. For example, the forecasting problems in the airline and transportation industries are typically associated with a large number of origin-destination pairs. Apart from passenger baggage weight forecasting, the proposed approach can be used to forecast other quantitative values, such as cargo no show rate and passenger booking rate. In addition, in the hospitality industry,

detailed customer arrivals with duration of stay are key forecasts in a revenue management system (Weatherford and Kimes, 2003). For a large hotel chain, the number of forecasting categories can reach more than ten thousands, so efficiency is of the essence. Moreover, in the retail industry, not only is the number of products at the stock-keeping-unit level expected to be very large, but also some structures exist in the forecasting dimensions of product line, location, and time (Han and Lam, 2007). As the above-mentioned forecasting problems have similar data characteristics to this study, the proposed approach can be applied to these forecasting problems as well.

The organization of the remaining sections in this chapter is as follows. We examine existing tree-based methods in Section 2 and describe the new approach in Section 3. In Section 4, we evaluate the forecasting performance of the new approach and compare with several tree-based methods through empirical studies and sensitivity analyses. In Section 5, we conduct a few simulation experiments to further investigate the performance of the proposed approach under different circumstances. Section 6 provides a concluding remark of this study and a discussion of future research directions and applications.

## ***5.2 Review of Existing Tree-based Methods***

Tree-based methods are one of popular prediction methods for exploratory study and data mining (Hastie et al., 2008). They can be applied to datasets having both a large number of samples and variables. They also have ability to deal with irrelevant inputs, and they are resistant to outliers. In practice, tree-based methods are especially appealing as they can provide interpretable rules in visual representation. A tree is typically shown growing upside down, beginning at its root. An observation passes down the tree through a series of splits or nodes. Finally, a terminal node or leaf is reached and a predicted value is obtained.

A typical tree-based method has three major tasks: (1) How to split the data at

each node, including the identification of a splitting variable and a splitting value? (2) When to stop growing a tree? How to control the size of a tree? (3) How to predict the value of a response at each terminal node? Regarding the first task, data are often split via either binary or non-binary recursive partitioning. The term binary refers to a parent node is always split into exactly two child nodes. The term recursive indicates each child node, in turn, becomes a parent node, unless it is a terminal node. To find a splitting variable and a splitting value, most tree-based methods employ a univariate split by an exhaustive search that optimizes a node impurity at each node. The second task is often achieved by using a stopping rule and/or via a pruning process. For the third task, prediction is often given by the most frequent class at the terminal node for a classification tree or the mean of observations at the terminal node for a regression tree.

There are many existing tree-based methods in the literature (Sutton, 2005; Loh, 2008). Below we introduce three well-known tree-based methods that have ability to predict a continuous dependent variable. We also compare their differences and the capability of handling a large number of predictor categories in the last subsection.

### **5.2.1 CART**

CART stands for classification and regression trees, which was developed by Breiman et al. (1984). CART recursively creates binary splits on categorical or continuous independent variables through an exhaustive search. The exhaustive search algorithm searches all the independent variables and all the possible values for each independent variable to obtain the optimal split that maximizes the reduction in some impurity function. For a categorical dependent variable, impurity is measured by the Gini index, the entropy index, or twoing; for a continuous dependent variable, impurity is measured by the sum of squared errors or the sum of absolute deviations from the median.

One significant contribution of CART is the way to control the size of a tree. Besides employing a stopping rule, such as the minimal size of a node, CART introduces retrospective cost-complexity pruning. That is, CART generates a sequence of subtrees by first growing a large tree and then pruning it back until only the root node is left. In the pruning process, it uses testing samples or cross validation to estimate either the misclassification rate or the sum of squared errors as the cost of each subtree and chooses the one with the lowest estimated cost as the final tree. Then the predicted value at a terminal node is given by the major class or the sample mean.

There are several issues regarding CART splitting. Notice that the exhaustive search used in CART requires the evaluation of all possible splits on each predictor variable, so *computation time* would be a great concern. For a categorical predictor with  $M$  distinct values present at a node, the number of possible binary splits is  $2^{M-1} - 1$ , growing exponentially with  $M$ . For a continuous predictor with  $M$  distinct values present at a node, the number of possible binary splits is  $M - 1$ . Thus, it can be seen that the number of possible binary splits that has to be examined and the associated computational efforts in CART become very large when there are large numbers of predictors with many distinct values. In addition to the large amount of computation time, the exhaustive search used in CART has also been demonstrated to have *selection bias* toward variables that allow more splits, in particular categorical variables with many distinct values (Loh, 2002). For example, a variable with 20 categories ( $2^{19} - 1 = 524,287$  splits) is preferred 35 thousand times more often than a variable with 5 categories ( $2^4 - 1 = 15$  splits). Moreover, even though recursive partitioning in CART is done with a computationally-intensive exhaustive search, this does not guarantee to find a *global optimal* tree because it only focuses on optimizing individual splits and pays no attention to the quality of the entire tree. Although this issue may be diminished by using look-ahead splits (Ragavan and Rendell, 1993), it

was, on the other hand, commented by other research that look-ahead requires even more computation time and does not help much in prediction (Murthy, 1998).

### 5.2.2 CHAID

Kass (1980) proposed a tree method, called Chi-squared Automatic Interaction Detector (CHAID), for the detection of interactions in categorical data. The original CHAID method requires both dependent and independent variables to be categorical. Continuous independent variables would be grouped into a number of categories with an equal number of observations. Later, CHAID was extended to handle categorical, ordinal, and continuous dependent variables by Magidson (1993) and SPSS Inc. (1999).

CHAID incorporates a sequential merge and split procedure. Within each predictor, the pair of predictor categories that is least significantly different with respect to the dependent variable would be merged based on the Pearson chi-square test for the categorical dependent variable or based on the F test for the continuous dependent variable. This merging procedure would be repeated within each predictor until all pairs of (merged) categories are significantly different with respect to the dependent variable. After this merging procedure completes cycling through all predictors, the predictor with the smallest Bonferroni adjusted p-value for the set of significant categories is then selected as the splitting variable and exactly one branch is set for each significant category. Continue this process until the smallest Bonferroni adjusted p-value of any predictor is greater than some threshold value; then no further splits would be performed and terminal nodes are formed by the average or the majority of the dependent variable. In this process, note that once a node or a predictor category is made by CHAID, no further evaluation would be considered. Thus, CHAID is regarded as a “forward” sequential tree method, in which no pruning procedure is performed.

Unlike CART and other tree methods, CHAID generates multiple-split trees and tends to be wider. Although every multi-way split can be expressed by a series of binary splits, the feature of multi-way splits has made CHAID been popular in marketing research applications, especially market segmentation studies (Chen, 2003; Magidson, 1994; Ratner, 2003), because this type of display matches the requirements of market segments (Hill and Lewicki, 2006).

As Kass was concerned about computation time when analyzing a large dataset, CHAID does not search for all possible combinations of the categories. Rather, it settles for the last split on a predictor while it may not be the most significant split or the best split. Biggs et al. (1991) introduced a modification of CHAID, named exhaustive CHAID, by replacing the last split with the most significant split among all possible category subsets. However, it requires more computation time and is likely to become intractable when dealing with a large dataset with a large number of categories.

### 5.2.3 GUIDE

Loh (2002) presented a new tree algorithm called GUIDE (for generalized, unbiased, interaction detection and estimation) for building piecewise constant and linear regression models. It avoids the selection bias and the computational problems of categorical variables in CART by selecting the splitting variable and the splitting value separately. GUIDE provides many choices in prediction. It can construct piecewise-constant, multiple linear, and simple polynomial tree models for least-square, quantile, Poisson, and proportional hazards regression (Loh, 2002).

Below we use the piecewise constant model to illustrate the GUIDE splitting algorithm. At each node, the sample mean is fitted and the residuals are computed. Then the observations at a node are divided into two groups, with one group for all the positive residuals and the other group for all the non-positive residuals. The idea is

to detect non-random patterns in the two groups of signed residuals. Two main tests are used for selecting a variable in GUIDE. One is a curvature test and the other is an interaction test. In the curvature test, for a variable with  $M$  categories, the Pearson chi-square test is applied to the count of observations in each cell of the two-by- $M$  contingency table, which is formed by the two groups of residuals as rows and  $M$  categories as columns. For a continuous variable, its numerical values are divided into four groups at the sample quantiles to construct a two-by-four contingency table where the Pearson chi-square test is applied. In the interaction test, the space of a pair of continuous variables is divided into four quadrants at each sample median. A two-by-four contingency table is then constructed with the residual signs as rows and the four quadrants as columns; then the Pearson chi-square test is applied to this table. For a pair of two categorical variables, a similar contingency table is formed by two rows and the number of columns is equal to the product of the numbers of categories in a pair of two variables. For a pair of one continuous and one  $M$ -category variable, the Pearson chi-square test is applied to a contingency table with two rows and  $4M$  columns. Repeat the curvature test for each variable and the interaction test for each pair of variables. If the smallest p-value of the Pearson chi-square test is from a curvature test, then the associated variable is selected. Otherwise, if the smallest p-value is from an interaction test and at least one of the two variables is categorical, then the one with the smaller curvature p-value is selected; if both variables are continuous, then the one with the smallest total sum of squared errors is selected. After a splitting variable is identified, a splitting value is then selected by either a greedy search or a sample median. After a large tree is constructed, the tree is pruned with the cross validation method as in CART.

The contributions of GUIDE are in two aspects. First, the two-step splitting process makes GUIDE free from the selection bias as in CART. This could be more

manifest to a variable with a large number of distinct variables. Second, the two different chi-square tests in GUIDE make a tree sensitive to curvature and local pairwise interactions between predictor variables. In addition, Loh (2002) commented that using a sample median as the splitting value is not always inferior to a greedy search. Based on his study, the performance of these two methods is actually even in times. The strengths of variable selection features in GUIDE have also been demonstrated in recent transportation research (Qin and Han, 2008).

#### 5.2.4 Summary

Among the three well-known tree-based methods introduced previously in this section, their differences can be categorized in three aspects: splitting, stopping/pruning, and prediction. In splitting, only CART adopts an exhaustive search while CHAID and GUIDE utilize statistical tests; CART and GUIDE simply generate binary splits while CHAID uses multi-way splits and merges categories in advance. Regarding stopping and pruning, besides the stopping rules utilized in all these methods, GUIDE uses the same pruning method as CART while pruning is not applicable in CHAID. For prediction, CART and CHAID builds piecewise constant models while GUIDE has more prediction model choices.

Next we compare the capability of these well-known tree-based methods when dealing with a large dataset with a large number of categories in predictors. As explained earlier in Section 5.2.1, CART would be computationally intractable in this situation due to the extremely large number of splits and also the selection bias. In this regard, CHAID and GUIDE would be more suitable because they do not adopt an exhaustive search algorithm and are likely to have better computational performance. One advantage of CHAID over GUIDE is its interpretability since CHAID uses multi-way splits. This offers CHAID efficiency in interpretation and presentation and also better suits for splitting a node by a variable with many categories.

