IMPROVING THE ACCURACY OF VARIABLE SELECTION USING THE WHOLE SOLUTION PATH

Yang Liu

A Dissertation

Submitted to the Graduate College of Bowling Green
State University in partial fulfillment of
the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2015

Committee:

Hanfeng Chen, Advisor

Jonathan David Bostic,
Graduate Faculty Representative

Peng Wang

James Albert

ABSTRACT

Hanfeng Chen, Advisor

The performances of penalized least squares approaches profoundly depend on the selection of the tuning parameter; however, statisticians did not reach consensus on the criterion for choosing the tuning parameter. Moreover, the penalized least squares estimation that based on a single value of the tuning parameter suffers from several drawbacks. The tuning parameter selected by the traditional selection criteria such as AIC, BIC, CV tends to pick excessive variables, which results in an over-fitting model. On the contrary, many other criteria, such as the extended BIC that favors an over-sparse model, may run the risk of dropping some relevant variables in the model.

In the dissertation, a novel approach for the feature selection based on the whole solution paths is proposed, which significantly improves the selection accuracy. The key idea is to partition the variables into the relevant set and the irrelevant set at each tuning parameter, and then select the variables which have been classified as relevant for at least one tuning parameter. The approach is named as Selection by Partitioning the Solution Paths (SPSP). Compared with other existing feature selection approaches, the proposed SPSP algorithm allows feature selection by using a wide class of penalty functions, including Lasso, ridge and other strictly convex penalties.

Based on the proposed SPSP procedure, a new type of scores are presented to rank the importance of the variables in the model. The scores, noted as Area-out-of-zero-region Importance Scores (AIS), are defined by the areas between the solution paths and the boundary of the partitions over the whole solution paths. By applying the proposed scores in the stepwise selection, the false positive error of the selection is remarkably reduced.

The asymptotic properties for the proposed SPSP estimator have been well established. It is showed that the SPSP estimator is selection consistent when the original estimator is either estimation consistent or selection consistent. Specially, the SPSP approach on the Lasso has been proved to be consistent over the whole solution paths under the irrepresentable condition.

Additionally, a number of simulation studies have been conducted to illustrate the performance

of the proposed approachs. The comparison between the SPSP algorithm and the existing selection criteria on the Lasso, the adaptive Lasso, the SCAD and the MCP were provided. The results showed the proposed method outperformed the existing variable selection methods in general.

Finally, two real data examples of identifying the informative variables in the Boston housing data and the glioblastoma gene expression data are given. Compared with the models selected by other existing approaches, the models selected by the SPSP procedure are much simpler with relatively smaller model errors.

ACKNOWLEDGMENTS

I would like to express my sincere appreciation and thanks to my academic advisor Prof. Peng Wang for the guidance on my PhD studying and dissertation research. His advices on both research as well as on my career have been priceless. Without his supervision and constant help, this dissertation would not have been possible.

I would like to thank Prof. Hanfeng Chen for his kind help in these years. Since Prof. Peng Wang went to the University of Cincinnati in my 4th PhD year, thus I was required to find a new chair in my Dissertation Committee officially to substitute Dr. Peng Wang while Dr. Peng Wang would continue his role as my advisor to supervise my dissertation research. Prof. Hanfeng Chen kindly took over the Committee chair position. I really appreciate his kindness and great help on the arrangement.

I would also like to thank Prof. James Albert for his insightful suggestions and understanding. I would like to thank Prof. Jonathan David Bostic for his time for serving as GFR in my committee.

Besides, I would like to thank all the other professors in the Math & Stats Department at BGSU, for the great help in my PhD studying, the brilliant suggestions on my research work.

In addition, I want to express my thanks to Seubert Marcia, Busdeker Mary and Berta Barbara for their kind helps these years.

A special thanks to my family for supporting me throughout my life. At the end I would like to thank all of my friends for encouraging me to strive towards my goal, for supporting me in writing the thesis and finding a job, for all the fun time we had in my PhD studying years.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 INTRODUCTION

1.1 Background and Motivation

Variable selection, also known as feature selection, is a procedure where a subset of relevant features are selected for constructing models. In the past two decades, feature selection has been one of the focuses of the statistical and machine learning research, mainly because data with hundreds of thousands of variables have largely occurred in application domains. Two typical examples are the gene selection of microarray data and text classification. In the gene selection problems, the variables are usually mRNA expressions from only a small number of patients. In a typical study, there are only fewer than 100 patients, while the number of variables, mRNA expressions, could range from several thousands to hundreds of thousands. For instance, the sample sizes of the two data sets in the glioblastoma microarray gene expression study of Horvath et al. (2006, [31]) are 55 and 65 respectively, while the number of the genes in both datasets is 3600. As a result, some initial screening procedures are often applied before the gene selection. In the text classification problems, the variables are the frequencies of hundreds of thousands of words. An initial pruning could reduce the number of words to around 15,000 ([28]).

