Prediction Oriented Marker Selection (PROMISE)
for High Dimensional Regression with Application
 to Personalized Medicine

by
Soyeon Kim

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree
Doctor of Philosophy

Approved, Thesis Committee:

David W. Scott, Chair
Noah Harding Professor of Statistics

J. Jack Lee, Director
Professor of Biostatistics, The University
of Texas MD Anderson Cancer Center

Veerabhadran Baladandayuthapani,
Co-Director
Associate Professor of Biostatistics, The
University of Texas MD Anderson Cancer
Center

Katherine B. Enos
Professor of Statistics

Luay K. Nakhleh
Associate Professor of Computer Science

Houston, Texas
April, 2015
ABSTRACT

Prediction Oriented Marker Selection (PROMISE) for High Dimensional Regression with Application to Personalized Medicine

by

Soyeon Kim

In personalized medicine, biomarkers are used to select therapies with the highest likelihood of success based on a patient’s individual biomarker profile. In statistics, two goals for personalized medicine are (1) selecting important biomarkers to accurately predict treatment outcome and (2) removing trivial markers to reduce cost of biological verification of the markers. These goals are challenging because the number of candidate biomarkers is much larger than the number of patients. For high dimensional variable selection, penalized regression methods such as the lasso and the elastic net generate promising results. However, selecting the right amount of penalization is critical because it directly affects variable selection, and therefore prediction as well. To select the regularization parameter, cross-validation (CV) is most commonly used. It tends to provide high prediction accuracy as well as a high true positive rate, but at the cost of a high false positive rate. Resampling methods such as stability selection (SS) conversely maintains a good control of the false positive rate, but at the cost of yielding too few true positives. We propose prediction oriented marker selection (PROMISE), which combines SS with CV, to have good prediction accuracy and low false positive rate. We applied PROMISE to (1) the lasso and (2) the elastic net for individual marker selection, (3) the group lasso for pathway selection, and
(4) the combination of the group lasso with the lasso for individual marker selection within the selected pathways. Data analysis show that PROMISE produces a more sparse solution than CV, significantly reducing the false positives compared to CV, while giving similar prediction accuracy and true positives with CV. In our studies, SS selects too few variables to accurately predict treatment outcomes. PROMISE can be applied in many fields to select regularization parameters when the goals are to minimize both type I and type II errors and to maximize prediction accuracy.
Acknowledgment

Foremost, I wish to thank Dr. Jack Lee for suggesting an interesting research topic to me. He provided motivation and encouragement when my progress was not going well to become an independent research. I appreciate his patience during this process, especially for his financial support over the past five years, including during my pregnancy. I also wish to thank my co-advisor Dr. Veera who always pointed me in the right direction at critical point in my research. He has encouraged me by talking about good side of research results.

I thank my committee members, Dr. Scott, Dr. Ensor, and Dr. Nakhleh, for being my committee members and giving me good feedback. I thank the department of statistics for supporting my tuition for 7 years and first years of stipend. I thank Dr. Allen for being my committee for thesis proposal. Much of my research is based on the material I learned from her class. I thank Garrett and Jaime, who were the same year as me. Thanks to them, I could enjoy my first years of Ph.D study.

I appreciate my husband, Hyun Jung, for endless support in terms of everything. I thank my daughter, Erin. She always makes me smile. Most of all, I thank my lord, Jesus Christ, for his endless patient and encouragement. Whenever I doubt about finishing Ph.D. he has always convinced me to continue it. He is my best advisor and friend. I devote my dissertation to Jesus Christ, my lord.
Contents

Abstract ii
List of Illustrations vii
List of Tables x

1 Introduction 1

2 A regression model for marker selection and overview of high dimensional regression models 6
   2.1 Logistic regression for biomarker selection .............................. 6
   2.2 Penalized methods .................................................................. 7
   2.3 Tuning parameter selection methods ......................................... 8
       2.3.1 Cross-validation .............................................................. 9
       2.3.1.1 Comparison of variable selection methods for biomarker selection using simulated data sets .................. 12
   2.3.2 Stability Selection (SS) ....................................................... 15
   2.4 Data analysis: Stability selection application to the elastic net .... 19
       2.4.1 Simulation studies ............................................................ 19
       2.4.2 The BATTLE data analysis .............................................. 24
   2.5 Simulation studies: Investigation of different cutoffs of SS and comparison with CV ........................................... 26

3 Prediction Oriented Marker Selection (PROMISE) 31
   3.1 Joint optimization based on CV and SS ..................................... 31
   3.2 Application for selecting more than one regularization parameter . 34
3.3 Illustrative examples ............................................. 35
3.4 Simulation studies based on real data .......................... 41
  3.4.1 Thirteen signals .............................................. 41
  3.4.2 No signal .................................................... 44
3.5 BATTLE data analysis ........................................... 44

4 PROMISE application to the group lasso ....................... 52
  4.1 Method ......................................................... 52
  4.2 BATTLE data simulation ..................................... 54
  4.3 BATTLE data analysis ........................................ 59

5 Discussion and Future direction ................................. 65

Bibliography ......................................................... 67
2.1 Prediction Accuracy: AUC (Area Under the ROC curve)  
(a) $s=3, n=100$  
(b) $s=6, n=200$  
(c) $s=12, n=400$

2.2 Probability of selecting at least $r$ number of important variables:
among 1000 runs, how many times at least $r$ number of variables are selected

(a) $s=3, n=100, \rho = 0.8$  
(b) $s=3, n=100, \rho = 0.5$  
(c) $s=3, n=100, \rho = 0.3$  
(d) $s=6, n=200, \rho = 0.8$  
(e) $s=6, n=200, \rho = 0.5$  
(f) $s=6, n=200, \rho = 0.3$  
(g) $s=12, n=400, \rho = 0.8$  
(h) $s=12, n=400, \rho = 0.5$  
(i) $s=12, n=400, \rho = 0.3$

2.3 Average number of falsely chosen variables when selecting at least $r$
number of the important variables

(a) $s=3, n=100, \rho = 0.8$  
(b) $s=3, n=100, \rho = 0.5$  
(c) $s=3, n=100, \rho = 0.3$  
(d) $s=6, n=200, \rho = 0.8$
(e) $s=6, n=200, \rho = 0.5$ ........................................... 23
(f) $s=6, n=200, \rho = 0.3$ ........................................... 23
(g) $s=12, n=400, \rho = 0.8$ ........................................ 23
(h) $s=12, n=400, \rho = 0.5$ ........................................... 23
(i) $s=12, n=400, \rho = 0.3$ ........................................... 23

3.1 Algorithm of PROMISE ............................................. 39
3.2 Comparison of variable selection paths using the elastic net ($\alpha = 0.4$) on the simulated data ........................................ 40
   (a) Regularization path ........................................... 40
   (b) Stability path using SS ....................................... 40
   (c) Stability path using PROMISE ................................. 40
3.3 Comparison of PROMISE (triangle) with CV (circle) for (a) the lasso and (b) the elastic net using the BATTLE data. Each point represents the mean value and the error bar represents 2 standard errors. ........................................... 46
   (a) Lasso ......................................................... 46
   (b) Elastic net .................................................... 46
3.4 Canonical pathways enriched for selected genes from the BATTLE data analysis ........................................... 51

4.1 Absolute value of coefficients for variables with nonzero coefficients for BATTLE simulation 1, 2, and 3 ........................................... 56
4.2 Group selection accuracy of BATTLE simulations ..................... 61
   (a) Simulation1 .................................................... 61
   (b) Simulation2 .................................................... 61
   (c) Simulation3 .................................................... 61
4.3 Variable selection accuracy of BATTLE simulations using the group lasso ........................................ 62
   (a) Simulation1 ...................................................... 62
   (b) Simulation2 ...................................................... 62
   (c) Simulation3 ...................................................... 62

4.4 Variable selection accuracy of BATTLE simulations using the group lasso with the lasso ............................ 63
   (a) Simulation1 ...................................................... 63
   (b) Simulation2 ...................................................... 63
   (c) Simulation3 ...................................................... 63

4.5 Comparison of Positive predictive value of BATTLE simulations using the group lasso ................................. 64
Tables

2.1 Markers and response rates ............................................. 13
2.2 Comparison of variable selection methods for identifying binary predictive markers .................................................. 14
2.3 Summary of the procedure for stability selection with the elastic net/the lasso .......................................................... 17
2.4 The battle data analysis using CV and Stability selection with $\pi_{th} = 0.6$ and $v = 5$ .......................................................... 25
2.5 Result of simulations: comparison of stability selection (SS) with 3 combinations of the cutoffs and cross validation (CV) using the lasso ........................................................ 29
2.6 Result of simulations: Comparison of stability selection (SS) with 3 combinations of the cutoffs and cross validation (CV) using the elastic net .................................................. 30

3.1 Treatment coding ............................................................. 37
3.2 Comparison of PROMISE, CV, and SS using the BATTLE simulation data for (a) the lasso and (b) the elastic net .......... 42
(a) Lasso ........................................................................... 42
(b) Elastic net ..................................................................... 42
3.3 Comparison of PROMISE, CV, and SS for (a) the lasso and (b) the elastic net when there is no signal ...................... 44
(a) Lasso ........................................................................... 44
(b) Elastic net ..................................................................... 44
3.4 List of gene/probe sets selected from the BATTLE trial analysis . . . 49
3.5 Top 9 most frequently selected genes from the BATTLE data analysis
and their association with cancer . . . . . . . . . . . . . . . . . . . . 50

4.1 Prediction accuracy (AUC) of BATTLE simulations . . . . . . . . 57
4.2 PPV of BATTLE simulations . . . . . . . . . . . . . . . . . . . . 57
4.3 Specific results of BATTLE simulation 3 . . . . . . . . . . . . . . 58
4.4 Result of BATTLE data analysis . . . . . . . . . . . . . . . . . . . 59
Chapter 1

Introduction

Recent advances in genomic technologies have enabled us to obtain large amounts of detailed biological information from single patient samples. This has led to challenges in identifying important biological information (biomarkers) associated with relevant clinical/disease outcomes from such high-throughput data sets. Biomarkers can be broadly classified into two groups: prognostic markers and predictive markers [1]. A prognostic marker is used to predict the patient’s disease outcome regardless of the choice of treatment. For example, a woman with a mutation of the BRCA gene, a prognostic marker, has higher risk of developing breast and ovarian cancer [2]. A predictive marker is used to predict the likelihood of benefit from a specific treatment. For example, in lung cancer, patients with mutations of the epidermal growth factor receptor (EGFR) are predicted to have higher response rates to erlotinib treatment than patients without EGFR mutations [3, 4]; therefore, EGFR status is a predictive marker for lung cancer. The search for such dual marker profiles comprises the holy grail of personalized medicine – one of the most important problems in cancer research today [5]. Personalized medicine aims to provide specific treatments for patients who have certain biomarkers, giving them therapies that are more effective and less toxic than the standard treatment.

Statistically, the task of selecting biomarker profiles becomes a variable selection problem in regression settings. Prognostic markers are identified through the main effects of the biomarkers on clinically relevant responses such as disease sta-
Predictive markers are detected through the observation of treatment outcomes by including marker-by-treatment interaction terms, for which the outcome is often the treatment response or survival times. However, the search for the optimal combination of the main effect and interaction terms substantially increases the search space of the variables. Also, identifying predictive markers requires the use of clinical trial data along with the patients’ genomic data, which were not routinely collected until recent years [6]. Due to these challenges, little published work is available on the identification of predictive markers. To date, most predictive markers have been developed on the basis of purely biological findings and have not been identified by statistical algorithms. As a result, few predictive markers have been approved by the FDA [7]. In addition, purely biologically driven candidate markers may not predict the treatment outcome [3] and need to be validated in clinical trials with statistical rigor.

Our methods are motivated by a clinical trial, the “Biomarker-integrated Approaches of Targeted Therapy for Lung Cancer” (BATTLE) trial, which was one of the first biomarker-based clinical trials [3]. The goals of that trial were to test treatments and prognostic and predictive marker effects. Four biomarker groups were used in the trial: *EGFR* mutation/amplification, mutations of the genes *KRAS* and *BRAF*, expression of the vascular endothelial growth factor (VEGF) and VEGR receptor (VEGFR), and expression of the cyclin D1/retinoid X receptor (RXR). Four targeted therapies were used in the trial: erlotinib, vandetanib, erlotinib plus bexarotene, and sorafenib. One therapy targeted one of the four biomarker groups. The four biomarkers were chosen before the trial on the basis of biological and clinical information. The marker status of each patient was then used to randomize the patients to one of the four treatments. Although some predictive markers were found to be useful, sev-
eral of the pre-selected markers could not predict the outcome of their companion
treatments: patients with and without the putative predictive markers had similar
treatment outcomes [3]. In the trial, no statistical knowledge was involved when the
candidate predictive markers were selected. That trial raised the awareness of the
importance of incorporating statistical processes to identify predictive markers. As
a result, in the BATTLE II trial, which is ongoing at MD Anderson Cancer Cen-
ter, both biological and statistical knowledge are involved when identifying predictive
markers. In the first stage, KRAS mutation, which was a putatively biologically
relevant marker, was used to adaptively randomize 200 patients to one of the four
treatments. Then, a statistical method will be applied to the data from the first stage
to select biomarkers. In the second stage, the statistically chosen biomarkers will be
used to guide the assignment of treatments for the next 200 patients [8]. The next
statistical question is how to develop a variable method that can be used to choose
important markers to use in identifying the best treatment for each patient according
to the patients biomarker profile.

More generally, an ideal statistical method should have the following attributes:

(a) Parsimony - A method should select a smaller set of highly (biologically)
relevant biomarkers. For a biomarker to be clinically approved (by the FDA), bio-
logical and clinical validations are critical but can be costly. Hence, selecting numer-
ous biomarkers for further validation and experimentation is prohibitively expensive.
Also, a finite amount of tissue samples limits the number of biomarkers that can be
tested.