However, CHAID features a procedure of merging categories. This may deteriorate its interpretability because a merged category is often harder to be interpreted than an individual one. Moreover, merging categories would require significant amount of computation time when the number of categories is large. Thus, the computational efficiency and interpretability of CHAID would be main concerns.

On the top of the above comparisons, it is more concerned that none of these tree-based methods can take into account the nested predictor structures, which is commonly observed in transportation industries. Hence, we comment that the existing tree-based methods are not completely suitable for the forecasting problem in this study as the forecasting problem is associated with a large number of categories in predictors and with some predictor structures. Accordingly, a more suitable forecasting method is needed for the capacity forecasting problem in the air cargo industry.

### ***5.3 Hierarchical Attribute-based Forecasting***

In this section, we describe the proposed new forecasting approach, hierarchical attribute-based forecasting (HABF), for the situation where predictors are with a large number of categories and with some observed structures. HABF, similar to tree-based methods, adopts piecewise constant prediction models (i.e., using the sample mean of a node), whereas HABF is different from existing tree-based methods in two tasks: splitting and stopping/pruning. We will elaborate the differences and the new features in the first two subsections. Then a comprehensive comparison of HABF and existing tree-based methods will be presented toward the end of this section.

#### **5.3.1 Hierarchical Splitting**

The first, also the most important step in building a predictive tree is partitioning samples into several branches. Most tree-based methods employ *recursive partitioning* to select splitting variables and values. However, this type of partitioning does not

take into account any observed structures of predictor variables as each variable is individually treated. To incorporate the predictor structures, HABF considers all predictor variables together, instead of one at a time, through a general linear model with structure settings. Then an analysis of variance, F-test, is utilized to identify significant variables and rank these variables by the level of significance. The next step is to build a predictor hierarchy which consists of the ordered and selected variables. The predictor hierarchy is then used for splitting, with one branch for each categorical value of a predictor. If a continuous predictor is involved, then a node is split into two branches at the median value of the predictor, which is also adopted as one of the two splitting ways in GUIDE. Finally, a large tree with multi-way hierarchical splits is generated. We refer to the above procedure as *hierarchical splitting*. Below we elaborate the features in this new approach.

The new approach features multi-way, non-binary splits. Although binary splitting is used in many popular tree-based methods (e.g., CART and GUIDE); however, binary splitting is prone to produce a deep tree, which may cause difficulties in comprehension because the brain needs to keep track of many levels of conditioning. In contrast, multi-way splits can save the levels of a tree and become more compact. Thus, from the interpretation perspective, *multi-way splitting* is more favorable and is featured in HABF.

In spite of the fact that the new approach is targeted to deal with datasets with a large number of categories in predictors and a large sample size, without merging categories a large tree with considerable nodes is likely to be created. However, when the number of categories is large, a merging process would take longer time and need significant amount of computational efforts. On the other hand, when the sample size is large, as long as there are moderate samples in a category, leaving each category alone would not necessarily cause damage to prediction performance. Moreover, a merged category is often harder to be interpreted than an individual one. On account

of these concerns, the gains from merging categories are not evident. As we want to create an efficient and fully interpretable prediction method in the presence of a large number of categories in predictors and a large sample size, the new approach does not feature a procedure of *merging categories* as CHAID.

### 5.3.2 Stopping and Pruning

The second task in tree growing is to control the size of a tree. If a tree is too small, it may not describe data well. In contrast, if a tree has too many nodes with too few observations, the prediction may not be reliable. Typical tree-based methods contain one or more stopping parameters, such as the maximal depth of a tree, the minimal sizes of parent and child nodes, to control the size of a tree. Most tree-based methods also allow users to freely specify any thresholds for stopping. However, an appropriate threshold for a stopping rule is unknowable before a tree starts growing. If a stopping rule is set too conservative, a tree may stop growing too early and miss detailed branches. The alternative to initially setting stopping thresholds is to grow a large tree and then prune the tree back to a smaller size. The latter is generally considered more favorable for it avoids an inappropriate threshold that limits tree growing and influences prediction performance.

Instead of specifying an arbitrary stopping threshold before growing a tree, HABF determines the minimal number of observations at a node through statistical tests after a large tree is generated. In this process, HABF measures the impurity of a node by the mean-variance idea. The impurity of a node is measured by the *coefficient of variation (COV)*. The original COV is defined as the ratio of the standard deviation to the mean, which aims to describe the dispersion of a variable in a way that does not depend on the measurement unit. In tree applications, we can modify COV to be the ratio of the root mean squared error (RMSE) to the mean at a node to describe the goodness-of-fit and also keep the unitless property. The lower the COV is, the

smaller the residuals are relative to the predicted value under the piecewise constant prediction model.

Below we describe the process of determining the minimal number of observations in a node: The modified COV's are measured within each node. The decision variables are the minimal number of observations ( $MinObs$ ) of a node at each level of the hierarchy. Under the threshold of  $MinObs$  at a level, nodes with the number of observations smaller than  $MinObs$  would be neglected. The COV's in each of the remaining nodes and the corresponding COV's in the parent nodes form paired samples. The COV pairs from two adjacent hierarchical levels are then tested by Wilcoxon Signed Rank Test (Wilcoxon, 1945), a nonparametric test for differences between paired samples, to see whether the COV's from the lower level are statistically smaller than the COV's from the higher level. The hypothesis tests start from the lowest level of the hierarchy and the maximum of  $\{2, \text{the smallest number of observations in a node at the targeted level}\}$ . If the above hypothesized statement is accepted, then such  $MinObs$  value is used to prune the nodes without enough observations at the targeted level and beneath, and then continue to identify the  $MinObs$  at one level up. If not, then move to the next discrete number of  $MinObs$  at the same level and redo testing until a significant difference is found or the maximal  $MinObs$  is reached at the targeted level. Once  $MinObs$ 's for each level are identified, the pruned tree is obtained.

This process can be presented in the following pseudo code with the outputs of  $MinObs$ 's for each hierarchical level.

- Set  $K = k$ , where  $k$  is the number of hierarchical levels.
- $S^{(K)}$ : a set of distinct numbers of observations within the nodes at the  $K^{th}$  hierarchical level. Let  $m = \max \{2, \min S^{(K)}\}$ .
- $cov$ : the modified coefficient of variation within each node.

- $H_0^{(K)} : \mathbf{COV}_m^{(K)} \geq \mathbf{COV}_m^{(K-1)}$  vs.  $H_1^{(K)} : \mathbf{COV}_m^{(K)} < \mathbf{COV}_m^{(K-1)}$

where  $\mathbf{COV}_m^{(K)}$  is the *cov*-mean or *cov*-median over the nodes with the number of observations  $\geq m$  at the  $K^{\text{th}}$  hierarchical level.  $\mathbf{COV}_m^{(K-1)}$  is the corresponding *cov*-mean or *cov*-median at the  $(K-1)^{\text{th}}$  hierarchical level.

- If  $H_0^{(K)}$  is rejected by a Wilcoxon Signed Rank Test,

$$T_m^K = 2 \sum_{i=1}^{n_m^{(K)}} \text{rank}(|D_i|) \cdot I(D_i > 0) - \frac{n_m^{(K)}(n_m^{(K)} + 1)}{2}$$

$$\xrightarrow{d} \mathcal{N}\left(0, \frac{n_m^{(K)}(n_m^{(K)} + 1)(2n_m^{(K)} + 1)}{6}\right),$$

where  $D_i = \text{cov}_{m,i}^{(K)} - \text{cov}_{m,i}^{(K-1)}$  and  $n_m^{(K)}$  is the number of nodes with the number of observations  $\geq m$  at the  $K^{\text{th}}$  hierarchical level, then  $\text{MinObs}^{(K)} = m$  and  $K = K - 1$ ; update  $S^{(K)}$  and  $m$ .

- If  $H_0^{(K)}$  is not rejected and  $m < \max S^{(K)}$ , then  $S^{(K)} = S^{(K)} \setminus (1 \cup m)$  and update  $m$ .
- If  $H_0^{(K)}$  is not rejected and  $m = \max S^{(K)}$ , then  $\text{MinObs}^{(K)} = m$ .
- Repeat the above testing until all  $\text{MinObs}^{(K)}$  for  $K = 2, \dots, k$  are identified.

### 5.3.3 Interpretability

Although HABF is likely to generate a large tree by the nature of hierarchical splits, it always possesses simple interpretability. Regardless of the tree size, the prediction from HABF can be easily explained in the following way: Assuming there are  $k$  significant predictor variables with respect to a dependent variable. If there are enough training samples that have the exact same set of  $k$  predictor values as of a testing sample, then the predicted value of a testing sample is based on and is explained by the training samples that have the exact same  $k$  predictor values. In contrast, if the number of training samples that have the exact same set of  $k$  predictor

**Table 17:** Comparison of tree-based methods

<b>Method</b> <b>Task</b>	<b>CART</b>	<b>CHAID</b>	<b>GUIDE</b>	<b>HABF</b>
<b>Splitting – variable selection</b>	exhaustive search for splitting variables and splitting values	chi-square or F tests for interaction and dependence	chi-square tests for interaction and curvature detection	F tests for ranking variables by importance
<b>Splitting – number of branches</b>	binary split by greedy search	multi-way split with merged categories	binary split by greedy search or sample median	multi-way split with each category or binary split by sample median
<b>Stopping rule and threshold</b>	pre-specified minimal number of observations in a node	pre-specified minimal number of observations in a node	pre-specified minimal number of observations in a node	post-identified minimal number of observations by hypothesis tests
<b>Pruning</b>	cross validation on mean squared error of subtrees	not applicable	cross validation on mean squared error of subtrees	hypothesis tests on coefficient of variation
<b>Prediction</b>	piecewise constant	piecewise constant	piecewise constant or linear regression	piecewise constant

values is smaller than the  $k^{th}$  threshold of *MinObs* but having the same set of  $k - 1$  predictor values is larger than  $(k - d)^{th}$  threshold of *MinObs*, where  $d$  is an integer starting from 1, then the predicted value of a testing sample is based on and is explained by the training samples that have the exact same  $k - d$  predictor values in order (i.e., the last  $d$  ordered predictors are dropped from prediction).

### 5.3.4 Comparison of Tree-based methods

Previously in Section 5.2.4, we summarized the similarities and the differences between three well-known tree-based methods (CART, CHAID, and GUIDE). Here we recap their characteristics and extend the comparisons to the new approach. Table 17 lists the important components of HABF and three well-known tree-based methods regarding five major tasks.