There are many potential benefits of feature selection. First, it helps to defy the curse of dimensionality by greatly reducing the time of training models. Moreover, it also facilitates data visualization and understanding, since it provides an estimate of the important features. Apparently, it also avoids over fitting problems to improve the sample prediction accuracy.

In practice, many traditional variable selection approaches such as stepwise selection and best subset selection have been widely applied. Stepwise selection, although simple and straightforward, suffers from the fact that the order a certain variable enters or leaves the model heavily influences the final results. Meanwhile, the best subset selection always requires exhaustive search. The computational cost can be huge if the number of the variables is large.

Due to the limitations of the traditional approaches, the penalized likelihood approach has been

a common approach for variable selection problems in the past two decades. The general idea of the approach is to penalize the model fitting with some regularization terms to produce a sparse model. Consider the following linear regression model,

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta}^* + \boldsymbol{\varepsilon}; \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}), \tag{1.1}$$

where $\mathbf{y} = (y_1, \cdots, y_n)^T$ is an n-dimensional response vector, $\boldsymbol{\beta}^* = (\beta_1^*, \cdots, \beta_p^*)^T$ is a p-dimensional vector of regression coefficients, $\mathbf{X} = (\mathbf{x_1}, \cdots, \mathbf{x_p})$ is an $n \times p$ design matrix, and $\boldsymbol{\varepsilon} = (\epsilon_1, \cdots, \epsilon_n)^T$ is an n-dimensional vector of i.i.d. random errors. Without loss of generality, we can assume in model (1.1), $\mathbf{x}_j, j = 1, \cdots, p$ are standardized and the response $\mathbf{y}$ are centered, i.e.,

$$\sum_{i=1}^{n} x_{ij} = 0, \sum_{i=1}^{n} x_{ij}^2 = 1, j = 1, \cdots, p, \sum_{i=1}^{n} y_i = 0.$$

For linear regression problems described in (1.1), the penalized likelihood approach is equivalent to the penalized least squares (PLS) regression, where the coefficients are estimated by minimizing the following objective function:

$$\frac{1}{n}\|\mathbf{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda \sum_{j=1}^{p} J(|\beta_j|), \tag{1.2}$$

where $J(\cdot)$ is a penalty function that controls the number of nonzero coefficients, and $\lambda > 0$ is a tuning parameter. Unlike traditional variable selection procedures, the penalized least squares approach is computationally efficient since it can carry out variable selection and estimation simultaneously. This is because the objective function 1.2 automatically shrinks estimates of some coefficients to zeros.

Furthermore, theoretical properties of the penalized least squares estimators are established when the tuning parameter is appropriately chosen (Fan and Li, 2001, [16]; Fan and Peng, 2004, [20]; Zhao and Yu, 2006, [64]; Zou, 2006, [65]; Zhang, 2010, [61]; Bühlmann and Van De Geer, 2001, [8]).

Apparently, the estimator of $\boldsymbol{\beta}$, denoted by $\hat{\boldsymbol{\beta}}(\lambda) = \{\hat{\beta}_1(\lambda), \ldots, \hat{\beta}_p(\lambda)\}^T$, is a function of the tuning parameter λ once the penalty function $J(\cdot)$ is specified. We name the function $\hat{\beta}_j(\lambda), j = 1, \cdots, p$ as the solution path of the coefficient β_j. In practice, one can compute $\hat{\boldsymbol{\beta}}(\lambda)$ for a number of different values of λ to obtain the solution paths of all the coefficients and then choose a tuning parameter λ using some type of criterion. The purpose of the tuning criterion is to find a λ that balances the fit and the complexity of the model. Therefore, one need to specify both a penalty function and a criterion to select the tuning parameter λ in order to carry out variable selection with the penalized regression approach.