(b) Selection accuracy - Even if a method produces a parsimonious model, if it
misses important variables, it is not an appropriate method for marker selection. We
need to consider whether a method selects important variables as well as ignores the
unimportant variables.

(c) Prediction accuracy for future data - The selected markers should be able to accurately predict patient responses for all the possible treatments that are involved in a clinical trial. This is because the predicted probability of response guides the choice of treatment: the treatment that is expected to yield the highest response rate based on the patients individual biomarker profiles has a higher probability of being assigned to a patient. If the prediction is not accurate, patients would be unlikely to be assigned to the most beneficial treatments in the trial.

(d) Ability to handle correlated data - Biomarkers (or genes) often tend to be correlated based on their biological functions and complex organization. For example, genes that are part of a common biological pathway share some functional similarly and the correlation between such genes can be high [9]. This correlation should be handled by an appropriate statistical method.

In the statistics literature, there is limited research in identifying predictive markers in high-dimensional settings. Werft et al.[6] used p-values for the permutation of the regressor residual test to determine the significance of individual predictive markers. That test can handle correlated variables. However, the paper showed only the false discovery rate and did not show results related to the type II error rate. That test is designed to select variables and does not address the prediction of future outcomes.

Gu et al.[10] proposed a method with a group lasso followed by adaptive lasso for marker selection. Each group consists of a marker and the marker-by-treatment interaction effects. However, since highly correlated genes in the same pathway are in different groups in this setting, the correlation between groups can be high. The group lasso cannot deal with such strong correlations between groups [11]. Also, the
paper focused on only variable selection, not prediction of treatment outcome.

To deal with high-dimensional variable selection for correlated data, one of the most commonly used penalization methods is the elastic net [9] for selecting predictive markers. However, finding the right amount of penalty is critical and challenging because selecting a penalty that is too small results in keeping all the variables, and selecting a penalty that is too large results in having no variables detected in the model. To find the regularization parameter, cross-validation (CV) [12] is commonly used: it selects the parameter based on prediction accuracy. CV often gives good prediction accuracy, but selects a model with too many variables, resulting in a high false positive rate [13]. An alternative method, stability selection (SS) was proposed to control the number of false positives [14]. It produces a sparse solution. However, two critical cutoffs in SS, which are theoretically determined for controlling false discoveries, yield results that are too conservative to identify important variables [15, 16]. As we show in the results, a consequence is that the model that uses SS to select variables shows low prediction accuracy due to the small number of true positives being detected.

To achieve the advantages of both methods, having good prediction accuracy and a small number of false positives, we propose prediction-oriented marker selection (PROMISE), which is a combination of SS and CV. In PROMISE, the sub-sampling method of SS acts to reduce the false discoveries and produce a sparse solution. To select the cutoff of SS, the cross-validation method acts to maximize the prediction accuracy and true positive rate. Our PROMISE algorithm is general and can be applied to any penalization method; we illustrate out methods lasso and elastic net for individual marker selection and the group lasso for the pathway selection. We provide our R code for public use.
Chapter 2

A regression model for marker selection and overview of high dimensional regression models

2.1 Logistic regression for biomarker selection

For prognostic and predictive marker selection, logistic regression is a standard model for a binary treatment outcome and where the predictors are individual markers, treatment indicator variables, and their interaction effects [17]. Suppose for the $i$th patient, we let $p_i$ denote the probability of the treatment response, $T_{ij}$ denote the $j$th treatment indicator, considering treatment 1 is the reference group, and $M_{ik}$ denote the $k$th marker value. When the number of patients is $n$, the number of treatments is $J$, and the number of markers is $K$, the basic model can be written as

$$\text{logit}(p_i) = \alpha_1 + \sum_{k=1}^{K} \eta_k M_{ik} + \sum_{j=2}^{J} \alpha_j T_{ij} + \sum_{k=1}^{K} \sum_{j=2}^{J} \gamma_{kj} M_{ik} T_{ij}. \quad (2.1)$$

The $j$th treatment is considered to be significant compared to the reference treatment if $\alpha_j \neq 0$ for $j = 2, 3, ..., J$. The $k$th marker is considered to be a prognostic marker if $\eta_k \neq 0$. Also, the $k$th marker is considered to be a predictive marker for the $j$th treatment if $\gamma_{kj} \neq 0$.

To make a general form of the logistic regression, the right side of equation (2.1) can be written as $\beta_0 + \sum_{i=1}^{p} \beta_i x_{ii}$, where $p = (J-1)K + J - 1$. In our data, $K$ is about 30000, the number of probe sets in the microarray data, and $J$ is 4, so $p$
is more than 120000. In this setting, the biomarker selection problem is a variable selection problem in ultrahigh dimensions. A popular way of selecting variables is using a penalized regression such as the lasso or the elastic net.

2.2 Penalized methods

When the response is binary, we maximize the penalized log likelihood,

$$\max_{(\beta_0, \beta) \in \mathbb{R}^{p+1}} \left[ \sum_{i=1}^{n} \left\{ y_i \log p_i + (1 - y_i) \log (1 - p_i) \right\} - P_\lambda(\beta) \right],$$

where

$$p_i = \frac{\exp(\beta_0 + \sum_{j=1}^{p} x_{ij} \beta_j)}{1 + \exp(\beta_0 + \sum_{j=1}^{p} x_{ij} \beta_j)}$$

is the probability of having a response of 1, and $P_\lambda(\beta)$ is a penalty function.

The lasso penalty [18], one of the most popular penalties for variable selection, is

$$P_\lambda(\beta) = \lambda ||\beta||_1 = \lambda \sum_{j=1}^{p} |\beta_j|.$$  \hspace{1cm} (2.2)

This L1 penalty shrinks some of the coefficients to exactly zero. Therefore, it automatically selects variables without considering multiple testing issues. However, it is known to select a singleton variable when the covariates are highly correlated [9].

The ridge penalty [19] is another well-known penalty method.

$$P_\lambda(\beta) = \lambda ||\beta||_2^2 = \lambda \sum_{j=1}^{p} \beta_j^2$$

This ridge regression handles the multicollinearity problem well, but the L2 penalty does not shrink some of the coefficients to exactly zero [9]. Therefore, this method is
not suitable for variable selection.

The elastic net penalty [9] is a combination of the lasso ($\alpha = 1$) and ridge penalty ($\alpha = 0$) methods.

$$P_{a,\lambda}(\beta) = \lambda \left\{ (1 - \alpha) \frac{1}{2} \|\beta\|_2^2 + \alpha \|\beta\|_1 \right\}$$

$$= \lambda \left\{ \sum_{j=1}^{p} \left[ \frac{1}{2} (1 - \alpha) \beta_j^2 + \alpha |\beta_j| \right] \right\} \quad (2.3)$$

Thus, it has the advantages of both methods: it simultaneously selects a group of variables that are correlated with each other, while providing automatic variable selection [9]. As genes can be correlated with each other when they share the same biological pathway [20] or have similar functionality, we apply the elastic net to choose the markers.

### 2.3 Tuning parameter selection methods

The challenge in fitting these models is the choice of the regularization parameter(s): $\lambda$ for the lasso in equation (2.2) and ($\alpha$, $\lambda$) for the elastic net in equation (2.3). This is critical issue because variable selection largely depends on the regularization parameter. When the parameter is too large, no variable is selected and when the parameter is too small, all variables are included. Two major strategies exist for choosing the regularization parameters, cross-validation and stability selection that are summarized below.

The challenge in fitting these models is the choice of the regularization parameter(s): $\lambda$ for the lasso in equation (2.2) and ($\alpha$, $\lambda$) for the elastic net in equation (2.3). This is a critical issue because variable selection largely depends on the regularization parameter. When the parameter is too large, no variable is selected and when
the parameter is too small, all variables are included. Two major strategies exist for choosing the regularization parameters, cross-validation and stability selection; these strategies are summarized below.

2.3.1 Cross-validation

Cross-validation (CV) is one of the methods most widely used to select the regularization parameter. CV uses part of the data to fit the model and the other part to test the model [21]. We use an example to explain the CV procedure in the lasso.

To perform K-fold CV, we split the data into K roughly equal sizes. For a candidate set of the regularization parameters, \( \lambda_r = \lambda_1, ..., \lambda_R \), we fit the lasso, selecting variables and estimating regression coefficients, using all the data except for k-fold of the data. The estimated coefficient of the lasso using \( \lambda_r \), computed with the data except for the k part is

\[
\hat{\beta}^{\text{lasso}(-k)}(\lambda_r) = \arg \max_{\beta} \left[ \sum_{i \notin k} y_i \beta^T x_i - \log(1 + \exp(\beta^T x_i)) - \lambda ||\beta||_1 \right],
\]

where \( \beta = (\beta_0, \beta_1, ..., \beta_p)^T \) and \( x_i \) includes the constant term 1 to include the intercept.

Then, with the k-fold of data, we predict the probability of response using the predicted coefficients for each regularization parameter. For \( i \in k \),

\[
\hat{p}_i(\lambda_r) = \frac{\exp((\hat{\beta}^{\text{lasso}(-k)}(\lambda_r))^T x_i^{(k)})}{1 + \exp((\hat{\beta}^{\text{lasso}(-k)}(\lambda_r))^T x_i^{(k)})}.
\]

Then, we calculate the prediction accuracy, measured by the area under the receiver operating characteristic curve (AUC), for example, using the response in the k-fold of data. We perform this procedure for k=1,...,K and then calculate the aver-
age prediction accuracy for each regularization parameter. The K-fold cross-validated AUC estimator to select \( \lambda \) in the lasso is

\[
CV_{\text{AUC}} = \frac{1}{K} \sum_{k=1}^{K} \text{AUC}(y^{(k)}, \hat{p}^{(k)}(\lambda_r)),
\]

where \( y^{(k)} \) is a vector of \( y_i \) for \( i \in k \) and \( \hat{p}^{(k)}_i \) is a vector of \( \hat{p}_i \) for \( i \in k \).

We choose the regularization parameter that yields a model that maximizes \( CV_{\text{AUC}} \) (the maximum rule) or which gives the most parsimonious model whose \( CV_{\text{AUC}} \) is not more than one standard error (1SE) difference from the model established by the maximum rule (1SE rule) [21]. After selecting the regularization parameter, the entire data set is used to fit the lasso with the parameter for the final model.

**Calibration: simultaneous selection of parameters (\( \alpha \) and \( \lambda \)) for the elastic net.** The glmnet package [22] is the most popular package to use the lasso and the elastic-net due to high speed even in high dimensional data. It provides a solution using CV when \( \alpha \) is given by a user. However, it does not provide simultaneous selection of \( \alpha \) and \( \lambda \) for the elastic-net using CV. (Any existing package provides simultaneous selection of the parameters by CV.) However, selecting right amount of \( \alpha \) is as difficult as selecting \( \lambda \): when \( \alpha \) is too small, the elastic net selects too many variables, and when \( \alpha \) is too large, it is the same as the lasso, which does not select correlated variables simultaneously. To resolve the issue of selection of \( \alpha \), we developed an algorithm that provides simultaneous selection of those two parameters using CV based on the glmnet package. The simultaneous selection steps are the following.

To use the minimum rule to choose the \( \alpha \) and the \( \lambda \), it is straightforward; first, we
select a set of $\alpha$. We choose a set of $\alpha = (0.1, 0.2, ..., 1)$. $\alpha = 0$ is excluded because this ridge penalty does not select variables. Second, for each $\alpha$, we select $\lambda$ that minimizes CV error and corresponding CV error. Then, we get 10 pairs of $(\alpha, \lambda)$ and their corresponding CV errors. Third, among the 10 pairs, we select the pair of $(\alpha, \lambda)$ that gives the smallest CV error. Let this model denote the ‘best’ model. However, to use 1se rule, selecting the pair of $(\alpha, \lambda)$ is a little bit tricky.

To use the 1SE rule, we need to define what is the most parsimonious model in the elastic-net. The more parsimonious model is, the fewer variables are included. Therefore, as $\alpha$ increases and $\lambda$ increases, the model becomes more parsimonious. Therefore, given $\alpha$, the largest $\lambda$ gives the most parsimonious model. Likewise, given $\lambda$, the largest $\alpha$ gives the most parsimonious model. However, we don’t fix any of these two parameters. Therefore, among the models whose CV error is not more than 1 SE difference of the smallest CV error using the ‘best’ model, we need to check how many variables are actually selected using each pair of parameters. Then, we choose the pair of parameters that selects the smallest number of variables. Details are the following.

The candidate $\alpha$s are $(0.1, 0.2, ..., 1)$. For each $\alpha$, we choose the largest $\lambda$ whose CV error is within 1 SE of the smallest CV error using the ‘best’ model if it exists. If it does not exist, the $\alpha$ is dropped for a candidate. Using these at most 10 pairs of parameters, we build models and see how many variables are included. Among the pairs, we select the pair which gives a model with the smallest number of variables. If more than one pair of these parameters give models with the same smallest number of variables, we select the one with the largest $\lambda$ regardless $\alpha$, assuming that the degree of parsimony is similar each other since these select the same number of variables.

CV flexibly selects the parameter based on the prediction accuracy; therefore, it
provides a satisfactory result for prediction accuracy. Also, to maximize the prediction accuracy, the model selected by CV includes the important variables. However, the model size is often too large and includes too many false positives [14].

2.3.1.1 Comparison of variable selection methods for biomarker selection using simulated data sets

In this section, we compare the lasso, the elastic net, and smoothly clipped absolute deviation (SCAD) penalty [23], AIC, and BIC for predictive and prognostic marker selection. We applied CV to select the penalty parameters for the lasso, the elastic net, and SCAD. We used 5 folds CV which select the model with the maximum mean of AUC. We used our simulation data sets for the comparison.