Among these methods, CHAID is the one closest to HABF. In the following, we give a closer look at the comparisons between CHAID and HABF. Both CHAID and

**Table 18:** Differences between CHAID and HABF

Differences \ Method		CHAID	HABF
<b>Technical</b>	<b>Categorical predictor</b>	merge indifferent categories	leave each category alone
	<b>Variable selection</b>	select the most significant variable recursively from F tests	select variables by the order of importance as a hierarchy
	<b>Min size of a node</b>	initially specified	determined by hypothesis testing
<b>Performance</b>	<b>Efficiency</b>	medium	high
	<b>Interpretability</b>	medium	high

HABF generate prediction by multi-way splits under the piecewise prediction model; however, they have three major technical differences. First, CHAID merges indifferent categories for each predictor variable before searching for a splitting variable while HABF leaves each category alone without merging. Second, CHAID adopts recursive partitioning while HABF features hierarchical splitting. Third, CHAID requires an initial setting of stopping rules before growing a tree while HABF determines appropriate thresholds after growing a large tree. Clearly, except for the third technical difference, HABF requires much less computation than CHAID, and therefore it would be more efficient to deal with large datasets having large numbers of samples and predictor categories. In addition, prediction results from HABF are expected to have better interpretability than CHAID for a merged category often needs more efforts in interpretation than individual ones. The summary of these differences is drawn in Table 18.

**Table 19:** Data attributes and categories

<b>Attributes</b>	<b>Number of categories</b>
Origin region (OR)	4
Destination region (DR)	4
Origin airport (OA)	146
Destination airport (DA)	141
Departure day of week (DOW)	7
Flight number (FLT)	1145

## 5.4 *Application*

In this section, we apply the proposed approach to a capacity forecasting problem in the air cargo industry. In the first part, we evaluate the forecasting performance of the new approach and compare it against three well-known tree-based methods through an empirical study. In the second part, we conduct more analyses to understand how forecasting accuracy would be affected in different situations.

### 5.4.1 Capacity Forecasting in the Air Cargo Industry

In this empirical study, one-year daily operations data were collected from one of top five passenger airlines in the world (IATA, 2010). The dataset contains baggage weight and passenger numbers on 344 thousands of flights in 2009 with 6 flight attributes (origin region, destination region, origin airport, destination airport, departure date, and flight number). The objective is to forecast baggage weight per passenger for every single future flight, including new flights that were not shown in history.

In the first evaluation, we extracted 8-week flights from October, 1, 2009 to November 25, 2009 as training samples and 2-week flights from November 26, 2009 to December 9, 2009 as testing samples. The number of observations in the training and testing samples are approximately 48 thousands and 12 thousands, respectively. The number of categories in each flight attribute is shown in Table 19.

The new approach was implemented in R version 2.10.0. The first step of HABF

**Table 20:** ANOVA table

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	F ratio
ORIG_REGION	3	385727	128576	8753.03	< 2.2E-16	3359.96
DEST_REGION	3	2168031	722677	49197.61	< 2.2E-16	18885.12
ORIG_AIRPORT   ORIG_REGION	138	243157	1762	119.95	< 2.2E-16	99.44
DEST_AIRPORT   DEST_REGION	133	1127902	8480	577.32	< 2.2E-16	477.02
DOW	6	71113	11852	806.86	< 2.2E-16	384.44
FLIGHT	1016	236647	233	15.86	< 2.2E-16	14.75
RESIDUALS	46682	685725	15			

is to rank attributes and build a hierarchy for splitting. From the ANOVA table (Table 20), we can see that all the six attributes were identified highly significant in terms of very small p-values to the dependent variable (i.e., baggage weight per passenger) with an adjusted  $R^2 = 0.86$  in the general linear model. Although all attributes are significant, their p-values from this table are not distinguishable for variable ranking, so a proxy measure, *Fratio*, is developed to take the place. The F ratio is defined as the F-test statistic (F value) over the critical value with respect to a significance level (e.g., 0.05) and a pair of the degrees of freedom that was used in the F-test statistic. For example, the F ratio of ORIG\_REGION, was calculated by  $8753.03/F_{0.95}(3, 46682) = 8753.03/2.6051 = 3359.96$ . Reading F ratios (the last column of Table 20) in a descending order, we are able to construct a six-level hierarchy. From top to bottom, these six levels are destination region (DR), origin region (OR), destination airport (DA), day of week (DOW), origin airport (OA), and flight number (FLT). This hierarchical order illustrates three things: (1) regions have more significant discrepancy than airports; (2) destinations are more influential on passenger baggage weight than origins; (3) baggage weights distribute differently across different days of week and flights.

Based on the built hierarchy and hierarchical splitting, a large tree with six layers was obtained. HABF then looks for an appropriate value of *MinObs* at each level of the hierarchy by the method discussed in Section 5.3.2. The results show that COV's were improved significantly down the tree even if there are only two observations in

a node. In other words, only nodes that are composed of single observation were pruned. The total number of nodes constructed by HABF was 6984, which yielded a mean absolute percentage error (MAPE) of 13.6% for the testing samples.

To compare the forecasting performance of HABF against other tree-based methods (i.e., CART, CHAID, and GUIDE), we took the same training and testing samples as the above analysis. We used the SPSS 16.0 software package to build CART and CHAID trees. GUIDE trees were obtained by Loh's GUIDE program, which can be downloaded from the author's website (<http://www.stat.wisc.edu/~loh/guide.html>). For each compared tree method, two runs with different settings were carried out. We use subscript 1 to indicate trees were built with the default parameter values of the tree method and use subscript 2 to indicate trees were built with the parameter values that were similar to those used in HABF. Specifically, the minimal number of observations was changed to a smaller value: 2 for CART and CHAID; 3 (the smallest number allowed) for GUIDE. Other parameters were set to the default values unless stated otherwise.

Table 21 summarizes the training and testing errors (measured by MAPE), the number of nodes, and computation time in CART, CHAID, GUIDE, and HABF. First, we focus on the comparisons between the two runs within each method. The results show that the use of an appropriate stopping threshold identified by HABF did improve forecasting accuracy and would be preferable to the use of the default stopping threshold. Second, among these tree-based methods, HABF performed relatively higher forecasting accuracy (i.e., smaller testing error) and higher computational efficiency (i.e., shorter run time) in this study. CHAID<sub>2</sub> has comparable forecasting accuracy to HABF, but it is not as efficient as HABF in terms of run time. GUIDE<sub>2</sub> follows this ranking of forecasting accuracy, but its computation time - almost two hours - makes it impracticable. CART trees produced the least favorable forecasting performance in this study.

**Table 21:** Forecasting performance of different tree-based methods

<b>Method</b>	<b>Training error</b>	<b>Testing error</b>	<b># of nodes</b>	<b>Run time (mins)</b>
CART <sub>1</sub>	18.5%	18.6%	47	28
CART <sub>2</sub>	18.5%	18.6%	51	28
CHAID <sub>1</sub>	17.4%	17.4%	103	14
CHAID <sub>2</sub>	11.9%	13.8%	765	20
GUIDE <sub>1</sub>	15.7%	15.7%	59	2
GUIDE <sub>2</sub>	14.0%	14.7%	327	115
<b>HABF</b>	<b>11.0%</b>	<b>13.6%</b>	<b>6984</b>	<b>2</b>

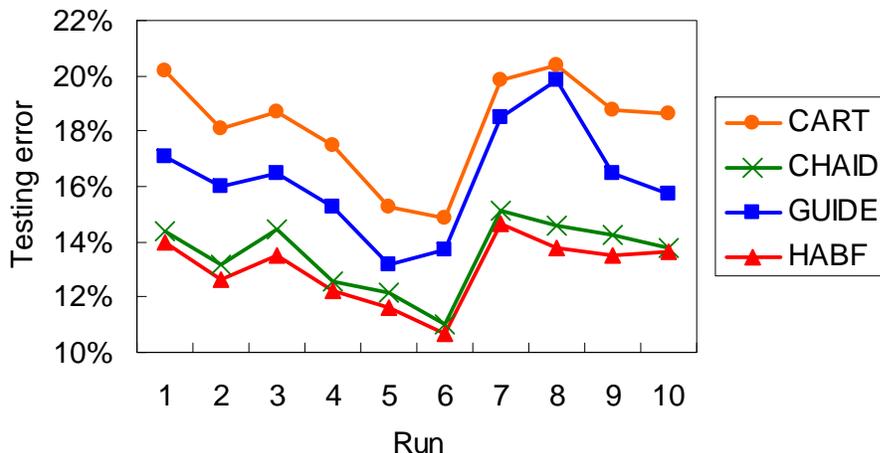
On the top of the above comparisons in accuracy and efficiency, we also noticed that CART, CHAID, and HABF selected all the six attributes for splitting while GUIDE<sub>1</sub> selected only four attributes and missed two important attributes: OR and DR. Following we draw attention to the details of the GUIDE<sub>1</sub> tree and point out its deficiency. First, we consider the GUIDE<sub>1</sub> tree is not easy to interpret because each splitting is made by grouping a large number of categorical attribute values together. For example, when FLT was used in splitting, 181 flights, including domestic and international flights, were grouped into a node. Also, when DA or OA was used, 55 airports from different regions were grouped together. Second, the prediction by GUIDE<sub>1</sub> is not thorough enough. For example, there was one terminal node that mixed 209 flights from the U.S. to Europe (EU) and 486 flights from the U.S. to Latin America (LA). Regarding these two origin-destination region pairs, the maximal checked baggage weight is the same: two pieces of 50 pounds, but the baggage fees are different. According to the historical records, the average baggage weight per passenger was 49 pounds for US-EU and 58 pounds for US-LA. Among 60% of US-EU flights and only 33% of US-LA flights, the baggage weight per passenger was less than 50 pounds (i.e., one piece). Clearly, passenger baggage behaviors in these two region pairs are quite different. However, the predicted value in GUIDE<sub>1</sub> is the same:

55 pounds for both region pairs. Thus, these 695 flights would not be considered well-predicted. Third, GUIDE, missed selecting region attributes, also affected new flight forecasting. That is, when a new flight with a new O-D pair appears, no matter what the new O-D airports are, the  $\text{GUIDE}_1$  tree would produce an identical predicted value because new flights always go into the right branch of a node and finally fall into the right-most terminal node down the tree.

To study how consistently HABF and other tree-based methods perform over time, nine more runs were carried out. Each run contains an 8-week training period, followed by a 2-week independent testing period. Run 1 starts from Jan. 1, 2009; run 2 starts from Feb. 1, 2009; ... ; run 10 starts from Oct. 1 2009. The year-round forecasting results show that HABF yielded higher forecasting accuracy than the other three tree-based methods consistently over ten runs, as shown in Figure 12. In addition, the variation of the testing errors in HABF was smaller than that in the other three methods. Despite similar testing errors given by CHAID and HABF, HABF was much more efficient than CHAID in computation. On the average, CHAID took 20 minutes while HABF finished in 2 minutes. Thus, HABF has proven to have superiority on both predictive accuracy and efficiency over the compared tree-based methods in this study. (Note: In this comparison, CART and CHAID were with  $\text{MinObs} = 2$ . To be consistent, we should have set  $\text{GUIDE}_2$  with  $\text{MinObs} = 3$ . However,  $\text{GUIDE}_2$  was considered impracticable for its long run time as mentioned earlier, so  $\text{GUIDE}_1$  with the default  $\text{MinObs}$  took the place here.)