Much research has been devoted to the development of the penalty function. In general, there are two classes of penalty functions, convex penalties and non-convex penalties. The L_1 penalty, referred as the Lasso (Least absolute shrinkage and selection operator, Tibshirani, 1996, [52]), is probably the most commonly used convex penalty. Zou (2006, [65]) proposed the adaptive Lasso approach which corrects the bias of the Lasso for nonzero regression coefficients by adding a weight to the L_1 penalty. As for the non-convex penalties, Fan and Li (2001, [16]) proposed the smoothly clipped absolute deviation (SCAD) penalty by using a quadratic spline function with knots at λ and $a\lambda$, where $a > 2$ is a constant; Zhang (2010, [63]) proposed the minimax concave penalty (MCP) by minimizing the maximum concavity of the model for variable selection and unbiasedness; Shen et al. (2012, [48]) also proposed the truncated L_1 penalty (TLP) as a surrogate of the L_0 penalty. These non-convex penalties all enjoy the oracle property in the sense that estimators obtained by applying these penalties are as efficient as if the nonzero coefficients are already known.

Another aspect of the feature selection involves the selection of the tuning parameter. Some general selection criteria include cross validation ([50]), generalized cross validation ([13]), AIC ([2]), BIC([47]), GIC ([63]). Chen and Chen (2008, [11]) pointed out that these criteria usually identify too many irrelevant features when the number of variables is large. Such phenomenon has also been described in Broman and Speed (2002, [7]), Siegmund (2004, [49]) and Bogdan et al. (2004, [6]) in their studies of quantitative loci mapping. Chen and Chen (2008, [11]) proposed the extended BIC (EBIC), which promotes model sparsity by adjusting BIC with an additional penalty

term for the growing number of parameters in the model. Recently, Sun et al. (2012, [51]) also proposed a new technique via variable selection stability, which directly focuses on the selection of the informative variables.

Although the above criteria have been well studied for more than a decade, there has been no concurrence of opinion on which criterion to employ for the choice of the tuning parameter. See, for examples, Table 1.1 for a list of publications on major statistics and machine learning journals and the different criteria they use. In fact, the currently used feature selection procedure, using only one chosen value for the tuning parameter, may suffer from inevitable drawbacks that it is often impossible to correctly identify all the features, no matter which criterion we use. In the following, we demonstrate these drawbacks with both a simulated example and a real data example.

Table 1.1: The general selection criteria in statistics and machine learning literature.

Criteria	References
cv	Adaptive Lasso Zou (2006) [65], JASA Wong et al. (2013) [58], ICML Fused Lasso Tibshirani et al. (2005) [54], JRSSB TLP Shen et al. (2012) [48], JASA
gcv	Lasso Tibshirani (1996) [52], JRSSB SCAD Fan and Li (2001) [16], JASA
AIC	Hurvich and Tsai (1989) [33], Biometrika
BIC	Wang et al. (2007) [56], JRSSB Yuan and Lin (2007) [60], Biometrika
EBIC	Tilting Cho and Fryzlewicz (2012) [12], JRSSB Group lasso Huang et al. (2010) [32], AOS

We would use the following simulated example to illustrate the aforementioned problem. Suppose there are 10 nonzero (relevant) variables and 30 zero (irrelevant) ones in the model (1.1), where the coefficients of these 10 nonzero variables are $\beta_1^* = \cdots = \beta_5^* = 3, \beta_6^* = \cdots = \beta_{10}^* = -2$ The entries of the variables $\mathbf{x}_j, j = 1, \cdots, p$ are generated from the standard normal distribution. The pairwise correlation between the first 10 variables is 0.9. The remaining 30 variables are independent with each other, and are also independent with the first 10 variables. Furthermore, we generate the error from the normal distribution $N(0, 3^2)$ and we set the sample size to be $n = 50$. This example

is proposed by Wang et al. (2011, [57]), which is designed to study the performance of the existing variable selection methods for the data with complicated correlation structure.

We apply the R package *lassoshooting* to obtain a set of parameter estimators $\hat{\boldsymbol{\beta}}(\lambda)$ at each value of λ and plot the Lasso solution paths in Figure 1.1. We pick the grid of the tuning parameters on the log scale, therefore we use the log values of the tuning parameter as the x-axis in the plot. The dashed lines in Figure 1.1 represent the solution paths for nonzero variables and the solid lines represent those for the zero ones. The tuning parameters chosen by the 2-fold CV, GCV, AIC, BIC and the extended BIC (EBIC) are shown by the vertical lines. We report the total number of the selected variables, the number of false positives (FP, the number of selected zero variables) and the number of false negatives (FN, the number of missed nonzero variables) by these criteria in Table 1.2. Here the true model is known, therefore we also record the result of the "oracle" selection in a sense that we select the best possible tuning parameter which minimizes the number of incorrect selections (i.e., the number of selected zero variables + the number of missed nonzero variables).