For the simulation, we generated \( n = 200 \) among which \( n=150 \) are used for training data set and \( n=50 \) are used for the test data set. 4 treatments were randomly assigned to patients. There were 100 biomarkers, which were all binary with 50% of marker positive. These were generate based on multivariate normal distribution. Here, biomarkers and treatments were coded with dummy coding (0 and 1). We generated coefficients based on response rate. The response rate is shown in Table 2.1 and corresponding model is

\[
\text{logit}(p) = -1.39(2.77M_1 + 1.79M_2 + 0.98M_3)T_2 + (2.77M_4 + 1.79M_5)T_3
+ 2.77M_6T_4 + 2.77M_7.
\]

The predictive markers for the same treatment are correlated each other (M1-M3, M4-M5) with correlation coefficients, \( \rho \) where \( \rho = 0, 0.5, \text{and} 0.9 \) for 3 different scenarios. The reason why we want to include correlated biomarkers is that Biomarkers (or
Table 2.1: Markers and response rates

<table>
<thead>
<tr>
<th></th>
<th>Trt1</th>
<th>Trt2</th>
<th>Trt3</th>
<th>Trt4</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>0.2</td>
<td>0.8</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>M2</td>
<td>0.2</td>
<td>0.6</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>M3</td>
<td>0.2</td>
<td>0.4</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>M4</td>
<td>0.2</td>
<td>0.2</td>
<td>0.8</td>
<td>0.2</td>
</tr>
<tr>
<td>M5</td>
<td>0.2</td>
<td>0.2</td>
<td>0.6</td>
<td>0.2</td>
</tr>
<tr>
<td>M6</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td>0.8</td>
</tr>
<tr>
<td>M7</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
</tr>
</tbody>
</table>

genes) often tend to be correlated based on their biological functions and complex organization such as pathways [9]. The markers outside of these groups are independent each other.

The simulation results are shown in Table 2.2. First, in all scenarios, the elastic-net works best in identifying important markers, showing highest true positive rate among the 5 methods. Especially, it detects the largest number of true positives when the important markers are highly correlated; but the other penalized methods identifies the least number of true positives this highly correlated scenario compared to low or medium correlation scenarios. However, the downside of the elastic net is having larger number of false positives than the other penalized methods: even with the most true positives among the 3 penalized methods, it shows lowest positive predictive value (PPV) except for highly correlated scenario.

Second, the 3 penalization methods works better than AIC and BIC in terms for prediction accuracy and variable selection accuracy in all scenarios. In terms of variable selection accuracy, the difference between the penalization methods and AIC/BIC is bigger when the significant variables are more correlated.

Third, in terms of prediction accuracy, all three penalization methods behave similarly.
Table 2.2: Comparison of variable selection methods for identifying binary predictive markers

<table>
<thead>
<tr>
<th>Method</th>
<th>AUC</th>
<th>TP</th>
<th>FP</th>
<th>TPR</th>
<th>PPV</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho = 0$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lasso</td>
<td>0.74</td>
<td>3.91</td>
<td>20.27</td>
<td>0.56</td>
<td>0.27</td>
</tr>
<tr>
<td>Elastic-net</td>
<td>0.74</td>
<td>4.29</td>
<td>30.72</td>
<td>0.61</td>
<td>0.23</td>
</tr>
<tr>
<td>SCAD</td>
<td>0.75</td>
<td>3.96</td>
<td>12.84</td>
<td>0.57</td>
<td>0.29</td>
</tr>
<tr>
<td>AIC</td>
<td>0.64</td>
<td>3.22</td>
<td>16.64</td>
<td>0.46</td>
<td>0.16</td>
</tr>
<tr>
<td>BIC</td>
<td>0.66</td>
<td>3.13</td>
<td>12.84</td>
<td>0.45</td>
<td>0.21</td>
</tr>
<tr>
<td>$\rho = 0.5$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lasso</td>
<td>0.77</td>
<td>3.99</td>
<td>11.84</td>
<td>0.57</td>
<td>0.46</td>
</tr>
<tr>
<td>Elastic-net</td>
<td>0.77</td>
<td>5.08</td>
<td>17.72</td>
<td>0.73</td>
<td>0.43</td>
</tr>
<tr>
<td>SCAD</td>
<td>0.78</td>
<td>4.37</td>
<td>8.21</td>
<td>0.62</td>
<td>0.47</td>
</tr>
<tr>
<td>AIC</td>
<td>0.66</td>
<td>3.6</td>
<td>15.72</td>
<td>0.51</td>
<td>0.19</td>
</tr>
<tr>
<td>BIC</td>
<td>0.69</td>
<td>3.51</td>
<td>11.67</td>
<td>0.5</td>
<td>0.26</td>
</tr>
<tr>
<td>$\rho = 0.9$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lasso</td>
<td>0.79</td>
<td>3.27</td>
<td>7.56</td>
<td>0.47</td>
<td>0.55</td>
</tr>
<tr>
<td>Elastic-net</td>
<td>0.79</td>
<td>5.46</td>
<td>12.52</td>
<td>0.78</td>
<td>0.57</td>
</tr>
<tr>
<td>SCAD</td>
<td>0.8</td>
<td>3.92</td>
<td>6.86</td>
<td>0.56</td>
<td>0.5</td>
</tr>
<tr>
<td>AIC</td>
<td>0.68</td>
<td>3.33</td>
<td>15.28</td>
<td>0.48</td>
<td>0.18</td>
</tr>
<tr>
<td>BIC</td>
<td>0.72</td>
<td>3.27</td>
<td>10.71</td>
<td>0.47</td>
<td>0.26</td>
</tr>
</tbody>
</table>

The mean and standard error of AUC, true positives (TP), false positives (FP), true positive rate (TPR), and positive predictive value (PPV). Bold and italic numbers are the best results.

All in all, the elastic-net works the best in terms of identifying important markers and predicting treatment outcome. However, selecting a large number of false positives is critical downfall because biological verification of biomarkers is costly and time consuming. Therefore, to resolve this issue, in the next chapter, we want to introduce stability selection to the elastic net to reduce false positives as a regularization parameter method. Before than we describe the basic concept of stability selection and then how to apply it to the elastic net.
2.3.2 Stability Selection (SS)

 Whereas the goal of CV is to maximize the prediction accuracy on future data sets, the goal of SS is to control false discoveries under a desired level [14].

 Let $I$ denote a random sample of size $\lfloor n/2 \rfloor$ from the observed data without replacement. We are interested in finding the set of nonzero regression coefficients, $S = \{ k : \beta_k \neq 0, k = 1, \ldots, p \}$.

 For each variable $k \in 1, \ldots, p$ and for each $\lambda$, we estimate the selection probability using sub-samples [24]:

 $$\hat{\Pi}_k^\lambda = \frac{1}{B} \sum_{m=1}^{B} \mathbb{1}\{ k \in \hat{S}^\lambda(I_m) \},$$

 where $B$ is the number of sub-sampling and $\hat{S}^\lambda(I_m)$ is the selected variables using $\lambda$, with the $m$th random sub-sample $I_m$ of size $\lfloor n/2 \rfloor$, and $\mathbb{1}$ is an indicator function.

 For a cutoff $0 < \pi_{thr} < 1$ and a set of regularization parameters $\Lambda$, the set of selected variables is

 $$\hat{S}_{\text{stable}} = \left\{ k : \max_{\lambda \in \Lambda} (\hat{\Pi}_k^\lambda) \geq \pi_{thr} \right\}.$$  \hfill (2.4)

 The next question is how to select $\pi_{thr}$ and the choice of the range of $\lambda$, $\Lambda$, in equation (2.4). Meinshausen and Buhlmann [14] claimed that the choice of $\pi_{thr}$ in range $(0.6, 0.9)$ does not affect the result of the variable selection. They set 0.9 as a default cutoff. Once a cutoff is chosen, $\Lambda$ is determined by the desired number of false positives, according to Meinshausen and Buhlmann [14]. For $\pi_{thr} \in (0.5, 1)$,

 $$E(V) \leq \frac{1}{2\pi_{thr} - 1} \frac{q_\Lambda}{p},$$  \hfill (2.5)

 where $V$ is the number of false discoveries and $q_\Lambda$ is the average number of selected
variables, \( E(\lvert \hat{S}^\lambda (I) \rvert) \) where \( \hat{S}^\lambda = \bigcup_{\lambda \in \Lambda} \hat{S}^\lambda \).

Let \( q \) denote the \( q_\Lambda \) when it achieves the equality, \( q = \sqrt{vp(2\pi_{\text{thr}} - 1)} \): it is the maximum average number of selected variables when the desired average false positive rate is under a certain level, say, \( v \). Then, given \( v \) and \( \pi_{\text{thr}} \), \( q \) determines the range of lambda, \( \Lambda \).

For example, for the default cutoff value \( \pi_{\text{thr}} = 0.9 \) and desired error control \( E(V) \leq 1 \), we choose \( \Lambda \) such that \( q = \sqrt{0.8p} \). Meinhausen and Buhlmann [14] suggested selecting \( \lambda_{\text{max}} \) and \( \lambda_{\text{min}} \) such that \( \bigl| \bigcup_{\lambda_{\text{max}} \geq \lambda \geq \lambda_{\text{min}}} \hat{S}^\lambda \bigr| \leq q \). However, obtaining the left side of the inequality is computationally very demanding. We simplify it as \( \bigl| \bigcup_{\lambda_{\text{max}} \geq \lambda \geq \lambda_{\text{min}}} \hat{S}^\lambda \bigr| = \lvert \hat{S}^{\lambda_{\text{min}}} \rvert \), because for a small enough \( \lambda \), \( \hat{S}^\lambda \subseteq \hat{S}^{\lambda'} \) for all \( \lambda \geq \lambda' \) [14]. Hence, in the range of \( \lambda \), we consider only \( \lambda_{\text{min}} \) to control the false discoveries.

In this way, we select \( \lambda_{\text{min}} \) to control \( E(V) \) under \( v \) in the following way:

\[
q_\Lambda = E(\lvert \hat{S}^\lambda (I) \rvert) = E(\lvert \hat{S}^{\lambda_{\text{min}}} (I) \rvert) = \frac{1}{B} \sum_{m=1}^{B} \lvert \hat{S}^{\lambda_{\text{min}}} (I_m) \rvert \leq q = \sqrt{vp(2\pi_{\text{thr}} - 1)} \tag{2.6}
\]

Therefore, in SS, only two parameters, \( \pi_{\text{thr}} \) and \( \lambda_{\text{min}} \), play a critical role in determining the final set of variables.

SS originally developed based on the lasso [14]. However, because of possibility of high collinearity between genes, we applied SS to the elastic net. Kirk and Stumpf [25] briefly mentioned the possibility of application of elastic-net to SS. We also simultaneously choose \( \alpha \) and \( \lambda \) as we do for CV. For this, we first choose \( \alpha \) in equation (2.3) using cross validation (1SE rule); we select the combination of \( \alpha \) and \( \lambda \) using CV and just used \( \alpha \). Then using the \( \alpha \), SS is applied to choose variables. The detailed procedure for SS is in Table 2.3.

The sub-sampling procedure in SS controls the false discoveries. However, the
1. Decide a cutoff, \( \pi_{\text{thr}} \in (0.5, 1) \) and the number of false positives desired to be controlled, \( v \). Calculate \( q \) from (2.6).

2. Select \( \alpha \) using 1SE rule with CV for the elastic-net or set \( \alpha = 1 \) for the lasso.

3. Preselect a range of \( \lambda \) (use glmnet function to get this range using entire training data)
   - (a) The maximum \( \lambda \) is the smallest \( \lambda \) that chooses no variables
   - (b) The minimum \( \lambda \) is the maximum \( \lambda \times 0.001 \)
   - (c) The range of \( \lambda \) is a sequence from maximum \( \lambda \) to minimum \( \lambda \) with \( R \) values on log scale

4. Repeat sub-sampling procedure with size \( \lfloor n/2 \rfloor \) 100 times
   - (a) Select variables with non-zero coefficients with the elastic net or the lasso using the range of \( \lambda \)
   - (b) Record how many times each variable is selected for each \( \lambda \) ⇒ Calculate the average over \( B \) sub-sampling procedure (\( \bar{\Pi}_k^\lambda \) in equation (2.3.2))
   - (c) Record how many variables are selected using each \( \lambda \) ⇒ Calculate the average over \( B \) sub-sampling procedure (\( q_\lambda = E(|\hat{S}_k^\lambda(I)|) \))

5. Get final set of variables
   - (a) Limit \( \lambda \) which selected variables not more than \( q \) on average (i.e. limit \( \lambda \) such that \( q_\lambda < q \) in 4c)
     - Final range of \( \lambda \)
       - \( \lambda_{\text{min}} \) is the largest \( \lambda \) that chooses less than or equal to \( q \) number of variables on average
       - \( \lambda_{\text{max}} \) is the smallest \( \lambda \) that chooses no variables
   - (b) Select the variables whose maximum selection probability in 4b is equal or greater than \( \pi_{\text{thr}} \) in the final range of \( \lambda \); see equation (2.4)

---

Table 2.3: Summary of the procedure for stability selection with the elastic net/the lasso
selection of two cutoffs ($\pi_{\text{thr}}$ and $\lambda_{\text{min}}$) based on inequality (2.5) or inequality (2.6) yields results that are too conservative [15, 16]. Therefore, it excludes quite a few important variables. SS was developed to select variables and does not build a model from the variables. Therefore, Meinshausen and Buhlmann [14] did not report prediction accuracy using the selected variables when using SS. However, it is obvious that a model that does not include a considerable number of the important variables cannot predict outcome well.

Even if we want to decide a generous cutoff to include enough number of important variables, it is difficult to decide it because it depends on the type of data.
2.4 Data analysis: Stability selection application to the elastic net

2.4.1 Simulation studies

The purpose of this simulation study is to show that 1) using stability selection, the number of variables is under control (usually under 1) 2) the combination of the elastic-net and stability selection makes it possible to have high prediction accuracy and true positive rate while having few false positives. We compare the lasso with CV, the lasso with stability selection, the elastic net with CV, the elastic net with stability selection.