#### 5.4.2 Sensitivity Analyses

In the above empirical study, we forecasted 2-week flights into the future by 8-week history, but we did not evaluate how far into history should one consider in a tree method. In this regard, we conduct following analyses to study how forecasting accuracy would be affected by the length of history, measured in weeks. The ultimate



**Figure 12:** Forecasting errors generated by different methods

goal is to determine the optimal number of historical weeks for the baggage weight forecasting problem.

In the first analysis, we varied the number of historical weeks from 2 to 16 and measured the corresponding forecasting errors in HABF and CHAID<sub>2</sub> as they presented high and similar forecasting accuracy in Section 5.4.1 (Figure 12). The results show that when the number of historical weeks increased from 2 to 8, the forecasting errors declined. However, the forecasting errors increased when the forecasts were based on more than 10-week history. Therefore, the optimal number of historical weeks was considered to be 8 – 10 weeks. Both HABF and CHAID reached the same conclusion as shown in Figure 13. In addition, we observed that HABF consistently outperformed CHAID except for using limited history, say 2 weeks. An interesting finding, in this regard, is that with 2-week history, only 85% of testing samples were forecasted at the FLT level (the lowest level of the hierarchy) by HABF, while this percentage substantially increased to 95% when 3-week or more history was used, as shown in Figure 14. In other words, using the limited and insufficient 2-week history in generating nodes could cause 15% of testing samples to lose at least one significant predictor in HABF. Consequently, forecasting performance got deteriorated in the

sense that the better utilization of significant predictors were, the better forecasting performance would be expected.

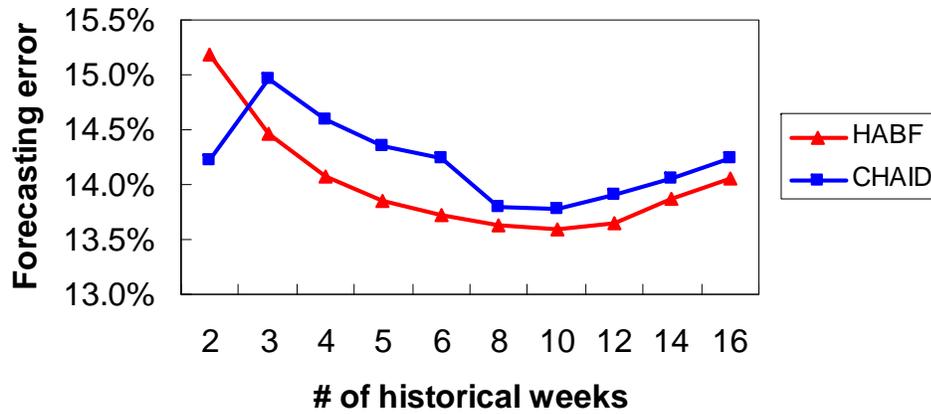


Figure 13: Forecasting errors vs. length of history

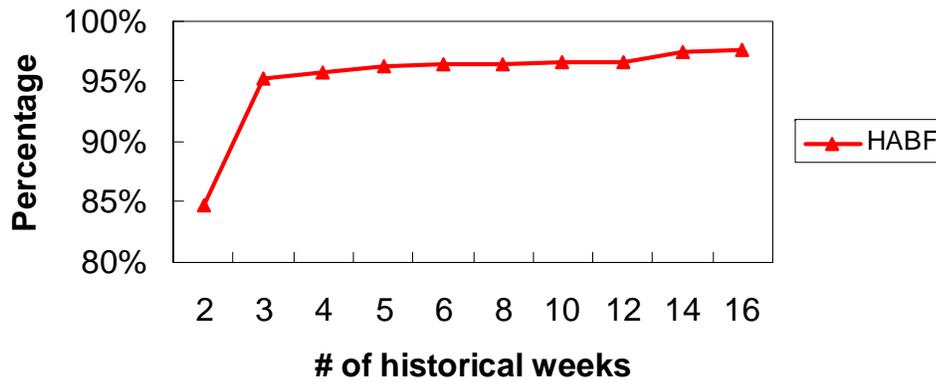


Figure 14: Percentage of testing samples forecasted by all significant predictors vs. length of history

The second analysis arises from the first analysis, where we know that the forecasting accuracy of HABF can be improved with longer history. However, the previous change in the length of history includes two concurrent facets: time and sample size. To better study the essential cause of forecasting results, we decompose the first analysis into two effects in the second analysis. For testing the pure sample size effect, we fix the horizon of history and only vary the training sample size. In Figure 15,

each curve represents a fixed historical period (i.e., 6, 8, 10, 12 weeks), and the x-axis contains varied sizes of training samples. From Figure 15, we see that all the four curves have a declined pattern. That is, given the same horizon of history (i.e., one curve), the more samples are considered in HABF, the smaller forecasting errors are obtained. Here we come to the conclusion that HABF is sensitive to the training sample size. Additionally, we can use these curves to study the time effect by drawing a vertical line to fix the training sample size and to compare the forecasting errors between different horizons of history (i.e., between curves). These four curves show that longer historical periods did not improve forecasting accuracy, but rather they make larger forecasting errors. Now we can clearly comment that the forecasting improvements in HABF did not truly come from longer history but from a larger training sample size. We extended similar analyses to CART, CHAID, and GUIDE, but there were no clear patterns regarding the time and the sample size effects in these methods.

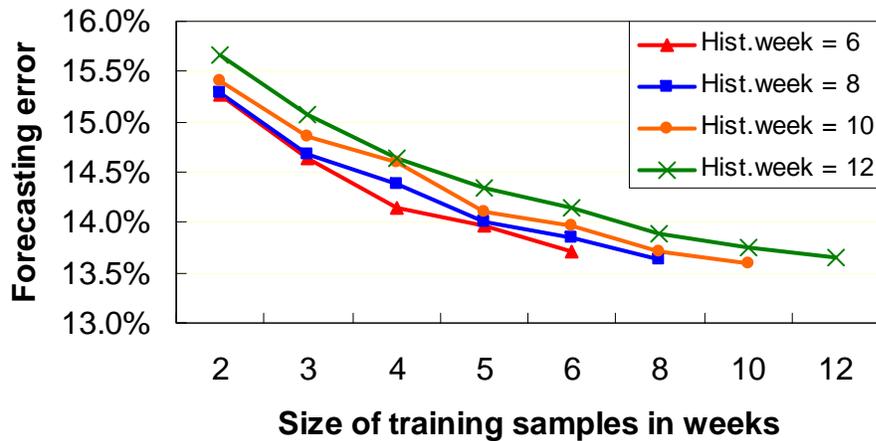


Figure 15: Forecasting errors vs. sample size and time effects

### 5.5 Simulation Study

In this section, we conduct a simulation study to investigate the performance of the proposed approach and the existing tree-based methods under different circumstances.

We first describe the simulation settings and then present the simulation results.

### 5.5.1 Settings

We used the previous real application in the air cargo industry as the baseline, on which we generated simulation data for this study. The baseline data include six independent variables ( $X_1$ : origin region,  $X_2$ : destination region,  $X_3$ : origin airport,  $X_4$ : destination airport,  $X_5$ : day of week,  $X_6$ : flight number) and a dependent variable ( $Y$ : the post-departure baggage weight per passenger carried on a flight). We considered 6050 flights between Oct. 25, 2009 and Oct. 31, 2009 for fitting a regression model, where we coded the six categorical independent variables (that have 4, 4, 130, 126, 7, and 975 categories, respectively) to dummy variable vectors, denoted by  $\mathbf{X}_1^d, \dots, \mathbf{X}_6^d$ , with binary coding. We then extracted the estimated least-squared regression coefficients with additional manipulations to simulate the response variable  $Y_{sim}$ . The general simulation model is as follows:

$$Y_{sim} = \hat{\alpha} + \sum_j \mathbf{X}_j^d \hat{\boldsymbol{\beta}}_j + \epsilon,$$

where  $\epsilon \sim \mathcal{N}(0, \sigma^2)$ ,  $\hat{\alpha}$  is the estimated intercept,  $\hat{\boldsymbol{\beta}}_j$  is a column vector of the estimated coefficients corresponding to  $\mathbf{X}_j^d$ ,  $\mathbf{X}^d = [1, \mathbf{X}_1^d, \dots, \mathbf{X}_6^d]$ , and  $[\hat{\alpha}, \hat{\boldsymbol{\beta}}_1^T, \dots, \hat{\boldsymbol{\beta}}_6^T]^T = ((\mathbf{X}^d)^T \mathbf{X}^d)^{-1} (\mathbf{X}^d)^T Y$ .

We performed six simulation experiments, which differ in terms of the scaled number of predictor categories, the inclusion of irrelevant predictors, and the interactions between predictors. The first experiment only includes predictors with small numbers of categories,  $M$  (i.e.,  $X_1$ ,  $X_2$ , and  $X_5$ ). The purpose is to examine how well the four tree-based methods (CART, CHAID, GUIDE, and HABF) would perform in this simple situation. Next, we added two predictors with larger  $M$  (i.e.,  $X_3$  and  $X_4$ ) in the second experiment to compare how much the prediction performance would change from the first experiment by the four tree-based methods. The third and the fourth experiments use the same regression models as the first two experiments, but

**Table 22:** Simulation settings

No.	Regression model	$\hat{\beta}_j$	Predictor	Remark
1	$Y \sim (X_1, \dots, X_6)$	$\hat{\beta}_3 = \hat{\beta}_4 = \hat{\beta}_6 = 0$	$X_1, X_2, X_5$	small $M$
2	$Y \sim (X_1, \dots, X_6)$	$\hat{\beta}_6 = 0$	$X_1, \dots, X_5$	large $M$
3	$Y \sim (X_1, \dots, X_6)$	$\hat{\beta}_3 = \hat{\beta}_4 = \hat{\beta}_6 = 0$	$X_1, \dots, X_5$	small $M$ , irrelevant $X_3, X_4$
4	$Y \sim (X_1, \dots, X_6)$	$\hat{\beta}_6 = 0$	$X_1, \dots, X_6$	large $M$ , irrelevant $X_6$
5	$Y \sim (X_1, \dots, X_6, X_1X_5, X_2X_5)$	$\hat{\beta}_3 = \hat{\beta}_4 = \hat{\beta}_6 = 0$	$X_1, X_2, X_5$	small $M$ , implicit interactions
6	$Y \sim (X_1, \dots, X_6, X_3X_5, X_4X_5)$	$\hat{\beta}_6 = 0$	$X_1, \dots, X_5$	large $M$ , implicit interactions

they include some irrelevant predictors when growing a tree. The purpose of these two experiments is to detect any possible variable selection bias in these tree-based methods. The next two experiments are designed to examine how the tree-based methods would perform when some predictors have interactions with other predictors. In the fifth experiment, two interactions (i.e.,  $X_1X_5$  and  $X_2X_5$ ) are included, and each of them has 28 categories. This is closer to the first experiment as all predictors are kept with small  $M$ . The sixth experiment involves another two interactions (i.e.,  $X_3X_5$  and  $X_4X_5$ ) with about 850 categories for each. This can be regarded as an extension from the second experiment where some predictors are with large  $M$ . Table 22 summaries these six simulation experiments with manipulated regression coefficients in the third column. Each simulation experiment generates 6000 training samples and 6000 testing samples (excluding new forecasting samples), with one of the two variance settings,  $\sigma^2 = 1$  or 25.