We observe that CV, GCV, AIC, and BIC tend to select too many spurious variables and the extended BIC tends to drop most of the nonzero variables. Even for the "oracle" selection, many nonzero variables are excluded in the model. The problem looks more evident when we focus on three lines in the lower panel of Figure 1.1. Here the two dotted lines (1 and 3) are the solution paths of two nonzero coefficients, while the solid line 2 is the solution path for an zero one. Apparently selecting a small λ, as AIC BIC and GCV do, misleads us to identifying all the three coefficients as nonzero. On the other hand, a large λ, as CV, EBIC and Oracle select, incorrectly shrinks both coefficients of the nonzero variables to zero. As a matter of fact, it is impossible to correctly identify all the three features regardless of the value of the tuning parameter we choose, although one can even tell the differences between the three features by simply observing the solution paths.

To better illustrate the limitations of selecting just one tuning parameter, we would use the glioblastoma gene expression data by McLendon et al. (2008, [38]) as a motivating example. In the dataset, by using all the censor subjects in the data and taking the logarithm of the survival time as the response variable, we obtain a dataset with $n = 185$ subjects and $p = 930$ genes. To iden-

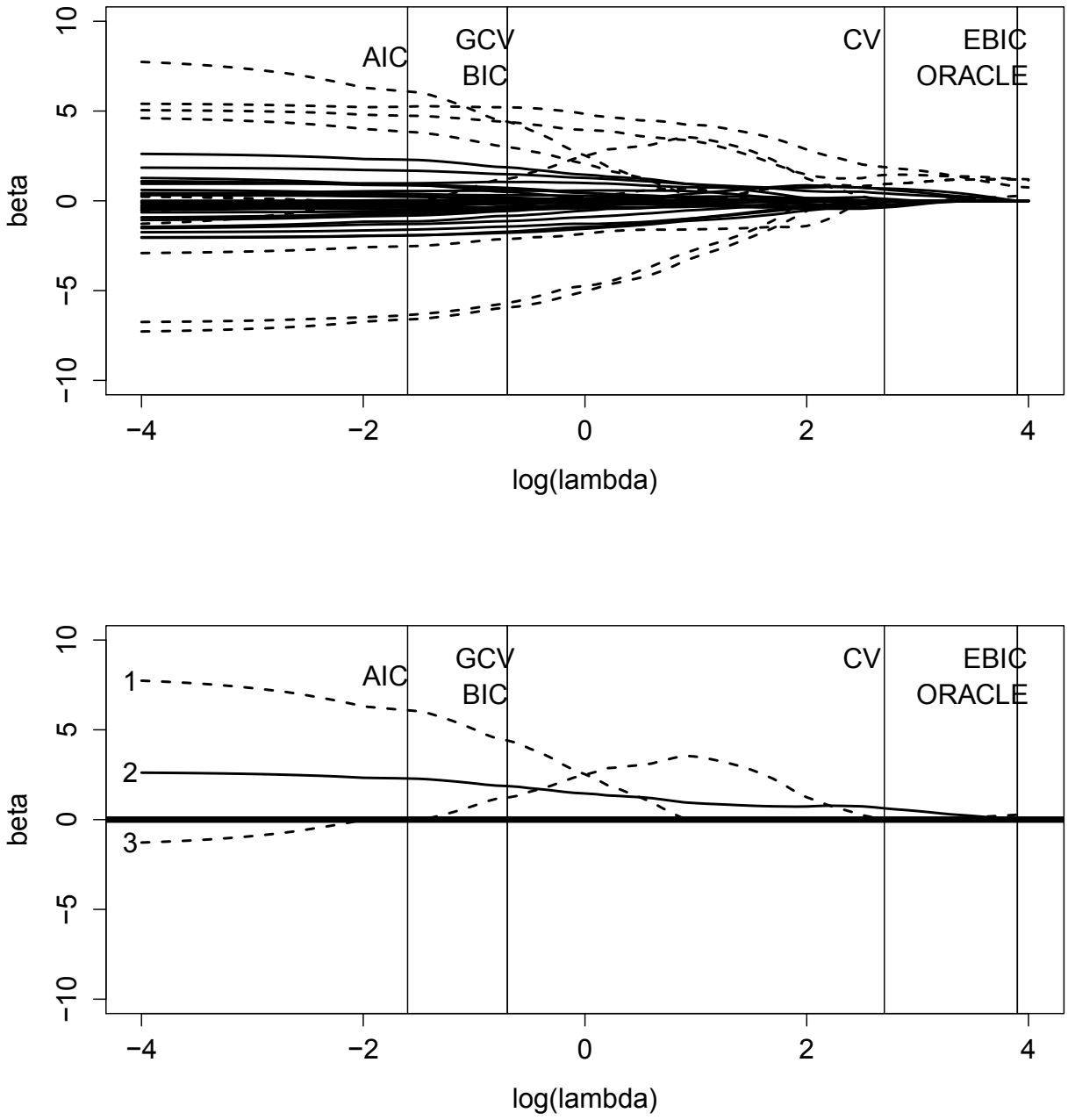

Figure 1.1: Top: The Lasso solution paths of the simulated example. The dashed lines are the paths of the 10 nonzero variables while the black lines are the paths of the 30 zero variables. The vertical lines represent the choices of the tuning parameter by the criteria: CV, GCV, AIC, BIC, EBIC, ORACLE. Bottom: The Lasso solution paths of the nonzero variables "1", "2" and the zero variable "3". The bold line represents the zero line.

Table 1.2: The number of the selected variables, the number of false positives (FP), the number of false negatives (FN) of the criteria: CV, GCV, AIC, BIC, EBIC, Oracle. Note that the true model contains 10 nonzero variables and 30 zero variables