There are 4 treatments, the first of which is the reference treatment. We assume a randomized clinical trial, meaning that one of four treatments is randomly assigned to a patient. For simplicity, we assume that there is no predictive marker for the reference treatment. The number of markers is 250. The first \( s \) number of markers are significant and these are all predictive markers (the markers that are not marginally significant but significant in their interaction term with a treatment). The markers are generated from multivariate normal distribution. Each marker has mean 0 and variance 1. The predictive markers for the same treatment are correlated with each other with correlation coefficient \( \rho \). The correlations outside of these groups are 0.2

We varied the number of significant markers \( s \) and within the group correlation \( \rho \). \( s = 3, 6, 12 \) and \( \rho = 0.8, 0.5, 0.3 \). When \( s = 3 \), only the second treatment has 3 predictive markers. When \( s = 6 \), only the second and third treatments each have 3 predictive markers. When \( s = 12 \), there are 3 number of predictive markers for each treatment except for the first treatment.

The total number of markers \( (K) \) is 250. Therefore, the total number of variables
(p) is 1003 including interaction terms. We will only show the results when (n=100, s=3), (s=6,n=200), (s=12,n=400) but in supplementary all n=100,200,300,400 are shown. You can find the case when K = 500 in the supplementary but it shows the similar result as K = 500. We ran the simulation 1000 times.

Figure 2.1 shows the prediction accuracy. Each figure was drawn using a different number of nonzero variables(s). Within each figure, we varied the correlation

![Graphs showing prediction accuracy](image)
coefficients between the predictive markers for the same treatment. The elastic-net with stability selection works the best among the four methods in terms of prediction accuracy. In these scenarios, it is interesting that even though stability selection selects a fewer number of variables (See Figure 2.3), prediction accuracy using stability selection is higher than CV.

Figure 2.2 shows the probability of selecting at least \( r \) number of important variables. The \( r \) was varied from 1 to \( s \). This shows how many times at least \( r \) number of significant variables are selected among 1000 runs. For example, when \( s = 3 \) and \( \rho = 0.8 \), using the elastic net with stability selection, the probability of selecting at least 2 important variables is around 0.9. It means that the method selected at least 2 important variables about 900 times in 1000 runs. Elastic-net with stability selection shows highest selection probability when \( s = 3 \) and 6. The gap between elastic-net and the lasso becomes larger when the correlation gets higher since the elastic-net works best in correlated data. Using the stability selection method, when the number of significant markers increases, it is harder to detect all important variables since it is intended to select a small number of variables. However, it always selects the top most important variables (at least 6). Also, the elastic-net with stability selection always shows higher selection probability than the lasso with stability selection.

Figure 2.3 shows the average number of falsely chosen variables when selecting at least \( r \) important variables. Stability selection reduces the number of false positives compared to CV except for the case where the number of false positives using stability selection is already lower than 1 (See Figure 2.3a and Figure 2.3d). Using stability selection, the number of false positives remains almost the same when \( r \) increases whereas using CV, the number of false positives significantly increases when \( r \) increases: using stability selection, the number of false positives remains small (be-
Figure 2.2: Probability of selecting at least $r$ number of important variables: among 1000 runs, how many times at least $r$ number of variables are selected.
Figure 2.3: Average number of falsely chosen variables when selecting at least $r$ number of the important variables.
tween 0.5 to 1) most cases. However, using CV, it varies from near 0 (Figure 2.3a) to over 6 (Figure 2.3i). Stability selection reduces greater number of false positives when $s$ is larger and $\rho$ is smaller. The reduction of the number of false positives is maximized in Figure 2.3i: the number of falsely chosen variables is 6.2 using the elastic-net with CV, while it is 2 using the elastic-net with stability selection when selecting 12 important variables.

All in all, when variables are highly correlated, the elastic net outperform the lasso. However, the model selected by CV can have high false positive rate. Stability selection with the elastic net controls false discoveries while having good prediction accuracy and including top most important variables.

2.4.2 The BATTLE data analysis

We analyzed the BATTLE trial data. We used clinical data as well as microarray data. We pre-screened 98 probe sets. Since 4 treatment were used in the trial, with the probe sets and treatment by interaction effect with the probe sets, $p=395$. And $n=101$ among which $n=75$ are used for training set and $n=26$ are used for test set. See details in section 3.3.

The result is shown in Table2.4. Even though we selected a more generous set of cutoffs ($\pi_{th}r = 0.6$ and $v = 5$) than the default cutoffs and the selected cutoffs work fine in simulation data set in the previous section, it is too conservative in this battle data set, selecting less than 2 variables on average using either the lasso or elastic net. CV outperforms SS in terms of prediction accuracy. Plus, in this data set, SS with the elastic net works as bad as SS with the lasso.

The performance of SS heavily depends on the combination of right cutoffs and the data set. We have investigate on selecting an absolute combination of cutoffs
Table 2.4: The battle data analysis using CV and Stability selection with $\pi_{thr} = 0.6$ and $v = 5$

<table>
<thead>
<tr>
<th>Methods</th>
<th>AUC</th>
<th># of variable selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>CV.lasso</td>
<td>0.84(0.01)</td>
<td>23.96(0.57)</td>
</tr>
<tr>
<td>SS.lasso</td>
<td>0.61(0.01)</td>
<td>1.89(0.12)</td>
</tr>
<tr>
<td>CV.elastic</td>
<td>0.88(0.01)</td>
<td>44.5(2.19)</td>
</tr>
<tr>
<td>SS.elastic</td>
<td>0.58(0.01)</td>
<td>1.49(0.12)</td>
</tr>
</tbody>
</table>

Mean and standard error of AUC and the number of selected variables. CV is cross-validation and SS is stability selection.

and we set $\pi_{thr} = 0.6$ and $v = 5$ as our default cutoffs to have good number of true positives while controlling false positives. It works fine in some simulated data set but it was still too conservative for the real data.
2.5 Simulation studies: Investigation of different cutoffs of SS and comparison with CV

The purpose of this simulation studies is to show that (1) Cross validation (CV) often selects too many false positives to have good prediction accuracy and (2) different cutoffs for Stability selection (SS) are required and these should depends on data to have enough true positives and good prediction accuracy. Therefore, deciding absolute cutoffs of SS based on the inequality (2.5) is not recommended. We compare 3 combination of cutoffs: (1) \( v = 1 \) and \( \pi_{thr} = 0.8 (q = \sqrt{0.8p}) \), the default cutoff in Meinshausen and Buhlmann [14] (2) \( v = 2.5 \) and \( \pi_{thr} = 0.6 (q = \sqrt{0.5p}) \), used in a part of data analysis in Meinshausen and Buhlmann [14] (3) \( v = 5 \) and \( \pi_{thr} = 0.6 (q = \sqrt{p}) \), more generous cutoffs. Three different simulation examples are shown for the comparison. We generated the simulation data set 100 times and applied CV, and SS with the 3 combinations of cutoffs with the lasso and the elastic net, respectively.

Within each example, our simulation data consist of a training set and a test set. For CV, the training data set is used two times. First, it is used for 5-fold validation to find the optimal regularization parameters (\( \lambda \) and \( \alpha \)) simultaneously for the elastic net and only \( \lambda \) for the lasso. The CV error is calculated by the AUC (Area under the ROC curve) since the outcome is binary. Second, to estimate \( \beta \), the training set is used to fit the elastic net/the lasso with the entire training set using \( \lambda \) and \( \alpha \) as chosen by CV.

For SS, the training data set is used in three parts. First, to choose \( \alpha \), 5-fold CV is used (this step is skipped for the lasso). CV selects \( \alpha \) and \( \lambda \) simultaneously, but only \( \alpha \) is used for SS. Second, to select variables, we perform SS using \( B \) sub-samplings with the selected \( \alpha \) and select variables using preselected cutoffs. Third, to estimate
We fit a logistic regression model using the variables selected by SS.

We use the test data set to estimate the prediction accuracy. Prediction accuracy is measured by the AUC. The AUC can have values between 0 and 1: AUC = 1 shows perfect prediction accuracy; and AUC = 0.5 is the same as random guessing. See details about how to calculate AUC in Fawcett [26] and LeDell et al. [?]

Here are the details of the three scenarios.

(a) In example 1, we simulated 250 markers. The markers are generated from multivariate normal distribution. Each marker has mean 0 and variance 1. There are 4 treatments, the first of which is the reference treatment. Treatments are randomly assigned to patients with equal probability. There are 3 predictive markers for treatment 2. The predictive markers are correlated with each other with correlation coefficient, 0.8. The correlations outside of this group are 0.2. The number of variables \( p \) is 1003 including interaction terms. We generated 100 samples, among which 75 are used for training set and 25 are used for test set. The logistic model is

\[
\text{logit}(p) = (M_1 + M_2 + M_3)T_2
\]

(b) Example 2 is the same as example 1 except for the following. The number of markers is 500. There are 4 predictive markers for each of the 3 treatment excluding the reference treatment. The predictive markers for the same treatment are correlated with each other with correlation coefficient, 0.3. The correlations outside of these groups are 0.2. The number of variables is 2003 including interaction terms. We generated 300 samples, among which 150 are used for training set and 150 are used...
for test set. The true regression equation is

\[
\logit(p) = (M_1 + M_2 + M_3 + M_4)T_2 - (M_5 + M_6 + M_7 + M_8)T_3 \\
\quad + (M_9 + M_{10} + M_{11} + M_{12})T_4.
\]

(c) In example 3, we simulated the BATTLE trial data. There are 101 patients, among which 75 are used for training set and 26 are used for test set. The number of markers used in this simulation is 98 so that the number of variables is 395. The details of the data is found in PROMISE example section. Our simulation model is

\[
\logit(p) = (5.51M_1 + 4.51M_2 - 0.78M_3 + 4.57M_4 + 4.51M_5)T_2 \\
\quad + (-5.83M_6 + 1.92M_7)T_3 \\
\quad + (-3.72M_8 - 2.63M_9 - 3.15M_{10} - 2.27M_{11} - 2.05M_{12} + 3.44M_{13})T_4.
\]

The result using the lasso is shown in Table 2.5. In example 1, all methods similarly work well except for the default cutoff (SS.v=1). They select enough number of true positive variables so that prediction accuracy is good. In example 2, the default cutoff of SS is so conservative that it selects less than 1 variable on average. On the other hand, CV selects so generous that it selects many false positives. SS.v=5 works pretty well, showing high prediction accuracy and small number of false positives. In example 3, all 3 combinations of cutoffs of SS fail to select enough important variables to have good prediction accuracy. CV works well to have good prediction accuracy but it selects so many variables that the number of false positive is larger than the number of true positives.

The result using the lasso is shown in Table 2.6. It is similar to the result of the
Table 2.5: Result of simulations: comparison of stability selection (SS) with 3 combinations of the cutoffs and cross validation (CV) using the lasso

<table>
<thead>
<tr>
<th>Method</th>
<th>AUC</th>
<th># of variables</th>
<th>TP</th>
<th>FP</th>
<th>TPR</th>
<th>PPV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SS.v=1</td>
<td>0.65(0.02)</td>
<td>0.4(0.05)</td>
<td>0.4(0.05)</td>
<td>0(0)</td>
<td>0.13(0.02)</td>
<td>0.39(0.05)</td>
</tr>
<tr>
<td>SS.v=2.5</td>
<td>0.88(0.01)</td>
<td>1.85(0.08)</td>
<td>1.72(0.07)</td>
<td>0.13(0.03)</td>
<td>0.57(0.02)</td>
<td>0.94(0.02)</td>
</tr>
<tr>
<td>SS.v=5</td>
<td>0.88(0)</td>
<td>1.78(0.02)</td>
<td>1.64(0.02)</td>
<td>0.14(0.01)</td>
<td>0.55(0.01)</td>
<td>0.93(0.01)</td>
</tr>
<tr>
<td>cv</td>
<td>0.88(0)</td>
<td>1.84(0.08)</td>
<td>1.43(0.02)</td>
<td>0.41(0.07)</td>
<td>0.48(0.01)</td>
<td>0.94(0.01)</td>
</tr>
<tr>
<td>Example 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SS.v=1</td>
<td>0.59(0.01)</td>
<td>0.61(0.07)</td>
<td>0.6(0.07)</td>
<td>0.01(0.01)</td>
<td>0.05(0.01)</td>
<td>0.5(0.05)</td>
</tr>
<tr>
<td>SS.v=2.5</td>
<td>0.77(0.01)</td>
<td>3.65(0.14)</td>
<td>3.44(0.14)</td>
<td>0.21(0.04)</td>
<td>0.29(0.01)</td>
<td>0.94(0.01)</td>
</tr>
<tr>
<td>SS.v=5</td>
<td>0.79(0.01)</td>
<td>4.27(0.15)</td>
<td>3.94(0.14)</td>
<td>0.33(0.06)</td>
<td>0.33(0.01)</td>
<td>0.93(0.01)</td>
</tr>
<tr>
<td>cv</td>
<td>0.81(0.01)</td>
<td>11.82(1.1)</td>
<td>6.57(0.24)</td>
<td>5.25(0.97)</td>
<td>0.55(0.02)</td>
<td>0.73(0.02)</td>
</tr>
<tr>
<td>Example 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SS.v=1</td>
<td>0.5(0)</td>
<td>0.07(0.03)</td>
<td>0.07(0.03)</td>
<td>0(0)</td>
<td>0.01(0)</td>
<td>0.07(0.03)</td>
</tr>
<tr>
<td>SS.v=2.5</td>
<td>0.6(0.01)</td>
<td>1.34(0.1)</td>
<td>1.11(0.09)</td>
<td>0.23(0.05)</td>
<td>0.09(0.01)</td>
<td>0.65(0.04)</td>
</tr>
<tr>
<td>SS.v=5</td>
<td>0.69(0.01)</td>
<td>2.78(0.13)</td>
<td>2.31(0.11)</td>
<td>0.47(0.07)</td>
<td>0.18(0.01)</td>
<td>0.84(0.02)</td>
</tr>
<tr>
<td>CV</td>
<td>0.9(0.01)</td>
<td>23.21(0.65)</td>
<td>9.67(0.18)</td>
<td>13.54(0.55)</td>
<td>0.74(0.01)</td>
<td>0.43(0.01)</td>
</tr>
</tbody>
</table>

The numbers are the mean and standard error of AUC, # of selected variables, true positives (TP), false positives (FP), true positive rate (TPR), and positive predictive value (PPV).

lasso. However, using the elastic net, CV selects more variables than using the lasso, showing more false positives: in example 2, it selects about 30% more false positives than true positives and in example 3, it selects more than twice more false positive than true positives.