### 5.5.2 Results

In this simulation study, we set all parameters by default except the minimal number of observations, which was set to 3 (the smallest number allowed) for GUIDE and 2 for other methods. Tables 23 and 24 present the results of six simulation experiments

from four different tree-based methods. The training/testing errors and computation time are the averages based on 50 runs. We also listed the independent variables with the order selected by each tree method while splitting. The first experiment results show that all the four methods were similar in prediction and computation when all predictors were with small  $M$ . In the second experiment where predictors with large  $M$  were involved, CART performed much worse than the other three methods in both prediction accuracy and computation time. GUIDE seemed to have a tendency either to select the nested predictors with large  $M$  (i.e.,  $X_3$  and  $X_4$ ) or miss predictors with small  $M$  (i.e.,  $X_1$  and  $X_5$ ), and thus its prediction errors were not as good as those in CHAID and HABF. In the third and the fourth experiments where irrelevant predictors were included, CART, CHAID, and GUIDE more or less selected irrelevant predictors (i.e.,  $X_3$  and  $X_4$  for the third experiment and  $X_6$  for the fourth experiment). Although the prediction accuracy of these methods did not deteriorate much comparing to the first two experiments, the inclusion of irrelevant predictors did increase computation time and interpretation difficulties in these methods, especially CART. For the fifth and the sixth experiments where predictor interactions were included, the simulation results show that all these four tree-based methods can handle interactions between predictors effectively while GUIDE slightly underperformed in prediction accuracy.

In summary, we found that CART failed to handle the situation where predictors are with a large number of categories as in experiments 2 and 4. CART not only runs slowly but also predicts poorly. GUIDE has ability to produce parsimonious trees in shorter time, which makes it computationally efficient deal with moderately large datasets. However, GUIDE is not perfect for variable selection in the presence of nested predictor structures. It suffered from higher chances of missing important variables as the results of experiments 2 and 4. In general, CHAID and HABF are comparable in handling moderately large data with some predictor structures, while

Table 23: Simulation results ( $\sigma^2 = 1$ )

Experiment	Method	Training error	Testing error	Splitting variables	Run time (mins)
1	CART	4.44%	4.57%		
	CHAID	4.32%	4.47%	X <sub>2</sub> , X <sub>1</sub> , X <sub>5</sub>	< 1
	GUIDE	4.36%	4.48%		
	HABF	4.33%	4.47%		
-----					
2	CART	28.57%	27.50%	X <sub>2</sub> , X <sub>4</sub> , X <sub>1</sub> , X <sub>3</sub> , X <sub>5</sub>	10
	CHAID	4.31%	4.96%	X <sub>2</sub> , X <sub>1</sub> , X <sub>3</sub> , X <sub>4</sub> , X <sub>5</sub>	< 1
	GUIDE	9.12%	9.47%	X <sub>4</sub> , X <sub>3</sub> , X <sub>2</sub> , X <sub>5</sub> , X <sub>1</sub>	< 2
	HABF	3.26%	4.91%	X <sub>2</sub> , X <sub>1</sub> , X <sub>4</sub> , X <sub>3</sub> , X <sub>5</sub>	< 1
-----					
3	CART	4.11%	4.62%	X <sub>2</sub> , X <sub>1</sub> , X <sub>5</sub> , X <sub>3</sub> , X <sub>4</sub>	8
	CHAID	4.31%	4.52%	X <sub>2</sub> , X <sub>1</sub> , X <sub>5</sub> , X <sub>4</sub> , X <sub>3</sub>	< 1
	GUIDE	4.79%	4.98%	X <sub>4</sub> , X <sub>5</sub> , X <sub>2</sub> , X <sub>1</sub> , X <sub>3</sub>	< 1
	HABF	4.33%	4.47%	X <sub>2</sub> , X <sub>1</sub> , X <sub>5</sub>	< 1
-----					
4	CART	28.53%	27.45%	X <sub>2</sub> , X <sub>4</sub> , X <sub>6</sub> , X <sub>1</sub> , X <sub>3</sub> , X <sub>5</sub>	45
	CHAID	4.27%	5.02%	X <sub>2</sub> , X <sub>1</sub> , X <sub>3</sub> , X <sub>4</sub> , X <sub>5</sub> , X <sub>6</sub>	5
	GUIDE	5.83%	6.50%	X <sub>4</sub> , X <sub>3</sub> , X <sub>6</sub> , X <sub>5</sub> , X <sub>1</sub>	3
	HABF	3.26%	4.91%	X <sub>2</sub> , X <sub>1</sub> , X <sub>4</sub> , X <sub>3</sub> , X <sub>5</sub>	2
-----					
5	CART	3.34%	3.45%		
	CHAID	3.34%	3.45%	X <sub>2</sub> , X <sub>1</sub> , X <sub>5</sub>	< 1
	GUIDE	3.87%	4.02%		
	HABF	3.34%	3.44%		
-----					
6	CART	5.51%	5.43%	X <sub>2</sub> , X <sub>1</sub> , X <sub>5</sub> , X <sub>4</sub> , X <sub>3</sub>	4
	CHAID	3.88%	4.03%	X <sub>2</sub> , X <sub>1</sub> , X <sub>5</sub> , X <sub>3</sub> , X <sub>4</sub>	< 1
	GUIDE	5.92%	5.75%	X <sub>4</sub> , X <sub>2</sub> , X <sub>3</sub> , X <sub>1</sub> , X <sub>5</sub>	< 1
	HABF	3.73%	3.98%	X <sub>2</sub> , X <sub>1</sub> , X <sub>5</sub> , X <sub>4</sub> , X <sub>3</sub>	< 1

Table 24: Simulation results ( $\sigma^2 = 25$ )

Experiment	Method	Training error	Testing error	Splitting variables	Run time (mins)
1	CART	12.54%	13.40%		
	CHAID	12.47%	13.30%		
	GUIDE	12.61%	13.49%	$X_2, X_1, X_5$	< 1
	HABF	12.47%	13.30%		
2	CART	21.14%	20.82%	$X_2, X_4, X_1, X_3, X_5$	15
	CHAID	13.76%	14.37%	$X_2, X_3, X_4, X_5, X_1$	< 1
	GUIDE	16.32%	16.94%	$X_3, X_4, X_2$	< 1
	HABF	13.27%	14.35%	$X_2, X_1, X_4, X_3, X_5$	< 1
3	CART	11.91%	13.69%	$X_2, X_1, X_5, X_3, X_4$	9
	CHAID	12.47%	13.30%	$X_2, X_1, X_5$	< 1
	GUIDE	12.82%	13.69%	$X_4, X_2, X_1$	< 1
	HABF	12.47%	13.30%	$X_2, X_1, X_5$	< 1
4	CART	21.16%	20.90%	$X_2, X_4, X_1, X_6, X_3, X_5$	24
	CHAID	13.31%	14.55%	$X_2, X_3, X_4, X_5, X_6$	6
	GUIDE	13.93%	14.89%	$X_3, X_4, X_6, X_5$	4
	HABF	13.27%	14.35%	$X_2, X_1, X_4, X_3, X_5$	2
5	CART	12.00%	12.63%		
	CHAID	12.11%	12.72%		
	GUIDE	12.19%	12.83%	$X_2, X_1, X_5$	< 1
	HABF	12.02%	12.64%		
6	CART	11.83%	12.71%	$X_2, X_1, X_4, X_5, X_3$	7
	CHAID	13.20%	12.64%	$X_2, X_1, X_5, X_4, X_3$	< 1
	GUIDE	13.13%	12.64%	$X_4, X_2, X_3, X_1, X_5$	< 1
	HABF	12.49%	12.60%	$X_2, X_1, X_5, X_4, X_3$	< 1

CHAID may slightly select irrelevant predictors as shown in experiments 3 and 4.

## ***5.6 Conclusion and Discussion***

In this study, we propose a new forecasting approach, hierarchical attribute-based forecasting (HABF), for large-scale datasets associated with a large number of predictor categories and with observed predictor structures. HABF is similar to the conventional tree-based methods that grow a number of nodes through splitting and adopt piecewise constant prediction at terminal nodes. However, the conventional tree-based methods do not accommodate intrinsic predictor structures, and they are not generally considered efficient to deal with a large number of categorical values in predictors. Beyond the conventional tree-based methods, HABF incorporates observed predictor structures by a general linear model and adopts multi-way hierarchical splits without merging categories to make the grown trees more considerate, efficient, and interpretable.

Through an empirical study of a capacity forecasting problem in the air cargo industry, we successfully showed that HABF has higher forecasting accuracy and higher computational efficiency than three well-known tree-based methods consistently over time. Furthermore, we investigated the performance of HABF and existing tree-based methods under different circumstances via a simulation study of six experiments. The simulation results showed that the forecasting accuracy and the computational efficiency of HABF is less influenced by the number of predictor categories and the irrelevant predictors than existing tree-based methods. Although the new approach was motivated by a capacity forecasting problem in the air cargo industry, similar data characteristics can also be observed in other industries, such as transportation, hospitality, and retail. Therefore, the proposed approach can be applied to other forecasting problems as well.

HABF can be extended in following ways: First, if the data quality is not so good

that outliers are suspected, piecewise constant prediction can be enhanced with a trimmed mean function. Second, if a trend pattern is generally observed in historical observations, piecewise constant prediction can be enhanced with a weighted mean function that gives more weights on recent observations. Third, if seasonal patterns are suspected, a set of hierarchical attributes for time components, such as quarter, month, and week, can be considered in building a hierarchy. Fourth, for continuous independent variables, more discretization methods (Liu et al., 2002) other than an equal-frequency method (e.g., median, quantile) may worth further investigation. Fifth, if many cross-over predictors (i.e., predictors without structures) are involved, the significance of a partial F-test from an increase in the sum of squared errors between a full model and a reduced model (that drops one predictor) can be used to order the cross-over predictors before applying all predictors to a general linear model. This can avoid HABF generating different ordering results that simply arise from different entering order of variables into the general linear model.

## CHAPTER VI

### FUTURE WORK

This chapter outlines a number of extended research topics from the present studies.