	CV	GCV	AIC	BIC	EBIC	Oracle
Total number	16	35	37	35	4	4
FP	12	27	30	27	0	0
FN	6	2	3	2	6	6

tify the highly informative genes to explain the glioblastoma tumor behavior, we consider the linear model (1.1) and apply the Lasso approach for selecting the relevant genes.

The solution paths of the Lasso on the glioblastoma gene expression data are shown in Figure 1.2. To better describe the problem, we focus on the solution paths of two genes: the gene "IGF1" (insulin-like growth factor-1) and the gene "FRAT1" (frequently rearranged in advanced T cell lymphomas-1). Previous studies have demonstrated that these two genes play important roles in the glioblastoma behavior (See [35] and [27]). However, as shown in the second panel of Figure 1.2, selecting a small tuning parameter would ignore "FRAT1" while choosing a larger value of the tuning parameters would miss the gene "IGF1". We also observe that the none of CV, GCV, AIC, BIC, or EBIC is able to identify both genes.

The above restriction of utilizing just one tuning parameter could seriously reduce the accuracy of the feature selection in general, since solution paths like those in Figure 1.1 and Figure 1.2 happen quite often no matter which penalty we employ. This is especially true when there exist large correlations among the variables or the dimensions of the features are extremely high ([18], [19]). To overcome this restriction, we develop an innovative and intuitive approach, which utilizes the whole solution paths to improve the selection accuracy. Our approach can correctly identify the relevance of the features like those 3 ones (the labeled lines 1, 2, and 3) in Figure 1.1.

We achieve the objective by developing a partitioning rule that cuts the whole solution paths into two regions, namely "zero region" and "nonzero region". First, we develop a new clustering method which divides all the variables into two clusters, the relevant set and the irrelevant set, for each value