In this section, we showed that (1) CV predicts outcome well but tends to select so many variables, yielding too many false positives. (2) the fixed cutoffs of SS based on the inequality (2.5) in Meinshausen and Buhlmann [14] may or may not work properly in terms of prediction accuracy and variable selection accuracy depending on the data set. One reason is that this inequality does not reflect the characteristics of the individual data set except for the number of variables, $p$. In the example 1, $v = 2.5$ or $v = 5$ work good in terms of prediction accuracy and variable selection accuracy. In example 2, only $v = 5$ works well in terms prediction accuracy, even
Table 2.6: Result of simulations: Comparison of stability selection (SS) with 3 combinations of the cutoffs and cross validation (CV) using the elastic net

<table>
<thead>
<tr>
<th>Method</th>
<th>AUC</th>
<th># of variables</th>
<th>TP</th>
<th>FP</th>
<th>TPR</th>
<th>PPV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SS, v=1</td>
<td>0.86(0.01)</td>
<td>2.24(0.11)</td>
<td>2.18(0.1)</td>
<td>0.06(0.04)</td>
<td>0.73(0.03)</td>
<td>0.91(0.03)</td>
</tr>
<tr>
<td>SS, v=2.5</td>
<td>0.89(0.01)</td>
<td>2.87(0.07)</td>
<td>2.62(0.06)</td>
<td>0.25(0.05)</td>
<td>0.87(0.02)</td>
<td>0.93(0.01)</td>
</tr>
<tr>
<td>SS, v=5</td>
<td>0.88(0)</td>
<td>3.36(0.03)</td>
<td>2.75(0.02)</td>
<td>0.61(0.03)</td>
<td>0.92(0.01)</td>
<td>0.85(0.01)</td>
</tr>
<tr>
<td>cv</td>
<td>0.88(0)</td>
<td>1.8(0.05)</td>
<td>1.66(0.02)</td>
<td>0.13(0.03)</td>
<td>0.55(0.01)</td>
<td>0.97(0)</td>
</tr>
<tr>
<td>Example 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SS, v=1</td>
<td>0.65(0.01)</td>
<td>1.21(0.1)</td>
<td>1.2(0.1)</td>
<td>0.01(0.01)</td>
<td>0.1(0.01)</td>
<td>0.72(0.04)</td>
</tr>
<tr>
<td>SS, v=2.5</td>
<td>0.79(0.01)</td>
<td>4.1(0.15)</td>
<td>3.88(0.14)</td>
<td>0.22(0.05)</td>
<td>0.32(0.01)</td>
<td>0.95(0.01)</td>
</tr>
<tr>
<td>SS, v=5</td>
<td>0.81(0.01)</td>
<td>5.33(0.18)</td>
<td>4.87(0.17)</td>
<td>0.46(0.08)</td>
<td>0.41(0.01)</td>
<td>0.92(0.01)</td>
</tr>
<tr>
<td>cv</td>
<td>0.79(0.01)</td>
<td>15.86(2.66)</td>
<td>6.68(0.3)</td>
<td>9.18(2.53)</td>
<td>0.56(0.02)</td>
<td>0.61(0.03)</td>
</tr>
<tr>
<td>Example 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SS, v=1</td>
<td>0.5(0)</td>
<td>0.1(0.03)</td>
<td>0.1(0.03)</td>
<td>0(0)</td>
<td>0.01(0)</td>
<td>0.1(0.03)</td>
</tr>
<tr>
<td>SS, v=2.5</td>
<td>0.56(0.01)</td>
<td>1.03(0.09)</td>
<td>0.8(0.09)</td>
<td>0.23(0.05)</td>
<td>0.06(0.01)</td>
<td>0.5(0.05)</td>
</tr>
<tr>
<td>SS, v=5</td>
<td>0.66(0.01)</td>
<td>2.48(0.13)</td>
<td>2(0.13)</td>
<td>0.48(0.06)</td>
<td>0.15(0.01)</td>
<td>0.74(0.03)</td>
</tr>
<tr>
<td>cv</td>
<td>0.9(0.01)</td>
<td>32.66(1.55)</td>
<td>10.5(0.25)</td>
<td>22.16(1.41)</td>
<td>0.81(0.02)</td>
<td>0.35(0.01)</td>
</tr>
</tbody>
</table>

The numbers are the mean and standard error of AUC, # of selected variables, true positives (TP), false positives (FP), true positive rate (TPR), and positive predictive value (PPV).

though it misses some important variables. In example 3, all cutoffs fail to select important variables and therefore fails to have good prediction accuracy. Therefore, we need to find a method, which selects enough true positives to have good prediction accuracy but not select too many false positives. Also, the method should be applied to any type of data.
Chapter 3

Prediction Oriented Marker Selection (PROMISE)

3.1 Joint optimization based on CV and SS

As we have discussed, the sub-sampling method in SS plays an important role in screening false positives. However, the selection of two cutoffs ($\pi_{\text{thr}}$ and $\lambda_{\text{min}}$) based on inequality (2.5) often yields too few variables and too few true positives [15, 16]. Also, it is impossible to find a combination of cutoffs that work well for all of the data in order to have good prediction accuracy and variable selection accuracy. To resolve these issues, we propose PROMISE, which flexibly selects the cutoffs ($\pi_{\text{thr}}$, $\lambda_{\text{min}}$) of SS based on the prediction accuracy of the individual data sets. We apply CV to select the cutoffs of SS. Since the goal of CV is to maximize prediction accuracy, important variables are automatically included in the model selected by CV. Therefore, PROMISE, with the application of CV to SS, also ensures that important variables are included in the model in order to have good prediction accuracy. On the other hand, since the sub-sampling method in SS greatly reduces the false discoveries and produces a sparse solution, PROMISE also reduces the false discoveries and produces a sparse solution compared to CV. Therefore, PROMISE, the combination of SS and CV, is developed to reduce the false positives and the number of selected variables of CV, and to increase prediction accuracy and the number of true positives of SS.

PROMISE uses the first K-1 part of the data to subsample, select variables using different combinations of $\pi_{\text{thr}}$ and $\lambda_{\text{min}}$, and fit a logistic regression model using the
selected set of variables. Then, it uses remaining K-th part of the data to estimate prediction accuracy [21]. Finally, it selects the combination of \( \pi_{\text{thr}} \) and \( \lambda_{\text{min}} \), which shows the best prediction accuracy using the 1SE rule.

To illustrate, we apply PROMISE to the lasso in the following way. Conventional CV is used to select the regularization parameter, \( \lambda \), as described in Section 2.3.1. PROMISE uses CV to determine \( \lambda_{\text{min}} \) and \( \pi_{\text{thr}} \), the cutoffs. Using K-1-fold of data, instead of fitting the lasso just one time each for a set of regularization parameters as in conventional CV, we fit the lasso \( B \) times using \( B \) sub-samplings for a set of regularization parameters. Then, we calculate the selection probability for each variable \( j \in 1, \ldots, p \) for each \( \lambda \), excluding the \( k \)th part of the data,

\[
\hat{\Pi}^{(-k)\lambda}_j = \frac{1}{B} \sum_{m=1}^{B} 1\{j \in \hat{S}^\lambda(I_m^{(-k)})\},
\]

where \( B \) is the number of sub-samplings and \( \hat{S}^\lambda(I_m^{(-k)}) \) is the selected variables using \( \lambda \) when it is applied to a random \( m \)th sub-sample \( I_m^{(-k)} \), excluding the \( k \)th part of the data with size \( \lfloor (n - n_k)/2 \rfloor \), where \( n_k \) is the number of observations in \( k \)-fold of the data, and \( 1 \) is an indicator function.

Then, instead of selecting the variables for each \( \lambda \), as in conventional CV, we select the variables for each combination of \((\lambda_{\text{min}}, \pi_{\text{thr}})\).

For a candidate set of the cutoffs, \((\lambda_r, \pi_r) = ((\lambda_1, \pi_1), \ldots, (\lambda_R, \pi_R))\), the set of selected variables using \((\lambda_r, \pi_r)\) without the \( k \)th part of the data is

\[
SS^{(-k)}(\lambda_r, \pi_r) = \left\{ j : \max_{\lambda_r \leq \lambda \leq \lambda_{\text{max}}} (\hat{\Pi}^{(-k)\lambda}_j) \geq \pi_r \right\}.
\]

Then, we estimate the regression coefficients to predict outcome, which is missing
in SS. The unselected variables have 0 coefficients,

\[ \hat{\beta}_j^{\text{SS}(-k)}(\lambda_r, \pi_r) = 0 \text{ for } j \notin \text{SS}^{(-k)}(\lambda_r, \pi_r). \]

For the set of selected variables, we fit a logistic regression model or a ridge logistic regression with a small penalty when the logistic regression model does not converge due to high dimensionality. The estimated coefficients using \((\lambda_r, \pi_r)\), computed with the data except for the k part, are

\[
\hat{\beta}^{\text{SS}(-k)}(\lambda_r, \pi_r) = \arg \max_\beta \left[ \sum_{i \notin k} y_i (\beta^T x_i^{\text{SS}}(\lambda_r, \pi_r)) - \log(1 + \exp(\beta^T x_i^{\text{SS}}(\lambda_r, \pi_r))) - \lambda \| \beta \|_2^2 \right],
\]

where \( \beta \) is a vector of \( \beta_j \) for \( j \in \text{SS}^{(-k)}(\lambda_r, \pi_r) \) and \( x_i^{\text{SS}}(\lambda_r, \pi_r) \) is a vector \( x_{ij} \) for \( j \in \text{SS}^{(-k)}(\lambda_r, \pi_r) \), including the constant term 1. For the logistic regression, \( \lambda = 0 \), and for the ridge logistic regression, \( \lambda = 0.01 \).

The remaining procedures of PROMISE are similar to those of conventional CV: We predict the probability of response in the kth part of the data using the predicted coefficients for each combination of the cutoffs. For \( i \in k \),

\[
\hat{p}_i(\lambda_r, \pi_r) = \frac{\exp((\hat{\beta}^{\text{SS}(-k)}(\lambda_r, \pi_r))^T x_i^{(k)})}{1 + \exp((\hat{\beta}^{\text{SS}(-k)}(\lambda_r, \pi_r))^T x_i^{(k)})}.
\]

Then, we calculate the prediction accuracy, the AUC, using the response in k-fold of the data. We perform this procedure for \( k=1, \ldots, K \) and then calculate the average prediction accuracy for each combination of the cutoff.

The K-fold cross-validated AUC estimator to select \( \lambda_{\text{min}} \) and \( \pi_{\text{thr}} \) is
\[ CV_{\text{AUC}} = \frac{1}{K} \sum_{k=1}^{K} \text{AUC}(y^{(k)}, \hat{p}^{(k)}(\lambda_r, \pi_r)), \]

where \( y^{(k)} \) is a vector of \( y_i \) for \( i \in k \) and \( \hat{p}^{(k)} \) is a vector of \( \hat{p}_i \) for \( i \in k \).

We choose values of \( \lambda_r \) and \( \pi_r \) that provide the most parsimonious model, for which \( CV_{\text{AUC}} \) is not more than one standard error (1SE) difference from the model that maximizes \( CV_{\text{AUC}} \) (1SE rule).

After choosing the cutoffs, similar to the conventional way, we use the entire data set to finalize the model. However, instead of fitting the lasso using the selected regularized parameter just one time, we fit the lasso \( B \) times using \( B \) sub-samplings for the selected set of regularization parameters \((\lambda_{\text{min}}, \lambda_{\text{max}})\) (only \( \lambda_{\text{min}} \) was selected by CV). Then, we calculate the selection probabilities. Instead of determining the final set of variables using the regularization parameter, selected by conventional CV, we determine the final set of variables using the values of \( \pi_{\text{thr}} \), selected by the CV procedure, by selecting the variables for which the selection probabilities are higher than \( \pi_{\text{thr}} \). To obtain the final model, we fit a logistic regression with the final set of variables. The PROMISE algorithm is succinctly summarized in Figure 3.1.

### 3.2 Application for selecting more than one regularization parameter

When we have more than one parameter to be selected, we want to determine the parameter using CV along with the cutoffs. For example, the PROMISE procedure for the elastic net is as follows. We set a candidate \( \alpha, (\alpha_1, \ldots, \alpha_R) \). For each \( \alpha \), we perform the CV procedures the same as the lasso, except that we fit the elastic net with \( \alpha \) instead of the lasso in the sub-sampling procedure. Then, we have CV errors
for each combination of $\alpha$, $\pi_{thr}$, and $\lambda_{min}$. Then, we choose the combination based on the 1SE rule. To decide the set of final variables, using the $\alpha$ selected by CV, we perform sub-samplings and select the variables using the cutoffs selected by CV. We fit a regression model for the final step.

**Other Details**

- To improve the sub-sampling procedure, we use a stratified random sampling method, which selects the same portion of response 1 or 0 as the entire data set.

- Pre-selection of $\lambda$ in the sub-sampling procedure is done by using the glmnet [22]; the maximum $\lambda$ is the smallest $\lambda$ that chooses no variables, and the minimum $\lambda$ is the maximum $\lambda \times 0.001$. The range of $\lambda$ is a sequence from maximum $\lambda$ to minimum $\lambda$, with 20 values on a log scale. This maximum $\lambda$ is also used for the final sub-sampling procedure to determine the final set of variables after CV.