- In Chapter 3, the size of variable sets ( $c$ ) used in IRPLRL<sub>1</sub> is arbitrarily chosen and the temperature ( $T$ ) in simulated annealing is assumed to be constant. The effects of  $c$  and  $T$  on the variable selection consistency and the convergence rate may need further evaluation.
- The new classification approach proposed in Chapter 3 is not limited to PLRL<sub>1</sub>. Any classification method that is applicable to the  $p > n$  situation and/or includes a variable selection scheme can play a role as penalized logistic regression. The iterative reselection algorithm can also be applied to other classification methods as well.
- New classification methods developed in Chapters 3 and 4 can be extended from two-class to multi-class problems in the future.
- Combining two methods from Chapters 3 and 4 may also be a potential research topic. Accommodating correlated samples in the discriminant analysis will be another challenging research topic.
- In Chapter 5, other feasible algorithms than a general linear model for variable ranking can be incorporated into the proposed forecasting approach.
- The influence of different discretization methods on prediction performance may worth further investigation.

- The proposed forecasting approach in Chapter 5 can be further applied to and assessed by other forecasting problems with similar data characteristics as future work.

## APPENDIX A

### PROOF OF THEOREM 1

#### Assumptions

A1: There exists a positive constant vector  $\eta$  such that

$$|C_{21}^N (C_{11}^N)^{-1} \text{sign}(\alpha_{(1)})| \leq \mathbf{1} - \eta,$$

where  $\mathbf{1}$  is a  $(p-q) \times 1$  vector of 1's, and the inequality holds element-wise.

A2: The inverse of  $(\mathbf{X}(1)^T \mathbf{X}(1))$  exists, and there exists  $0 \leq c_2 \leq 1$  and  $M_1, M_2, M_3, M_4 > 0$ , so the following holds:

$$\frac{1}{N} (\mathbf{x}_{ij})^T \mathbf{x}_{ij} \leq M_1, \forall i \text{ and } j,$$

$$\min \left\{ a^T \left( (C_{11}^N)^T \left( \frac{\mathbf{X}(1)^T \mathbf{X}(1)}{N} \right)^{-1} C_{11}^N \right) a, a^T C_{11}^N a \right\} \geq M_2, \forall \|a\|_2^2 = 1,$$

and

$$N^{\frac{1-c_2}{2}} \min_{k=1, \dots, q} |\alpha_k| \geq M_3.$$

A3:  $E|x_{ij,s}x_{ij,t}x_{ij,k}| < \infty$  for all  $1 < s, t, k < p$ .

LEMMA 1: Under assumption A1 with  $\eta > 0$ , then

$$P(\hat{\boldsymbol{\alpha}}(\lambda) =_S \boldsymbol{\alpha}) \geq P(\Upsilon_1 \cap \Upsilon_2),$$

where

$$\Upsilon_1 = \left\{ \left| (C_{11}^N)^{-1} \frac{\mathbf{X}(1)^T [\mathbf{y} - \pi(\boldsymbol{\alpha}, \sigma_b)]}{\sqrt{N}} \right| < \sqrt{N} \left( |\alpha_{(1)}| - \frac{\lambda}{N} |(C_{11}^N)^{-1} \text{sign}(\alpha_{(1)})| \right) \right\}$$

and

$$\Upsilon_2 = \left\{ \left| \left( C_{21}^N (C_{11}^N)^{-1} \mathbf{X}(1)^T - \mathbf{X}(2)^T \right) \frac{[\mathbf{y} - \pi(\boldsymbol{\alpha}, \sigma_b)]}{\sqrt{N}} \right| \leq \frac{\lambda}{\sqrt{N}} \eta \right\}.$$

PROOF OF LEMMA 1: If the random effect follows a normal distribution, the PLMM likelihood function (10) can be written as

$$|\Sigma|^{-1/2} \int \exp \left( \sum_{i=1}^n \sum_{j=1}^m (y_{ij} \log \frac{\pi_{ij}^{\boldsymbol{\beta}}}{1 - \pi_{ij}^{\boldsymbol{\beta}}} + \log(1 - \pi_{ij}^{\boldsymbol{\beta}})) - \frac{1}{2} \boldsymbol{\beta}' \Sigma^{-1} \boldsymbol{\beta} \right) d\boldsymbol{\beta} - \lambda \sum_{i=1}^p |\alpha_i|. \quad (17)$$

Because of the difficulty in implementing the integration, Laplace's method is applied (Barndorff-Nielsen and Cox, 1989, Sec. 3.3, Tierney and Kadane, 1986). Follow the same derivation in Breslow and Clayton (1993), the integrated log-likelihood in the first part of (17) can be approximated by

$$-\frac{1}{2} |\mathbf{I} + Z^T W Z \Sigma| + \sum_{i=1}^n \sum_{j=1}^m \left( (y_{ij} \log \frac{\pi_{ij}^{\boldsymbol{\beta}}}{1 - \pi_{ij}^{\boldsymbol{\beta}}} + \log(1 - \pi_{ij}^{\boldsymbol{\beta}})) \right) - \frac{1}{2} \boldsymbol{\beta}' \Sigma^{-1} \boldsymbol{\beta},$$

and the penalized log likelihood estimator can be calculated by

$$\begin{aligned} \hat{\boldsymbol{\alpha}} &= \arg \min_{\boldsymbol{\alpha}} \sum_{i=1}^n \sum_{j=1}^m \left( -y_{ij} \log \frac{\pi_{ij}(\boldsymbol{\alpha}, \sigma_b)}{1 - \pi_{ij}(\boldsymbol{\alpha}, \sigma_b)} - \log(1 - \pi_{ij}(\boldsymbol{\alpha}, \sigma_b)) \right) + \frac{1}{2} \boldsymbol{\beta}' \Sigma^{-1} \boldsymbol{\beta} + \lambda \|\boldsymbol{\alpha}\|_1 \\ &= \arg \min_{\boldsymbol{\alpha}} NL(\boldsymbol{\alpha}) + \lambda \|\boldsymbol{\alpha}\|_1. \end{aligned}$$

Let  $u = \hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}$ , we have

$$\hat{u} = \arg \min_u \Gamma(u),$$

where

$$\Gamma(u) = NL(\boldsymbol{\alpha} + u) + \lambda \|\boldsymbol{\alpha} + u\|_1.$$

Using Taylor expansion, the first term on the right hand side can be written as

$$NL(\boldsymbol{\alpha} + u) = NL(\boldsymbol{\alpha}) + A_1 + A_2 + A_3,$$

where

$$A_1 = - \sum_{i=1}^n \sum_{j=1}^m [y_{ij} - \pi_{ij}(\boldsymbol{\alpha}, \sigma_b)] \mathbf{x}_{ij}^T u = -u^T \mathbf{X}^T (\mathbf{y} - \boldsymbol{\pi}),$$

$$A_2 = \sum_{i=1}^n \sum_{j=1}^m \frac{1}{2} [\pi_{ij}(\boldsymbol{\alpha}, \sigma_b) (1 - \pi_{ij}(\boldsymbol{\alpha}, \sigma_b))] u^T \mathbf{x}_{ij} \mathbf{x}_{ij}^T u = \frac{1}{2} u^T \mathbf{X}^T \mathbf{W} \mathbf{X} u,$$

and

$$A_3 = \frac{1}{6} \sum_{i=1}^n \sum_{j=1}^m \pi_{ij}(\boldsymbol{\alpha}, \sigma_b) (1 - \pi_{ij}(\boldsymbol{\alpha}, \sigma_b)) (2\pi_{ij}(\boldsymbol{\alpha}, \sigma_b) - 1) (\mathbf{x}_{ij}^T u)^3.$$

Based on assumption A3,  $A_3 \rightarrow 0$ . Therefore, we have

$$NL(\boldsymbol{\alpha} + u) = NL(\boldsymbol{\alpha}) - \mathbf{X}^T [\mathbf{y} - \boldsymbol{\pi}(\boldsymbol{\alpha}, \sigma_b)] u + \frac{(\sqrt{N}u)^T C^N (\sqrt{N}u)}{2}. \quad (18)$$

The differentiation of (18) with respect to  $u$  leads to

$$\sqrt{N} \left( C^N (\sqrt{N}u) - \frac{\mathbf{X}^T [\mathbf{y} - \boldsymbol{\pi}(\boldsymbol{\alpha}, \sigma_b)]}{\sqrt{N}} \right).$$

Based on the KKT optimality condition, we achieve a result which is similar to Proposition 1 in Zhao and Yu (2006). That is, if there exists  $\hat{u}$ , the following holds:

$$C_{11}^N (\sqrt{N} \hat{u}(1)) - \frac{\mathbf{X}(1)^T [\mathbf{y} - \boldsymbol{\pi}(\boldsymbol{\alpha}, \sigma_b)]}{\sqrt{N}} = -\frac{\lambda}{\sqrt{N}} \text{sign}(\alpha_{(1)}), \quad (19)$$

$$|\hat{u}(1)| < |\alpha_{(1)}|, \quad (20)$$

and

$$-\frac{\lambda}{\sqrt{N}} \mathbf{1} \leq C_{21}^N (\sqrt{N} \hat{u}(1)) - \frac{\mathbf{X}(2)^T [\mathbf{y} - \boldsymbol{\pi}(\boldsymbol{\alpha}, \sigma_b)]}{\sqrt{N}} \leq \frac{\lambda}{\sqrt{N}} \mathbf{1}. \quad (21)$$

Then  $\text{sign}(\hat{\alpha}_{(1)}) = \text{sign}(\alpha_{(1)})$  and  $\hat{\alpha}_2 = u(2) = 0$ .

From (19), (20), and (21), we have

$$\left| (C_{11}^N)^{-1} \frac{\mathbf{X}(1)^T [\mathbf{y} - \boldsymbol{\pi}(\boldsymbol{\alpha}, \sigma_b)]}{\sqrt{N}} \right| < \sqrt{N} \left( |\alpha_{(1)}| - \frac{\lambda}{N} |(C_{11}^N)^{-1} \text{sign}(\alpha_{(1)})| \right)$$

and

$$\left| \left( C_{21}^N (C_{11}^N)^{-1} \mathbf{X}(1)^T - \mathbf{X}(2)^T \right) \frac{[\mathbf{y} - \pi(\boldsymbol{\alpha}, \sigma_b)]}{\sqrt{N}} \right| \leq \frac{\lambda}{\sqrt{N}} \left( \mathbf{1} - |C_{21}^N (C_{11}^N)^{-1} \text{sign}(\alpha_{(1)})| \right).$$

PROOF OF THEOREM 1 (SIGN CONSISTENCY):

Based on LEMMA 1 and

$$\begin{aligned} 1 - P(\Upsilon_1 \cap \Upsilon_2) &\leq P(\Upsilon_1^C) + P(\Upsilon_2^C) \\ &\leq \sum_{i=1}^q P(|(h_i^A)^T [\mathbf{y} - \pi]| > \sqrt{N} (|\alpha_{(1)i}| - \frac{\lambda}{N} |b_i|)) \\ &\quad + \sum_{i=1}^{p-q} P(|(h_i^B)^T [\mathbf{y} - \pi]| \geq \frac{\lambda}{\sqrt{N}} \eta_i), \end{aligned}$$

where  $b = (b_1, \dots, b_p)^T = (C_{11}^N)^{-1} \text{sign}(\alpha_{(1)})$ ,  $H_A^T = (h_1^A, \dots, h_q^A)^T = (C_{11}^N)^{-1} \frac{\mathbf{X}(1)^T}{\sqrt{N}}$ , and  $H_B^T = (h_1^B, \dots, h_{p-q}^B)^T = \left( C_{21}^N (C_{11}^N)^{-1} \frac{\mathbf{X}(1)^T}{\sqrt{N}} - \frac{\mathbf{X}(2)^T}{\sqrt{N}} \right)$ .