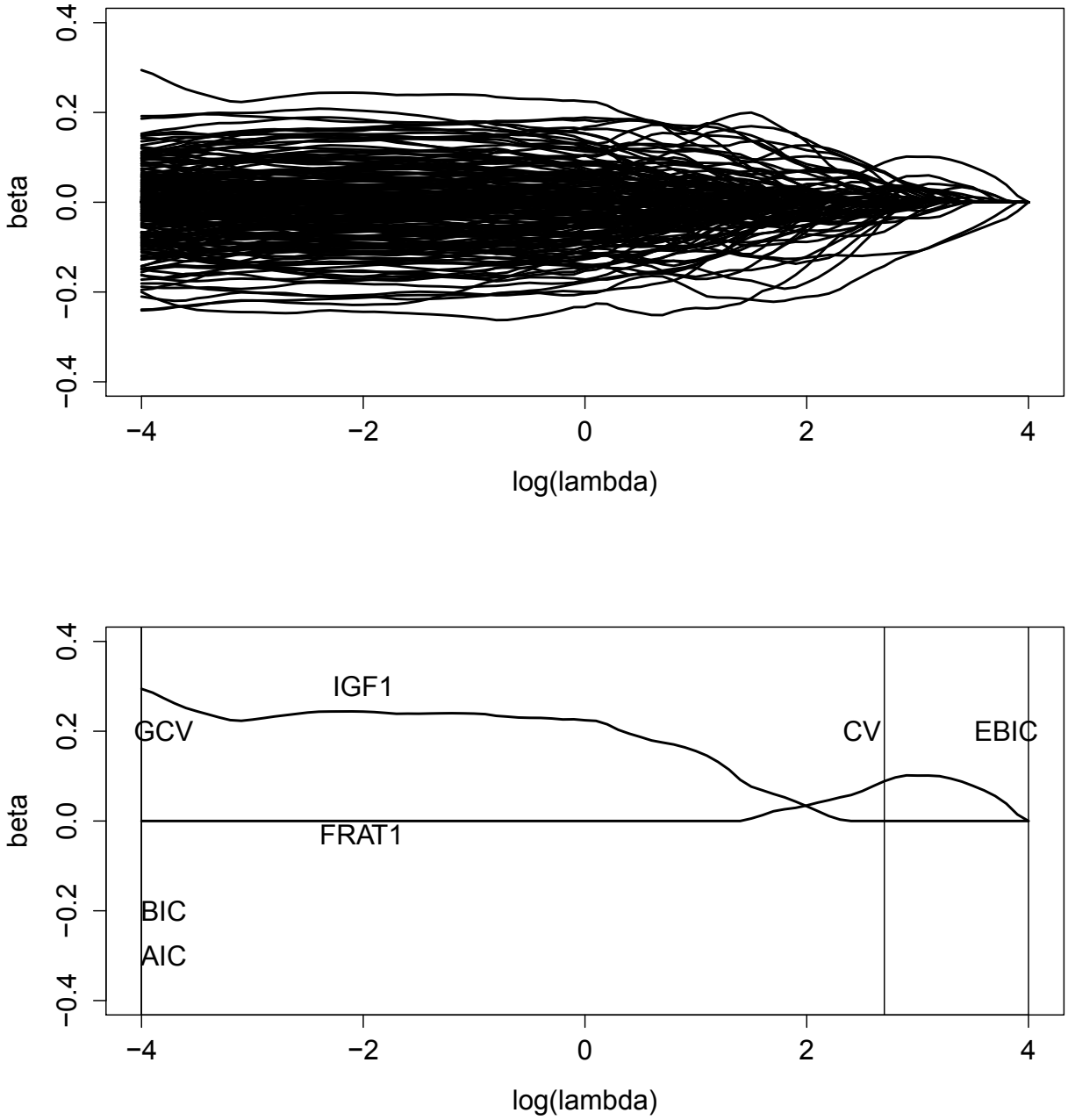

Figure 1.2: Top: The Lasso solution paths of the glioblastoma gene expression data. There are 930 lines in the plot. Bottom: The Lasso solution paths of the genes "IGF1" and "FRAT1". The vertical lines represent the choices of the tuning parameter by the criteria: CV, GCV, AIC, BIC, EBIC.

of the tuning parameter λ. Then the whole plot of the solution paths can be partitioned into two regions by the red curves as shown in Figure 1.3 (the detailed partitioning rule will be introduced in Chapter 2). We name the region inside the two red curves as the zero region, and that outside as the nonzero region. Finally, we choose all the variables, which have been identified as a relevant variable for at least one value of λ, as the important features. We consider a feature unimportant if its solution path never goes out of the zero region. We name the above procedure *selection by partitioning the solution paths (SPSP)*. It can be well observed from Figure 1.3 that this SPSP procedure correctly selects 9 out of the 10 relevant variables and drops all irrelevant ones, outperforming the result from any single value of λ in terms of selection accuracy. Another advantage of the SPSP is that it does not require the coefficients of the unimportant variables shrunk to zero and it allows us to carry out feature selection with just a ridge regression.

We consider a feature important even if its solution path enters the nonzero region just once. The strategy may seem aggressive in identifying relevant variables. This is because, we start the SPSP process rather conservative, in the sense that for the smallest value of λ, we consider every variable "unimportant". We initiate the partitioning process with the smallest λ. Then the clustering at a larger value of λ depends on the results from the previous λ. Therefore, the SPSP procedure combines a conservative starting point with an aggressive selection strategy to optimize the selection accuracy.