- $B = 100$ and $\pi_r = (0.3, ..., 0.8)$.

- $\alpha_r = (0.1, ..., 1)$ for the elastic net.

### 3.3 Illustrative examples

We want to illustrate how PROMISE works using data from the BATTLE trial and generating treatment responses from the data.

**BATTLE trial specifics** The BATTLE trial, a phase II trial, involved 255 patients with advanced non-small cell lung cancer and the evaluation of 4 treatments. In the trial, the primary endpoint was the 8-week disease control rate, which was defined as
the percentage of patients who achieved complete response, partial response or stable disease following a therapeutic intervention in the clinical trial [27]. Note that the disease control rate is a binary outcome. The trial was an adaptive randomized trial, which assigned patients to a treatment with a higher disease control rate based on their biomarker-guided profiles [8]. In the trial, 11 candidate predictive markers were pre-selected on the basis of biological background information such as mutations, copy numbers, and protein expression. However, several of the markers could not function as predictive markers: they could not predict treatment outcome when they were used with the corresponding treatment [3]. In this study, we want to use the microarray gene expression data set, which was not used for treatment assignment in the trial, to identify predictive markers so that patients can be assigned to the best treatment according to their marker status.

**Pre-screening variables and generating the treatment response** We benchmark our method with the BATTLE data. The data matrix $X$ is a subset of the BATTLE microarray data set. We set the regression coefficients to generate a “pseudo treatment outcome” based on the underlying true models in different scenarios. The real data analysis is described in Section 3.5.

The microarray data (platform: Affymetrix HG1.0ST) were collected from 101 patients among 255 evaluable patients in the BATTLE trial. In this paper, we use probe-set-level data as candidate markers in our model, which consist of 33297 probe sets/genes. For predictive marker detection, we included the gene-by-treatment interaction effects in addition to the marginal effects as variables. Therefore, the number of variables is $33297 \times 4 + 3 = 133191$. Since there were 4 treatments in the trial, we used 3 treatment indicator variables, defined in Table 3.1.
<table>
<thead>
<tr>
<th>Treatments</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Erlotinib</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>Vandetanib</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>Erlotinib+bexarotene</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>Sorafenib</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3.1: Treatment coding

Since this dimension was too high for either the lasso or the elastic net to work properly, we pre-screened the variables, which consisted of probe sets and their interaction effects with treatments. Prior to that step, we standardized each variable with mean 0 and variance 1. Using a univariate t-test with threshold p-value of 0.0013, we selected 98 genes for which either the marginal effects or the interaction effects with treatments are significant. Therefore, the number of pre-screened variables, including the interaction effects and treatment indicator variables, is $98 \times 4 + 3 = 395$. This forms our final data matrix $X$ in this section.

To mimic the treatment response of the real data, we set $\beta$ as the estimated coefficients from the real data analysis. We set the 13 most frequently selected probe sets as nonzero variables, and ordered the data matrix to make the first 13 markers significant. The coefficients are from mean value of estimated coefficients using the elastic net with PROMISE among the runs with high prediction accuracy (AUC $> 0.9$). Our simulation model is

$$
\text{logit}(p) = (5.51M_1 + 4.51M_2 - 0.78M_3 + 4.57M_4 + 4.51M_5)T_2 \\
+ (-5.83M_6 + 1.92M_7)T_3 \\
+ (-3.72M_8 - 2.63M_9 - 3.15M_{10} - 2.27M_{11} - 2.05M_{12} + 3.44M_{13})T_4.
$$

Then, we simulated the treatment response from the Bernoulli distribution in the
Results. The PROMISE algorithm simultaneously selects both $\alpha$ and $\lambda$ in the elastic net. The value of $\alpha$ is fixed at 0.4, which is selected by PROMISE, to show the variable selection paths (Figure 3.2). In the regularization path, (Figure 3.2a), some of the important variables stand out, but many are mixed with irrelevant variables and it is hard to identify and select important variables without selecting many noise variables. As a result, CV selected twice as many noise variables as important variables. Important variables are better identified in the stability path (Figure 3.2b) than in the regularization path because few variables are mixed with noise variables. However, using the default cutoffs, $\pi_{\text{thr}} = 0.9$ and $q = 17.78$, no variable was selected by SS. The stability path using PROMISE (Figure 3.2c) is similar to the stability path using SS. However, a difference arises from the cutoffs, which are automatically selected by PROMISE, and which identify most of the important variables while selecting far fewer noise variables than CV. CV selected almost three times more noise variables than PROMISE. These paths are drawn using the entire data set of observations. The analysis in Section 3.4 uses separate training and test data sets to evaluate prediction accuracy.
Figure 3.1: Algorithm of PROMISE

1. Divide a data set into K parts

2. For each part, k=(1,...,K),
   - Using the samples except those in part k,
     (a) Randomly select $n_{sub}$ sub-samplings and fit a variable selection method $B$ times to obtain the selection probability, $\hat{\Pi}_i^\lambda$ for $i \in \{1,...,p\}$ and pre-selected $\lambda$s
     (b) Set candidate $\pi_{thr}$, $(\pi_1,...,\pi_R)$, and candidate $\lambda_{min}$, $(\lambda_1,...,\lambda_R)$, all the pre-selected $\lambda$s except for the smallest $\lambda$
     (c) Using each combination of $(\pi_{thr}, \lambda_{min})$, select the variables
       i. Fit a logistic regression model to estimate the regression coefficients
       ii. Using the samples in part k, predict the outcomes and calculate the prediction accuracy

3. For each combination of $(\pi_{thr}, \lambda_{min})$, calculate the CV errors

4. Select the cutoffs $(\lambda_{min}, \pi_{thr})$ based on the 1SE rule

5. Obtain the final model
   - (a) Perform $B$ sub-samplings using the entire training data set
   - (b) Using the selected $\pi_{thr}$ and $\lambda_{min}$ from equation 4, select the final set of variables
   - (c) Fit a logistic regression to obtain the final model
Solid lines are the paths of important variables; dotted lines are the paths of noise variables. (a) A regularization path: x-axis is log $\lambda$ and the corresponding number of selected variables is shown along the top of the plot. The y-axis shows $\hat{\beta}^\lambda$. The vertical line is at the log $\lambda$ that is selected by CV. It selects 39 variables, including all 13 important variables and 26 noise variables. (b) A stability path using the default cutoffs in SS: The x-axis is log $\lambda$ and the average number of selected variables ($q^\lambda$) using 100 sub-samplings is shown along the top of the plot. The y-axis shows selection probability ($\hat{\pi}^k$). The horizontal and vertical lines are the default cutoffs: $\pi_{thr} = 0.9$ and $q = 17.78$ (log $\lambda_{min} = -0.94$), respectively. The selected variables are those with selection probabilities higher than $\pi_{thr}$ in the range of log $\lambda \geq \log \lambda_{min}$. These are in the upper right corner formed by the intersection of the two cutoffs. No variable is selected. (c) A stability path using PROMISE: the cutoffs are selected by CV: $\pi_{thr} = 0.6$ and log $\lambda_{min} = -3.28$. PROMISE selects 21 variables, among which 12 are important and 9 are noise; it selects far fewer false positives and a similar number of true positives compared to CV.
3.4 Simulation studies based on real data

3.4.1 Thirteen signals

. We analyzed the simulation data described in Section 3.3. In Section 3.3, we selected variables using the entire data set. However, in this section, we not only select variables but also predict treatment outcome. To measure prediction accuracy, we randomly select three quarters of the samples as the training data set, while preserving the proportion of response 1, and use the remaining one quarter of the samples as the test data.

For CV, the training data set is used two times. First, 5-fold cross-validation is used to find the optimal regularization parameters ($\lambda$ and $\alpha$) simultaneously for the elastic net and only $\lambda$ for the lasso. Since three quarters of the data are the training set, 3/20 of the samples are in each fold. The CV error is calculated by the AUC since the outcome is binary. Second, to estimate $\beta$, we fit the elastic net/the lasso with the entire training set using $\lambda$ and $\alpha$ as chosen by CV.

For SS, the training data set is used in three parts. First, to choose $\alpha$, 5-fold CV is used (this step is skipped for the lasso). We select $\alpha$ and $\lambda$ simultaneously, but use only $\alpha$ for SS. Second, to select variables, we perform SS using $B$ sub-samplings with the selected $\alpha$. Third, to estimate $\beta$, we fit a logistic regression model using the variables selected by SS.

For PROMISE, the training data set is used in three parts. First, 5-fold CV is used to choose $\alpha$, $\lambda_{\text{min}}$, and $\pi_{\text{thr}}$ for the elastic net and $\lambda_{\text{min}}$, and $\pi_{\text{thr}}$ for the lasso. Second, to select the variables, we perform SS using $B$ sub-samplings with the selected $\alpha$ ($\alpha = 1$ for the lasso). Third, to estimate $\beta$, we fit a logistic regression model using the variables selected by PROMISE.
We use the test data set to estimate the prediction accuracy. Prediction accuracy is measured by the AUC. The AUC can have values between 0 and 1: AUC = 1 shows perfect prediction accuracy; and AUC = 0.5 is the same as random guessing. See details about how to calculate AUC in Fawcett [26] and LeDell et al. [?]

We generated the simulation data set 100 times and applied CV, SS, and PROMISE with the lasso and the elastic net, respectively.

Table 3.2: Comparison of PROMISE, CV, and SS using the BATTLE simulation data for (a) the lasso and (b) the elastic net

(a) Lasso

<table>
<thead>
<tr>
<th>Method</th>
<th>PROMISE</th>
<th>CV</th>
<th>SS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUC</td>
<td>0.87(0.01)</td>
<td>0.9(0.01)</td>
<td>0.5(0)</td>
</tr>
<tr>
<td># Sel</td>
<td>12.98(0.48)</td>
<td>23.21(0.65)</td>
<td>0.01(0.01)</td>
</tr>
<tr>
<td># TP</td>
<td>7.62(0.23)</td>
<td>9.67(0.18)</td>
<td>0.01(0.01)</td>
</tr>
<tr>
<td># FP</td>
<td>5.36(0.32)</td>
<td>13.54(0.55)</td>
<td>0(0)</td>
</tr>
<tr>
<td># FN</td>
<td>3.42(0.19)</td>
<td>2.5(0.25)</td>
<td>13(0)</td>
</tr>
<tr>
<td># Errors</td>
<td>10.74(0.28)</td>
<td>16.87(0.51)</td>
<td>12.99(0.01)</td>
</tr>
<tr>
<td>TPR</td>
<td>0.59(0.02)</td>
<td>0.74(0.01)</td>
<td>0(0)</td>
</tr>
<tr>
<td>PPV</td>
<td>0.62(0.01)</td>
<td>0.43(0.01)</td>
<td>0.01(0.01)</td>
</tr>
</tbody>
</table>

(b) Elastic net

<table>
<thead>
<tr>
<th>Method</th>
<th>PROMISE</th>
<th>CV</th>
<th>SS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUC</td>
<td>0.9(0.01)</td>
<td>0.9(0.01)</td>
<td>0.5(0)</td>
</tr>
<tr>
<td># Sel</td>
<td>23.69(1.32)</td>
<td>32.66(1.55)</td>
<td>0(0)</td>
</tr>
<tr>
<td># TP</td>
<td>9.58(0.19)</td>
<td>10.5(0.25)</td>
<td>0(0)</td>
</tr>
<tr>
<td># FP</td>
<td>14.11(1.22)</td>
<td>22.16(1.41)</td>
<td>0(0)</td>
</tr>
<tr>
<td># FN</td>
<td>5.38(0.23)</td>
<td>3.33(0.18)</td>
<td>12.99(0.01)</td>
</tr>
<tr>
<td># Errors</td>
<td>17.53(1.14)</td>
<td>24.66(1.29)</td>
<td>13(0)</td>
</tr>
<tr>
<td>TPR</td>
<td>0.74(0.01)</td>
<td>0.81(0.02)</td>
<td>0(0)</td>
</tr>
<tr>
<td>PPV</td>
<td>0.46(0.01)</td>
<td>0.35(0.01)</td>
<td>0(0)</td>
</tr>
</tbody>
</table>

The numbers are the mean and standard error of AUC, # of selected variables (# Sel), # of true positives (# TP), # of false positives (# FP), # of false negatives (# FN), # of false positive + false negative errors (# Errors), true positive rate (TPR), and positive predictive value (PPV)
**Result**  
Table 3.2 summarizes the prediction and variable selection results using the 100 simulated data sets. First, the stability selection does not properly function as a variable selector using either the elastic net or the lasso; it selects almost no variables and results in a random prediction (AUC=0.5). Second, although the prediction accuracy of PROMISE (around 90%) is similar to that of CV, PROMISE produces a much more parsimonious solution than CV: PROMISE selects 80% (using the lasso) or 40% (using the elastic net) less variables than CV. Third, PROMISE selects variables more accurately than CV, yielding fewer type I errors (false positive) plus type II errors (false negative) than CV. This is because PROMISE selects far fewer noise variables than CV even though it misses fewer nonzero variables than CV. Using the lasso, PROMISE selects 8 fewer noise variables than CV but misses only two more nonzero variables than CV on average. Also, using the elastic net, PROMISE selects 7 fewer noise variables than CV but misses only one more important variable than CV on average. As a result, PROMISE shows higher positive predictive value (PPV) than CV even though PROMISE shows a lower true positive rate (TPR) than CV. In biomarker discovery, PPV is a more important measure than TPR because, in reality, the denominator of TPR, which is the number of true biomarkers, cannot be known. Also, investigators focus more on PPV, the proportion of successes among the total findings, because it directly relates to the decision making when assigning patients to treatments or performing additional tests to confirm the initial findings.