Because

$$H_A^T H_A = (C_{11}^N)^{-1} \frac{\mathbf{X}(1)^T \mathbf{X}(1)}{\sqrt{N} \sqrt{N}} ((C_{11}^N)^{-1})^T$$

and

$$H_B^T H_B = \mathbf{X}(2)^T \left[ I + \mathbf{X}(1) ((C_{11}^N)^{-1} \mathbf{X}(1)^T \mathbf{X}(1) C_{11}^N - 2(C_{11}^N)^{-1}) \mathbf{X}(1)^T \right] \mathbf{X}(2).$$

Based on assumption A2, we have

$$\|h_k^A\|_2^2 \leq \frac{1}{M_2}, \forall k = 1, \dots, q \text{ and } \|h_k^B\|_2^2 \leq M_1, \forall k = 1, \dots, p - q. \quad (22)$$

Also, assuming that the number of active factors will not increase with  $n$ , we have

$$\left| \frac{\lambda}{N} b \right| = \left| \frac{\lambda}{N} (C_{11}^N)^{-1} \text{sign}(\alpha_{(1)}) \right| \leq \frac{\lambda}{N M_2} \sqrt{q}. \quad (23)$$

Assuming that  $y_{ij} - \pi_{ij}(\boldsymbol{\alpha}, \sigma_b)$ 's are normally distributed with mean 0 and variance  $\pi_{ij}(\boldsymbol{\alpha}, \sigma_b)(1 - \pi_{ij}(\boldsymbol{\alpha}, \sigma_b))$ . Based on (22), (23), and the fact that for  $t > 0$ , the normal distribution tail is bounded by  $1 - \Phi(t) < t^{-1} e^{-\frac{1}{2}t^2}$ , we have, for  $\lambda \propto N^{\frac{1+c_4}{2}}$  and

$c_4 < c_2$ ,

$$\begin{aligned}
& \sum_{k=1}^q P(|(h_k^A)^T[\mathbf{y} - \pi]| > \sqrt{N}(|\alpha_k| - \frac{\lambda}{N}|b_k|)) \\
&= qO(1 - \Phi((1 + o(1))\frac{M_3 M_2 N^{c_2/2}}{\sqrt{\max_{ij} E(y_{ij} - \pi_{ij})^2}})) \\
&= o(e^{-N^{c_3}})
\end{aligned} \tag{24}$$

and

$$\begin{aligned}
& \sum_{k=1}^{p-q} P(|(h_k^B)^T[\mathbf{y} - \pi]| \geq \frac{\lambda}{\sqrt{N}}\eta_k) \\
&= (p - q)O(1 - \Phi(\frac{1}{M_1} \frac{\lambda}{\sqrt{N}} \frac{\eta}{\sqrt{\max_{ij} E(y_{ij} - \pi_{ij})^2}})) \\
&= o(e^{-N^{c_3}}).
\end{aligned} \tag{25}$$

## REFERENCES

- [1] Aarts, E. and Lenstra, J. (1997) *Local Search in Combinatorial Optimization*, Wiley, Chichester.
- [2] Airbus (2009), *Global Market Forecast: 2009-2028*, Airbus, Toulouse, France.
- [3] Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S. Mack, D., and Levine, A. J. (1999) “Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays,” *Proceedings of the National Academy of Sciences USA*, **96**, 6745-6750.
- [4] Amaratunga, D. and Cabrera, J. (2004) *Exploration and Analysis of DNA Microarray and Protein Array Data*, Wiley, New Jersey.
- [5] Antoniadis, A. and Fan, J. (2001) “Regularization of wavelet approximations,” *Journal of American Statistical Association*, **96**, 939-967.
- [6] Bair, E. Hastie, T., Paul, D., and Tibshirani, R. (2006) “Prediction by supervised principal components,” *Journal of American Statistical Association*, **101**, 119-137.
- [7] Barndorff-Nielsen, O. E. and Cox, D. R. (1989) *Asymptotic Techniques for Use in Statistics*, Chapman & Hall, London.
- [8] Benjamini, Y. and Hochberg, Y. (1995) “Controlling the false discovery rate: a practical and powerful approach to multiple testing,” *Journal of Royal Statistical Society Series B*, **57**, 289-300.

- [9] Biggs, D., De Ville, B, and Suen, E. (1991) "A method of choosing multiway partitions for classification and decision trees," *Journal of Applied Statistics*, **18**, 49-62.
- [10] Blum, A. and Langley, P. (1997) "Selection of relevant features and examples in machine learning," *Artificial Intelligence*, **97**, 245-271.
- [11] Boeing (2003), *World Air Cargo Forecast: 2008-2009*, Boeing Commercial Aircraft Company, Seattle.
- [12] Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J. (1984) *Classification and Regression Trees*, Wadsworth, Belmont, California.
- [13] Breslow, N. E. and Clayton, D. G. (1993) "Approximate inference in generalized linear mixed models," *Journal American Statistical Association*, **88**, 9-25.
- [14] Chan, K. S. and Ledolter, J. (1995) "Monte Carlo EM estimation for time series models involving counts," *Journal of American Statistical Association*, **90**, 242-252.
- [15] Chen, J. S. (2003) "Developing a travel segmentation methodology: a criterion-based approach," *Journal of Hospitality and Tourism Research*, **27**, 310-327.
- [16] Donoho, D. and Johnstone I. (1994) "Ideal spatial adaption by wavelet shrinkage," *Biometrika*, **81**, 425-455.
- [17] Dudoit, S., Fridlyand, J. and Speed, T. P. (2002) "Comparison of discrimination methods for the classification of tumors using gene expression data," *Journal of American Statistical Association*, **97**, 77-87.
- [18] Duda, R. O. and Hart, P. E. (1973) *Pattern Classification and Scene Analysis*, John Wiley & Sons, New York.

- [19] Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004) “Least angle regression,” *Annals of Statistics*, **32**, 407-451.
- [20] Fan, J. and Li, R. (2001) “Variable selection via nonconcave penalized likelihood and its oracle properties,” *Journal of American Statistical Association*, **96**, 1348-1360.
- [21] Fisher R. A. (1951) *The Design of Experiments*, 6<sup>th</sup> edition, Oliver and Boyd., London.
- [22] Friedman, J., Hastie, T., Hofling, H., and Tibshirani, R. (2007) “Pathwise coordinate optimization,” *The Annals of Applied Statistics*, **1**, 302-332.
- [23] Friedman, J., Hastie, T., and Tibshirani, R. (2010) “Regularization paths for generalized linear models via coordinate descent,” *Journal of Statistical Software*, **33**(1).
- [24] Fu, W. J. (1998) “Penalized regressions: The bridge versus the lasso,” *Journal of Computational and Graphical Statistics*, **7**, 397-416.
- [25] Goeman, J. (2008) “An efficient algorithm for  $L_1$ -penalized estimation,” *Technical Report*, Dept. of Medical Statistics and Bioinformatics, Leiden University.
- [26] Ghosh, D. (2003) “Penalized discriminant methods for the classification of tumors from gene expression data,” *Biometrics*, **59**, 992-1000.
- [27] Ghosh, D. and Chinnaiyan, A. M. (2005) “Classification and selection of biomarkers in genomic data using Lasso,” *Journal of Biomedicine and Biotechnology*, **2**, 147-154.
- [28] Gidas, B. (1985) “Nonstationary Markov chains and convergence of the annealing algorithm,” *Journal of Statistical Physics*, **39**, 73-131.

- [29] Glover, F. (1986) “Future paths for integer programming and links to artificial intelligence,” *Computers & Operations Research*, **13**, 533-549.
- [30] Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D., and Lander, E. S. (1999) “Molecular classification of cancer: class discovery and class prediction by gene expression monitoring,” *Science*, **286**, 531-537.
- [31] Guo, Y., Hastie, T. and Tibshirani, R. (2007) “Regularized linear discriminant analysis and its application in microarrays,” *Biostatistics*, **8**, 86-100.
- [32] Guyon, I. and Vapnik, V. (2002) “Gene selection for cancer classification using support vector machines,” *Machine Learning*, **46**, 389-422.
- [33] Hajek, B. (1988) “Cooling schedules for optimal annealing,” *Mathematics of Operations Research*, **13**, 311-329.
- [34] Hall, M. (1999) *Correlation-based Feature Selection for Machine Learning*, Ph.D. Thesis, Dept. of Computer Science, Waikato University, New Zealand.
- [35] Han, Y and Lam, W. (2007) “Utilizing hierarchical feature domain values for prediction,” *Data and Knowledge Engineering*, **61**, 540-553.
- [36] Hastie, T., Tibshirani, R. and Friedman, J. (2008) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, New York.
- [37] Hill T. and Lewicki P. (2006) *Statistics: Methods and Applications*, StatSoft, Inc., Tulsa, Oklahoma.
- [38] Holland J. H. (1975) *Adaption in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Michigan.
- [39] Huang, J., Ma, S. and Zhang, C. H. (2008), “Adaptive Lasso for sparse high-dimensional regression models,” *Statistics Sinica*, **18**, 1603-1618.

- [40] International Air Transport Association (2010) “World air transport statistics: scheduled passenger - kilometres flown,” 54<sup>th</sup> edition, Montreal, Quebec.  
<http://www.iata.org/ps/publications/wats-passenger-km.htm>
- [41] Kass, G. V. (1980) “An exploratory technique for investigation large quantities of categorical data,” *Applied Statistics*, **29**, 119-127.
- [42] Kerr, M. K. (2003) “Design considerations for efficient and effective microarray studies,” *Biometrics*, **59**, 822-828.
- [43] Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P. (1983) “Optimization by Simulated Annealing,” *Science*, **220**, 671-680.
- [44] Knight, K. and Fu, W. (2000) “Asymptotics for Lasso-type estimators,” *Annals of Statistics*, **28**, 1356-1378.
- [45] Kohavi, R. and John, G. (1997) “Wrappers for feature selection,” *Artificial Intelligence*, **97**, 273-324.
- [46] Lee, M. L. T. (2004) *Analysis of Microarray Gene Expression Data*, Kluwer Academic Publishers, Boston, Massachusetts.
- [47] Lee, M. L. T., Kuo, F. C., Whitmore, G. A., and Sklar, J. (2000) “Importance of replication in microarray gene expression studies: Statistical methods and evidence from repetitive cDNA hybridizations,” *Proceedings of the National Academy of Sciences USA*, **97**, 9834-9839.
- [48] Lee, M. L. T., Lu, W., Whitmore, G. A. and Beier, D. (2002) “Models for microarray gene expression data,” *Journal of Biopharmaceutical Statistics*, **12**, 1-19.
- [49] Leng, C., Lin, Y. and Wahba, G. (2006) “A note on the Lasso and related procedures in model selection,” *Statistica Sinica*, **16**, 1273-1284.