The SPSP procedure is connected with the stability selection approach, proposed in a discussion paper by Meinshausen and Bühhmann (2010, [41]). Their approach is based on the probabilities of the variables being selected, and these probabilities are obtained from generic sub-sampling approach. Therefore the stability selection does not require the selection of the tuning parameter and also utilizes the information of the whole solution paths. However, the SPSP procedure would work on any shrinkage penalty function, while with stability selection, one would still need to employ a penalty that can shrink coefficients to zero. Moreover, the computational cost of the SPSP procedure is much smaller as no sub-sampling is involved and we only need to compute the solution paths once. In addition, the cut off probability in stability selection is arbitrary, while for SPSP, the

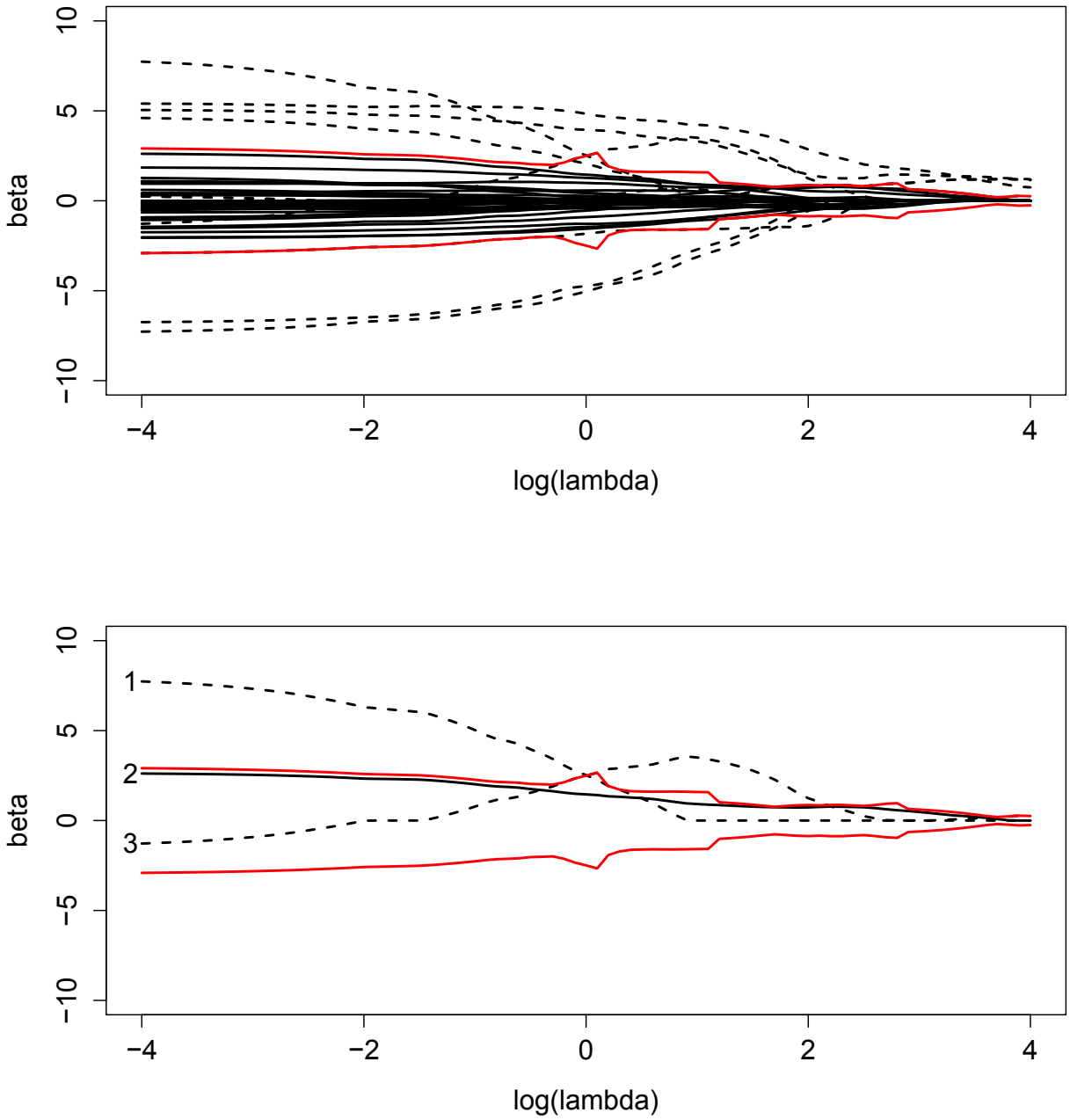

Figure 1.3: Top: Partitions on the Lasso solution paths of the same simulated example. The red lines are the partitioning curves. Bottom: Partitions on the Lasso solution paths of nonzero variables “1”, “3” and zero variable “2”.

constant playing the similar role is data-adaptive. Finally, we find from simulation studies that the stability selection tends to select too few variables, therefore produces a higher false negative rate compared to the SPSP.