Overall, PROMISE achieves a more sparse solution and more accurate variable selection than CV while giving similar prediction accuracy. In contrast, the stability selection method does not function at all for this data set.
3.4.2 No signal

In this subsection, we use the same covariates as in the previous section, but we set all the coefficients to zero. The goal is to check the type I error rate. Table 3.3 shows the results. First, SS does not select any variable, which works well in this scenario. PROMISE selects less than one-fourth of the false positives selected by CV using the lasso, and PROMISE selects less than half of the false positives selected by CV using the elastic net. Note that PROMISE with the lasso selects only 1.6 false positives on average among 395 variables, which shows a very low type I error rate (0.004).

Table 3.3: Comparison of PROMISE, CV, and SS for (a) the lasso and (b) the elastic net when there is no signal

<table>
<thead>
<tr>
<th></th>
<th>PROMISE</th>
<th>CV</th>
<th>SS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUC</td>
<td>0.51(0.01)</td>
<td>0.51(0.01)</td>
<td>0.5(0)</td>
</tr>
<tr>
<td>FP</td>
<td>1.63(0.29)</td>
<td>8.87(1.38)</td>
<td>0(0)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>PROMISE</th>
<th>CV</th>
<th>SS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUC</td>
<td>0.5(0.01)</td>
<td>0.48(0.01)</td>
<td>0.5(0)</td>
</tr>
<tr>
<td>FP</td>
<td>10.63(2.81)</td>
<td>27.87(4.66)</td>
<td>0(0)</td>
</tr>
</tbody>
</table>

The numbers are the mean and standard error of AUC and the number of false positives (# FP)

3.5 BATTLE data analysis

Data set and pre-processing. For this application to real data, we use the BATTLE trial data as described before. This analysis is similar to the simulated data analysis we already described, except that the treatment response is obtained from a real clinical trial. Therefore, we want to identify the important biomarkers
and their biological meaning. Also, we predict the treatment outcome based on the selected markers.

We use the same subset of microarray data as in Section 3.3. We add the clinical covariates and candidate predictive markers that were used to assign patients to a treatment in the trial. There were 13 clinical covariates and 14 markers, and we pre-screen those variables. We find that 2 clinical covariates and 6 markers were statistically significant in predicting the disease control status at level $\alpha = 0.05$, either by Kim et al.\cite{kim2003} or by a z-test in a simple logistic regression model, which consists of a single variable and its interaction effects with the treatments. These pre-selected markers are EGFR mutation, EGFR amplification, Kras/Braf marker group, VEGFR-2 expression, RXR beta expression, and cyclin D1 expression. The pre-selected clinical covariate is histology, which consists of adenocarcinoma, squamous cell carcinoma, and others. Since it has 3 categories, 2 indicator variables are created for this covariate.

As a result, we pre-select 98 probe sets and 8 clinical + marker variables. Therefore, the total number of variables is $(98 + 8) \times 4 + 3 = 427$. We apply CV, SS, and PROMISE with the lasso and the elastic net to this data set to select significant markers and predict the 8-week disease control rate. We split the data into a training set and a test set as described in the simulation studies. We perform the random division 100 times to evaluate the performance of each method.

**Results**

Figure 3.3 summarizes the prediction and variable selection results. Since SS with the default cutoff does not select any variable (on average 0.04 variable selected), we remove it from the graph. The figure shows that although PROMISE predicts as well
as CV, PROMISE produces a much more parsimonious solution than CV: PROMISE selects almost half (12) the number of variables selected by CV (24) using the lasso and 60% less variables (28) than CV (45) using the elastic net.

Table 3.4 shows the gene symbols/probe set ids that are selected by PROMISE with the elastic net at least 20 times, which are sorted by the number of selections among 100 runs. The original microarray data included only probe set ids for the probe sets; however, when a gene can be matched with a probe set id, a gene symbol is listed in the table.

**Biological Interpretation** We checked whether there is any association between the type of cancer and individual genes that are selected with high probability by PROMISE using the elastic net. We used Ingenuity® Systems (www.ingenuity.com) to conduct the gene search. Even though that resource does not distinguish genes by whether they are predictive or prognostic markers, we explored the biological
implications of our findings.

The most frequently selected gene, TXNDC12 (thioredoxin domain containing 12), is closely related to cancer. It is one of two members of protein-disulfide reductase (glutathione). Since high glutathione levels protect tumor cells, a targeted therapy that lowers glutathione levels can protect normal cells while allowing tumor cells to be sensitive to chemotherapy. This anti-neoplastic therapeutic approach has been tested in adenocarcinoma and in ovarian and breast cancer [28]. Also, glutathione disulfide reductase, which catalyzes the increase of glutathione [29], is being tested in clinical trials as a predictive marker for leukemia and lymphoma. It has been approved as a predictive marker for multiple myeloma.

Table 3.5 shows that the 9 genes most frequently selected by PROMISE with the elastic net are related to cancer. In particular, 5 of the 9 genes are related to adenocarcinoma, which is the most common cancer type in non-small cell lung cancer.

In addition, we conducted Ingenuity pathway analysis using Ingenuity Systems (www.ingenuity.com) to gain biological insight into the functional roles of the selected genes in signaling pathways. Figure 3.4 shows the significant pathways to which the selected genes belong.

Many of these pathways are related to lung cancer. Non-small cell lung cancer signaling and small cell lung cancer signaling are directly related to lung cancer. Granulocyte-macrophage colony-stimulating factor (GM-CSF) signaling reduces tumor proliferation and invades lung cancer cells [32]. Activation of the vitamin D receptor (VDR) and retinoid X receptor (RXR) pathway mediates calcitriol, which inhibits tumor growth in lung cancer [33]. The aryl hydrocarbon receptor pathway has been suggested as a biomarker for lung cancer [34]. Down-regulation of Gadd45 expression is associated with tumor differentiation in non-small cell lung cancer [35].
Inhibition of Wnt reduces the proliferation of non-small cell lung cancer cell lines [36]. Many of the other significant pathways are related to cancer, such as thyroid cancer signaling, glioma signaling, and molecular mechanisms of cancer.

These findings indicate that the genes frequently selected by PROMISE with the elastic net have biological plausibility, especially in their association with lung cancer or adenocarcinoma. These findings suggest that the genes shown in Table 3.5 in particular are possible predictive markers for non-small cell lung cancer.
Table 3.4: List of gene/probe sets selected from the BATTLE trial analysis.

<table>
<thead>
<tr>
<th>Gene/Probe Symbol</th>
<th>Type</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>TXNDC12</td>
<td>Vandetanib</td>
<td>0.95</td>
</tr>
<tr>
<td>ZNF334</td>
<td>Sorafenib</td>
<td>0.92</td>
</tr>
<tr>
<td>SLTIRK6</td>
<td>Vandetanib</td>
<td>0.85</td>
</tr>
<tr>
<td>GAGE12B/GAGE88</td>
<td>Erlotinib+bexarotene/Marginal</td>
<td>0.82/0.23</td>
</tr>
<tr>
<td>7893000</td>
<td>Sorafenib</td>
<td>0.81</td>
</tr>
<tr>
<td>8003846</td>
<td>Vandetanib</td>
<td>0.79</td>
</tr>
<tr>
<td>AGPAT4</td>
<td>Sorafenib</td>
<td>0.76</td>
</tr>
<tr>
<td>7890225</td>
<td>Sorafenib/Vandetanib</td>
<td>0.71/0.39</td>
</tr>
<tr>
<td>8088915</td>
<td>Vandetanib</td>
<td>0.70</td>
</tr>
<tr>
<td>7894855</td>
<td>Erlotinib+bexarotene</td>
<td>0.70</td>
</tr>
<tr>
<td>RFT1</td>
<td>Sorafenib</td>
<td>0.70</td>
</tr>
<tr>
<td>SLC16A12</td>
<td>Sorafenib</td>
<td>0.66</td>
</tr>
<tr>
<td>7896557</td>
<td>Erlotinib+bexarotene/Vandetanib</td>
<td>0.65/0.28</td>
</tr>
<tr>
<td>SLC1A4</td>
<td>Vandetanib</td>
<td>0.61</td>
</tr>
<tr>
<td>UNC80</td>
<td>Vandetanib</td>
<td>0.61</td>
</tr>
<tr>
<td>ZNF674</td>
<td>Vandetanib/Sorafenib</td>
<td>0.60/0.37</td>
</tr>
<tr>
<td>8113784</td>
<td>Vandetanib</td>
<td>0.57</td>
</tr>
<tr>
<td>PRH1</td>
<td>Sorafenib</td>
<td>0.57</td>
</tr>
<tr>
<td>HCP5</td>
<td>Marginal</td>
<td>0.52</td>
</tr>
<tr>
<td>7895065</td>
<td>Sorafenib</td>
<td>0.52</td>
</tr>
<tr>
<td>Olfr1288</td>
<td>Marginal/Erlotinib+bexarotene/Vandetanib</td>
<td>0.49/0.32/0.30</td>
</tr>
<tr>
<td>7893401</td>
<td>Marginal/Vandetanib</td>
<td>0.49/0.42</td>
</tr>
<tr>
<td>RPL38</td>
<td>Vandetanib</td>
<td>0.46</td>
</tr>
<tr>
<td>NPY5R</td>
<td>Vandetanib</td>
<td>0.45</td>
</tr>
<tr>
<td>CRABP2</td>
<td>Erlotinib+bexarotene/Marginal</td>
<td>0.34/0.34</td>
</tr>
<tr>
<td>ACTR6</td>
<td>Vandetanib</td>
<td>0.29</td>
</tr>
<tr>
<td>GABRB1</td>
<td>Vandetanib</td>
<td>0.28</td>
</tr>
<tr>
<td>7892819</td>
<td>Vandetanib</td>
<td>0.24</td>
</tr>
<tr>
<td>RRP7A</td>
<td>Sorafenib</td>
<td>0.24</td>
</tr>
<tr>
<td>TMX3</td>
<td>Vandetanib</td>
<td>0.23</td>
</tr>
<tr>
<td>DYSPL5</td>
<td>Marginal</td>
<td>0.22</td>
</tr>
<tr>
<td>KDM3A</td>
<td>Vandetanib</td>
<td>0.21</td>
</tr>
<tr>
<td>SNX32</td>
<td>Vandetanib</td>
<td>0.21</td>
</tr>
<tr>
<td>RXRB</td>
<td>Vandetanib</td>
<td>0.21</td>
</tr>
<tr>
<td>CTRC</td>
<td>Vandetanib</td>
<td>0.20</td>
</tr>
</tbody>
</table>

Genes/probe sets that are selected more than 20 times among 100 runs by PROMISE with the elastic net. This list is sorted by the probability of selection among 100 runs. A gene symbol begins with a letter and a probe set id begins with a number. Type indicates whether each gene/probe set has a marginal effect or an interaction effect with a treatment indicator variable: if it has an interaction effect with a treatment indicator variable, the treatment name is shown. Selection probability indicates how many times each variable is selected among the 100 runs.
Table 3.5: Top 9 most frequently selected genes from the BATTLE data analysis and their association with cancer.

<table>
<thead>
<tr>
<th>Gene Symbol</th>
<th>Associated Cancer</th>
</tr>
</thead>
<tbody>
<tr>
<td>TXNDC12</td>
<td>adenocarcinoma, breast cancer, ovarian cancer [28]</td>
</tr>
<tr>
<td>ZNF334</td>
<td>adenocarcinoma, endometrioid carcinoma, melanoma</td>
</tr>
<tr>
<td>SLITRK6</td>
<td>adenocarcinoma, endometrioid carcinoma, melanoma</td>
</tr>
<tr>
<td>GAGE12B/GAGE8</td>
<td>bone marrow cancer [30]</td>
</tr>
<tr>
<td>AGPAT4</td>
<td>melanoma, endometrioid carcinoma</td>
</tr>
<tr>
<td>RFT1</td>
<td>melanoma</td>
</tr>
<tr>
<td>SLC16A12</td>
<td>melanoma, endometrioid carcinoma</td>
</tr>
<tr>
<td>SLC1A4</td>
<td>adenocarcinoma [31], bone cancer cell lines, colon cancer cell lines</td>
</tr>
<tr>
<td>UNC80</td>
<td>adenocarcinoma, melanoma, endometrioid carcinoma</td>
</tr>
</tbody>
</table>
Figure 3.4: Canonical pathways enriched for selected genes from the BATTLE data analysis

These pathways are significant (p-value < 0.05) using Fisher’s exact test, which identifies whether the selected genes are in each pathway by chance. The pathways are sorted by significance.
Chapter 4

PROMISE application to the group lasso

In this chapter, we apply PROMISE to the group lasso to select biomarkers using biological information such as gene pathways. We first select important pathways using the group lasso and then select individual genes within the pathway using the lasso.

4.1 Method

Logistic group lasso was proposed by Meir et al [37]. The coefficients are estimated by maximizing the penalized log likelihood,

$$
\max_{(\beta_0, \beta) \in \mathbb{R}^{(p+1)}} \left[ \sum_{i=1}^{n} \left\{ y_i \log p_i + (1 - y_i) \log(1 - p_i) \right\} - \lambda \sum_{g=1}^{G} ||\beta^{(g)}||_2 \right],
$$

where $\beta^{(g)}$ is the coefficient vector of group $g$ and

$$
p_i = \frac{\exp(\beta_0 + \sum_{j=1}^{p} x_{ij} \beta_j)}{1 + \exp(\beta_0 + \sum_{j=1}^{p} x_{ij} \beta_j)}
$$

is the probability of having a response of 1.

Through group lasso penalty, if one variable is significant, entire group members which the variable belongs to have nonzero coefficients. Therefore, this penalty makes it possible to select groups together.

In prognostic marker selection problem, a group can be genes that belong to
a pathway. However, in our predictive marker selection problem, since predictive markers are detected by gene by treatment interaction effects, a group consists of genes belong to a pathway and their interaction effects with treatment indicator variables. Genes and their interaction effects with treatments are in the same group so that between-group-correlation are not high for the group lasso to work properly [11].