- [50] Liao, J. and Chin, K. (2007) “Logistic regression for disease classification using microarray data: model selection in a large p and small n case,” *Bioinformatics*, **23**, 1945-1951.
- [51] Liu, H., Hussain, F., Tan, C. L., and Dash, M. (2002) “Discretization: an enabling technique,” *Data Mining and Knowledge Discovery*, **6**, 393-423.
- [52] Liu, J., Iba, H. and Ishozuka, M. (2001) “Selecting informative genes with parallel genetic algorithms in tissue classification,” *Genome Informatics*, **12**, 14-23.
- [53] Liu, Z. Jiang, F. Tian, G., Wang, S., Sato, F., Meltzer, S., and Tan, M. (2007) “Sparse logistic regression with  $L_p$  penalty for biomarker identification,” *Statistical Applications in Genetics and Molecular Biology*, **6**(1), 6.
- [54] Loh, W. (2002) “Regression trees with unbiased variable selection and interaction detection,” *Statistica Sinica*, **12**, 361-386.
- [55] Loh, W. (2008) “Classification and regression tree methods,” in Ruggeri F., Kenett, R., and Faltin F. (eds), *Encyclopedia of Statistics in Quality and Reliability*, Wiley, 315-323.
- [56] Lundy, M. (1985) “Applications of annealing algorithm to combinatorial problems in statistics,” *Biometrika*, **72**, 191-198.
- [57] Lundy, M. and Mees, A. (1986) “Convergence of an annealing algorithm,” *Mathematical Programming*, **34**, 111-124.
- [58] Ma, S. and Huang, J. (2008) “Penalized feature selection and classification in bioinformatics,” *Briefings in Bioinformatics*, **9**, 392-403.
- [59] Magidson, J. (1993) “The use of the new ordinal algorithm in CHAID to target profitable segments,” *The Journal of Database Marketing*, **1**, 29-48.

- [60] Magidson, J. (1994) “The CHAID approach to segmentation modeling: chi-squared automatic interaction detection,” in Bagozzi, R. (eds), *Advanced Methods of Marketing Research*, Blackwell, Oxford, 118-159.
- [61] Martin, K. J., Graner, E., Li, Y, Price, L. M., Kritzman, B. M., Fournier, M. V., Rhei, E. and Pardee, A. B. “High-sensitivity array analysis of gene expression for the early detection of disseminated breast tumor cells in peripheral blood,” *Proceedings of the National Academy of Sciences USA*, **98**, 2646-2651.
- [62] McCulloch, C. E. (1997), “Maximum likelihood algorithms for generalized linear mixed models,” *Journal of American Statistical Association*, **92**, 162-170.
- [63] McCulloch, W. and Pitts, W. (1943) “A logical calculus of the ideas immanent in nervous activity,” *Bulletin of Mathematical Biophysics*, **7**, 115-133.
- [64] Meinshausen, N. and Bhlmann, P. (2006) “High dimensional graphs and variable selection with the lasso,” *Annals of Statistics*, **34**, 1436-1462.
- [65] Mitra, D., Romeo, F. and Sangiovanni-Vicentelli, A. (1986) “Convergence and finite-time behavior of simulated annealing,” *Advances in Applied Probability*, **18**, 747-771.
- [66] Murty K. (1995) *Operations Research: Deterministic Optimization Models*, Prentice-Hall, New Jersey.
- [67] Murthy, S. K. (1998) “Automatic construction of decision trees from data: a multi-disciplinary survey,” *Data Mining and Knowledge Discovery*, **2**, 345-389.
- [68] Nguyen, D. V. and Rocke, D. M. (2002) “Tumor classification by partial least squares using microarray gene expression data,” *Bioinformatics*, **18**, 39-50.

- [69] Pan, W., Shen, X., Jiang, A. and Hebbel, R. P. (2006) "Semi-supervised learning via penalized mixture model with application to microarray sample classification," *Bioinformatics*, **22**, 2388-2395.
- [70] Parmigiani, G., Garrett, E. S., Irizarry, R. A. and Zeger, S. L. (2003) *The Analysis of Gene Expression Data: Methods and Software*, Springer, New York.
- [71] Press, S. J. and Wilson, S. (1978) "Choosing between logistic regression and discriminant analysis," *Journal of American Statistical Association*, **73**, 699-705.
- [72] Qin, X. and Han, J. (2008) "Variable selection issues in tree-based regression models," *Transportation Research Record*, **2061**, 30-38.
- [73] Ragavan, H. and Rendell, L. (1993) "Lookahead feature construction for learning hard concepts," *Proceedings of the 10<sup>th</sup> International Conference on Machine Learning*, 252-259.
- [74] Rajasekaran, S. (2000) "On simulated annealing and nested annealing," *Journal of Global Optimization*, **16**, 43-56.
- [75] Ratner, B. (2003) *Statistical Modeling and Analysis for Database Marketing: Effective Techniques for Mining Big Data*, CRC Press, Boca Raton, Florida.
- [76] Roth, V. (2002) "The generalized Lasso," *Technical Report*, Dept. of Computer Science, University of Bonn.
- [77] Rosa, G. J. M., Steibel, J. P. and Tempelman, R. J. (2005) "Reassessing design and analysis of two-Color microarray experiments using mixed effects models," *Comparative and Functional Genomics*, **6**, 123-131.
- [78] Rzhetsky Andrey (2008) *Major Research Topics in Bioinformatics*, Dept. of Biomedical Informatics, Columbia University.  
<http://www.dbmi.columbia.edu/bioinformatics/docs/ResearchOpportunities.html>

- [79] Saeys, Y., Inza, I. and Larranaga, P. (2007) “A review of feature selection techniques in bioinformatics,” *Bioinformatics*, **23**, 2507-2517.
- [80] Segal, M. R., Dahlquist, K. D. And Conklin, B. R. (2003) “Regression approaches for microarray data analysis,” *Journal of Computational Biology*, **10**, 961-980.
- [81] Shevade, S. K. and Keerthi, S. S. (2003) “A simple and efficient algorithm for gene selecting using sparse logistic regression,” *Bioinformatics*, **19**, 2246-2253.
- [82] Shi, L. (2002) *DNA Microarray - Monitoring the Genome on a Chip*.  
<http://www.gene-chips.com/>
- [83] Speed, T. P. (2003) *Statistical Analysis of Gene Expression Microarray Data*, CRC Press, Florida.
- [84] SPSS (1999) “AnswerTree algorithm summary,” White paper, SPSS Inc.
- [85] Sutton, C. D. (2005) “Classification and regression trees, bagging, and boosting,” *Handbook of Statistics*, **24**, 303-329.
- [86] Tanner, M. A. (1993) *Tools for Statistical Inference: Observed Data and Data Augmentation*, 2<sup>nd</sup> edition, Springer-Verlag, Berlin.
- [87] Tibshirani, R. (1996) “Regression shrinkage and selection via the lasso,” *Journal of Royal Statistical Society Series B*, **58**, 267-288.
- [88] Tibshirani, R. (1997) “The Lasso method for variable selection in the Cox model,” *Statistics in Medicine*, **16**, 385-395.
- [89] Tierney L and Kadane J. (1986) “Accurate approximations for posterior moments and marginal densities,” *Journal of American Statistical Association*, **81**, 82-86.

- [90] Tusher, V., Tibshirani, R. and Chu, C. (2001) “Significance analysis of microarrays applied to transcriptional responses to ionizing radiation,” *Proceedings of the National Academy of Sciences USA*, **98**, 5116-5121.
- [91] Vapnik, V. (1998) *Statistical Learning Theory*. John Wiley & Sons, New York.
- [92] Weatherford, L. R. and Kimes, S. E. (2003) “A comparison of forecasting methods for hotel revenue management,” *International Journal of Forecasting*, **19**, 401-415.
- [93] Welch, B. L. (1947) “The generalization of ‘student’ problem when several different population variances are involved,” *Biometrika*, **34**, 28-35.
- [94] West, M., Blanchette, C., Dressman, H., Huang, E., Ishida, S., Spang, R., Zuzan, H., Olson, J., Marks, J. and Nevins, J. (2001) “Predicting the clinical status of human breast cancer by using gene expression profiles,” *Proceedings of the National Academy of Sciences USA*, **98**, 11462-11467.
- [95] Weston, J., Muckerjee, S., Chapelle, O., Pontil, M., Poggio, T. and Vapnik, V. (2000) “Feature selection for SVMs,” *Proceedings of NIPS*.
- [96] Wilcoxon, F. (1945) “Individual comparisons by ranking methods,” *Biometrics*, **1**, 80-83.
- [97] Wong, G. et al. (2005) *DNA Microarray Data Analysis*, 2<sup>nd</sup> edition, CSC Scientific Computing Ltd., Helsinki, Finland.
- [98] Wu, B. (2006) “Differential gene expression detection and sample classification using penalized linear regression models,” *Bioinformatics*, **22**, 472-476.
- [99] Wu, T. T. and Lange, K. (2008) “Coordinate descent algorithms for lasso penalized regression,” *The Annals of Applied Statistics*, **2**, 224-244.

- [100] Yang, Y., Hoh, J., Broger, C., Neeb, M., Edington, J., Lindpaintner, K. and Ott, J. (2003) “Statistical methods for analyzing microarray feature data with replications,” *Journal of Computational Biology*, **10**, 157-169.
- [101] Zhang, C., Fu, H., Jiang, Y. and Yu, T. (2007) “High-dimensional pseudo-logistic regression and classification with applications to gene expression data,” *Computational Statistics and Data Analysis*, **52**, 452-470.
- [102] Zhao, P. and Yu, B. (2006), “On model selection consistency of lasso,” *Journal of Machine Learning Research*, **7**, 2541-2563.
- [103] Zhu, J. and Hastie, T. (2004) “Classification of gene microarrays by penalized logistic regression,” *Biostatistics*, **5**, 427-443.
- [104] Zou, H. (2006) “The adaptive lasso and its oracle properties,” *Journal of American Statistical Association*, **101**, 1418-1429.
- [105] Zou, H. and Hastie, T. (2005) “Regularization and variable selection via the elastic net,” *Journal of Royal Statistical Society Series B*, **67**, 301-320.