However, when the penalty parameter is selected by minimizing prediction errors, which cross validation (CV) method uses, the group lasso is likely to select a model that is larger than the underlying model [38]. It results in relatively high false positive group selection rate. Stability selection (SS) can be an alternative method to reduce false positive group selection rate. However, it is likely to select overly smaller number of variables with low true positive group selection rate. As a result, the variables selected by SS tends not to predict outcome well. Therefore, in this paper, we apply PROMISE to the group lasso not only to have small false positive group selection rate but also to have good prediction accuracy.

The algorithm of PROMISE is in Figure 3.1. Since the algorithm is general, \( \lambda \) in the algorithm can be any regularization parameter. In Chapter 3, we used this algorithm to find \( \lambda \) in the lasso and the elastic net. And, in this chapter, we use the same algorithm to find \( \lambda \) for the group lasso.

Since the group lasso selects members in a group together, it selects entire genes in a pathway and their interaction effects with treatments. To select individual genes within the pathway and to see whether those have marginal effect or interaction effects with treatments, we further select individual variables using the lasso after selecting groups using the group lasso. We apply PROMISE for individual variable selection as well to significantly reduce false positives compared to CV and to have
select important variables to have good prediction accuracy.

The group lasso is implemented based on R package gglasso[39] and the lasso is implemented based on R package glmnet[22].

4.2 BATTLE data simulation

Pathway mapping and pre-screening We used the same BATTLE microarray data described in Section 3.3. Among 33297 probe sets in the battle microarray data, 22089 probe sets are mapped to genes. Among the 22089 genes, 6995 genes are mapped to 291 Kyoto Encyclopedia of Genes and Genomes (KEGG) pathways from the KEGG database (http://www.kegg.jp/kegg/rest/keggapi.html). Among the 6995 genes, we preselect 430 genes, whose marginal effects or interaction effects with a treatment are significant using uni-variate t-test with threshold p-value of 0.022. These 430 genes belong to 88 pathways. 293 genes belong to only one pathway and 137 genes belong to multiple pathways. Since the group lasso does not allow overlap in groups, those genes are needed to be assigned to only one pathway. To remove the overlap, we counted the number of connection between one of the 137 genes with other genes in each of the overlapping pathways using R package KEGGgraph [40]. Then, we assigned those 137 genes to one pathway in which the genes has the largest number of connections with other genes. A group consists of genes in a pathway and their interaction effects with 3 treatment indicators. Therefore, a size of group is 4 times of a pathway size. Median group size is 12.

After the pre-screening process, the number of variables, including the interaction effects with treatment indicator variables, is \(430 \times 4 = 1720\). The sample size is 101 among which 75 is used for training data set and 26 is used for test data set.

We simulated the treatment response using 3 different scenarios. For simulation
1, variables with nonzero coefficients are set as the 16 mostly selected variables by the group lasso + lasso using PROMISE in the real data analysis. The coefficients were set by averaging coefficients of the each variables from real data analysis. The coefficients are shown in the first row in Figure 4.1. For simulation 2, variables with nonzero coefficients are set as the 10 mostly selected variables and 5 medium selected variables. The coefficients are shown in the second row in Figure 4.1. Notice that the highly selected variables have coefficient higher than 2 and the medium selected variables have coefficient less than 2. In simulation 2, variables with nonzero coefficients are set as the 5 mostly selected variables and 10 medium selected variables. The coefficients are shown in the third row in Figure 4.1.
Figure 4.1: Absolute value of coefficients for variables with nonzero coefficients for BATTLE simulation 1, 2, and 3.

Right most side indicates the simulation number. Red lines are at $|\beta| = 2$, we call $|\beta| > 2$ large signals and $|\beta| < 2$ weak signals.
Table 4.1 : Prediction accuracy (AUC) of BATTLE simulations

<table>
<thead>
<tr>
<th>Method1</th>
<th>Method2</th>
<th>Simulation1</th>
<th>Simulation2</th>
<th>Simulation3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group Lasso</td>
<td>PROMISE</td>
<td>0.79(0.01)</td>
<td>0.81(0.01)</td>
<td>0.85(0.01)</td>
</tr>
<tr>
<td></td>
<td>CV</td>
<td>0.74(0.01)</td>
<td>0.8(0.02)</td>
<td>0.8(0.01)</td>
</tr>
<tr>
<td></td>
<td>SS</td>
<td>0.5(0)</td>
<td>0.54(0.02)</td>
<td>0.62(0.02)</td>
</tr>
<tr>
<td>Group Lasso + Lasso</td>
<td>PROMISE</td>
<td>0.81(0.01)</td>
<td>0.82(0.01)</td>
<td>0.92(0.01)</td>
</tr>
<tr>
<td></td>
<td>CV</td>
<td>0.83(0.01)</td>
<td>0.87(0.01)</td>
<td>0.92(0.01)</td>
</tr>
</tbody>
</table>

Table 4.2 : PPV of BATTLE simulations

<table>
<thead>
<tr>
<th>Method1</th>
<th>Method2</th>
<th>Simulation1</th>
<th>Simulation2</th>
<th>Simulation3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group Lasso</td>
<td>PROMISE</td>
<td>0.17(0.01)</td>
<td>0.2(0.01)</td>
<td>0.25(0.02)</td>
</tr>
<tr>
<td></td>
<td>CV</td>
<td>0.06(0)</td>
<td>0.06(0)</td>
<td>0.09(0)</td>
</tr>
<tr>
<td></td>
<td>SS</td>
<td>0(0)</td>
<td>0.25(0.03)</td>
<td>0.25(0.01)</td>
</tr>
<tr>
<td>Group Lasso + Lasso</td>
<td>PROMISE</td>
<td><strong>0.63</strong>(0.02)</td>
<td><strong>0.55</strong>(0.02)</td>
<td><strong>0.8</strong>(0.03)</td>
</tr>
<tr>
<td></td>
<td>CV</td>
<td>0.35(0.01)</td>
<td>0.29(0.01)</td>
<td>0.42(0.02)</td>
</tr>
</tbody>
</table>

Median number of selected variables. The numbers in parenthesis are the corresponding standard errors of medians estimated by using the bootstrap with 200 re-samplings on the 100 AUC.

Figure 4.2 shows group selection result. SS selects no or one group. While PROMISE and CV selects similar number of important groups, PROMOSE selects much less false positive groups than CV. Also, note that PROMISE selects the true groups most of time where as CV selects false groups more than half of times.

Figure 4.3 shows variable selection result using the group lasso. While PROMISE and CV selects similar number of true positives, CV selects much more false positives than PROMISE (3.5, 4.3, and 5.9 times, respectively). Notice that for PROMISE, most of false positives comes from the false positives in the important groups, while
for CV, the most of false positives comes from the false positives in the false groups. Since SS selects 0 or 1 group, it select 0 or 4 variables. Even though PROMISE selects important groups most of the times, there are still a large number of false positives due to the non important variables within the important group, even though PROMISE selects important groups most of the times. So we applied the group lasso with the lasso to reduce the false positives within important groups. After applying group lasso with the lasso, the number of false positives are significantly reduced (Figure 4.4) using either PROMISE or CV. However, again, PROMISE selects much less false positives than CV while selecting similar number of true positives. CV includes more false positives than true positives. Table 4.1 shows prediction accuracy. Using the group lasso, PROMISE shows a better prediction accuracy than CV. Prediction accuracy of SS is not good. Using Group lasso with the lasso shows better prediction accuracy than using the group lasso using either PROMISE or CV since the false positives harms prediction accuracy. CV shows similar or a slightly better prediction accuracy compared to PROMISE. Table 4.2 shows PPV. Using the group lasso, PROMISE shows higher PPV than CV but SS also shows similar PPV as PROMISE in simulation 2 and 3. Still, PROMISE outperforms SS in other measures. Using the group lasso with the lasso, PROMISE outperforms CV.

<table>
<thead>
<tr>
<th>Method</th>
<th>AUC</th>
<th>Strong signal</th>
<th>Weak signal</th>
<th>FP</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROMISE</td>
<td>0.85(0.01)</td>
<td>5(0)</td>
<td>1(0.07)</td>
<td>14.5(2.45)</td>
</tr>
<tr>
<td>CV</td>
<td>0.8(0.011)</td>
<td>5(0)</td>
<td>3(0.44)</td>
<td>85(6.83)</td>
</tr>
<tr>
<td>SS</td>
<td>0.62(0.02)</td>
<td>1(0)</td>
<td>0(0.07)</td>
<td>2(0.20)</td>
</tr>
</tbody>
</table>

Median number of AUC, the number of true positives for strong signals and weak signals, respectively, and the number of false positives. The numbers in parenthesis are the corresponding standard errors of medians estimated by using the bootstrap with 200 re-samplings on the 100 simulation results.
One interesting thing is that using group lasso, in terms of variable selection, PPV is increasing using PROMISE while it is almost the same using CV when the number of weak signal increases (Figure 4.5). The reason can be found in the result of simulation 3 (Table 4.3) when there are mixture of strong and weak signals, PROMISE focuses on selecting strong signals if outcome can be predicted well using only the strong signals. As a result, they catch all the strong signals having small number of false positives. However, CV still try to catch not only the strong signals but also the weak signals to maximize prediction accuracy. As a result, they select all the strong signals but only catch 3 weak signals and have huge number of false positives.

4.3 BATTLE data analysis

We analyze BATTLE data described in Section 4.2. In this section, the treatment outcome is from the BATTLE clinical trial.

<table>
<thead>
<tr>
<th>Method1</th>
<th>Method2</th>
<th>AUC</th>
<th># of variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group Lasso</td>
<td>PROMISE</td>
<td>0.7(0.02)</td>
<td>84(9.9)</td>
</tr>
<tr>
<td></td>
<td>CV</td>
<td>0.68(0.01)</td>
<td>224(13.74)</td>
</tr>
<tr>
<td></td>
<td>SS</td>
<td>0.5(0)</td>
<td>0(0)</td>
</tr>
<tr>
<td>Group Lasso + Lasso</td>
<td>PROMISE</td>
<td>0.66(0.01)</td>
<td>15.5(0.61)</td>
</tr>
<tr>
<td></td>
<td>CV</td>
<td>0.68(0.02)</td>
<td>29(0.9)</td>
</tr>
</tbody>
</table>

Median number of AUC and selected variables. The numbers in parenthesis are the corresponding standard errors of medians estimated by using the bootstrap with 200 re-samplings on the 100 runs

**Results** Table 4.4 shows the result. The group lasso result shows that although PROMISE predicts as good as CV, PROMISE produces a much more parsimonious solution than CV: PROMISE selects less than a half (84) the number of variables
selected by CV (224). SS does not select any variable most of time and the prediction accuracy is the same as random guessing.

The group lasso with the lasso shows that again, PROMISE selects almost half (15.5) number of variables selected by CV (29), while prediction accuracy is similar. In addition, when we compare the group lasso and the group lasso plus the lasso, the group lasso plus the lasso produces more sparse solution than the group lasso while prediction accuracy is not significantly different (p-value:0.55). Using PROMISE, the group lasso plus the lasso selects about only 20% of the variables selected by the group lasso.
Figure 4.2: Group selection accuracy of BATTLE simulations

Median number of selected groups. Blue indicates the median number of true positive groups. Red indicates the median number of false positive groups.
Figure 4.3: Variable selection accuracy of BATTLE simulations using the group lasso

Blue indicates variables that belong to relevant groups (to the response). Red indicates variables that belong to irrelevant groups.
Figure 4.4: Variable selection accuracy of BATTLE simulations using the group lasso with the lasso.

Blue indicates variables that belong to relevant groups (to the response). Red indicates variables that belong to irrelevant groups.
Figure 4.5: Comparison of Positive predictive value of BATTLE simulations using the group lasso
Chapter 5

Discussion and Future direction

Discussion  We have proposed PROMISE as a regularization parameter selection method to select predictive markers for personalized medicine. We compared PROMISE, CV, and SS using simulated and real data. While having prediction accuracy similar to that of CV, PROMISE outperforms CV in terms of variable selection. Three manifestations of this superior variable selection performance: (1) PROMISE produces a much more sparse solution than CV; (2) PROMISE selects variables more accurately than CV, producing fewer type I and type II errors than CV; and (3) PROMISE selects fewer noise variables than CV. In comparison, SS does not select any variable most of the time in both data sets.

CV tends to select too many variables to ensure good prediction accuracy. On the contrary, SS tends to select too few variables to ensure few false discoveries. However, PROMISE strikes a balance between the two, having good prediction accuracy and yielding few false positives. First, the sub-sampling method of PROMISE plays a role in reducing false discoveries. Second, using the cross-validation method in PROMISE to select the cutoffs \((\lambda_{\text{min}}, \pi_{\text{thr}})\) makes it possible to maximize the prediction accuracy and therefore includes important variables for that purpose.

Overall, PROMISE is the best method for selecting the regularization parameters to identify predictive markers in high-dimensional data. It significantly reduces false positives compared to CV so that the cost of biological experimental verification can be reduced, while maintaining good prediction accuracy so that each individual can
receive an optimal personalized treatment.

PROMISE can be applied to many parameter selection problems where the goals are both to minimize the false discoveries/the number of selected variables and to maximize the prediction accuracy/true discoveries. We applied PROMISE to binary outcomes; however, the method can be generalized to survival, Poisson, multinomial, or continuous outcome. In addition, we applied PROMISE along with the lasso and the elastic net, but it can be applied with other statistical learning methods. We are currently working on the application of PROMISE with the group lasso.

**Future direction** In the chapter 4, we employed biological information using pathway information to select biomarkers. To further employ more biological information, we want to use relationship between genes. For this, we can use KEGG pathway maps, which contains nodes(genes) and edges and their relation with direction. We can model KEGG pathway maps using such as directed acyclic graph [41]. Using the graphical model accompanied with more biological knowledge, we expect to select important markers and predict treatment outcome more accurately.
Bibliography