The work in this dissertation is also remotely related to Bayesian variable selection approaches, where the tuning parameters or candidate models are assigned a prior distribution, and the posterior distributions of the models are evaluated. A related idea of applying a collection of models for variable selection is the Bayesian model averaging approach, which calculates the posterior probability that a variable enters the model by averaging over all of the models (See [30], [46], [44]). Another similar idea is from Barbieri and Berger (2004, [4]), which constructs the posterior inclusion probabilities for all the features using Bayesian model averaging technique. The final model would include all the variables whose posterior probabilities of being in the model are 0.5 or higher. The so-called probability median model also has the flavor of utilizing the results from different tuning parameters, rather than just choosing one of them.

Furthermore, we also develop an original type of scores to rank the importance of the variables based on the partitions of the solution paths. These scores provide precise directions on which variables to keep, in case one would like to monitor the complexity of the model. Specifically, we consider a feature more important if its solution path is farther out of the zero region, and less important if it is farther inside the zero region. Hereafter, we refer these scores as *area-out-of-zero-region importance scores (AIS).*

1.2 Literature Review

In this section, we provide a brief introduction of the previous research in the penalized least squares estimation. The first part presents several penalty functions which are widely applied for variable selection. The second part provides the general selection criteria for selecting the tuning parameter.

1.2.1 Penalty Functions

Extensive research has been devoted to the development of the penalty functions. In general, two classes of penalty functions are presented, convex penalties and non-convex penalties. In this section, we will introduce four penalty functions: the Lasso, the adaptive Lasso, the SCAD and the MCP in details and apply these penalty functions in the simulation part of this dissertation. Note that the former two functions are convex penalties while the latter two functions are non-convex penalties. Furthermore, some other popular penalty functions are also listed in the last part of this section.

Lasso

Tibshirani(1996, [52]) proposed the L_1 norm as the penalty to shrink the variables into zero, which is famously known as Lasso, an acronym for Least Absolute Shrinkage and Selection Operator. In variable selection, the Lasso has become very popular for its computational feasibility with relatively high selection and prediction accuracy.

Specially, the Lasso estimator can be solved by the following optimization problem,

$$\hat{\boldsymbol{\beta}}^{(Lasso)} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{n}||\mathbf{y} - \boldsymbol{X}\boldsymbol{\beta}||_2^2 + \lambda||\boldsymbol{\beta}||_1,$$

where $||\boldsymbol{\beta}||_1 = \sum_{j=1}^{p} |\beta_j|$ is the L_1 norm of the parameters and λ is the tuning parameter. The optimization problem is convex and has an equivalent form as

$$\hat{\boldsymbol{\beta}}^{(Lasso)} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{n}||\mathbf{y} - \boldsymbol{X}\boldsymbol{\beta}||_2^2 \text{ , subject to } ||\boldsymbol{\beta}||_1 \leq t,$$

where t is a constant which has a data-dependent one-to-one relation with λ.

The Lasso can perform variable selection and estimation simultaneously in the sense that some coefficients can be shrunk into exactly zero. Under the orthonormal design, i.e., $\frac{1}{n}\mathbf{X^T X} = \mathbf{I}_p$, the

Lasso estimator equals the soft-thresholding estimator,

$$\hat{\beta}_j^{(Lasso)} = sign(Z_j)(|Z_j| - \lambda/2)_+,$$

where $Z_j = (\mathbf{X^T Y})_j/n, j = 1 \cdots, p$ is the ordinary least squares estimator and "+" means the positive part. Clearly, the Lasso estimator shrinks the coefficient whose OLS estimator is less than $\lambda/2$ into zero.

Many theoretical results have been established on the prediction and selection consistency of the Lasso estimator. One main result is that the Lasso is selection consistent if and only if the neighborhood stability (Meinshausen and Bühlmann, 2006, [40]) or the irrepresentable condition (Zhao and Yu, 2006, [64]) holds.

A number of algorithms for the computation of the Lasso estimators have been proposed. For instance, the coordinate descent algorithm is very efficient for solving the Lasso estimators, especially for high dimensional problems. It updates the estimator fast in a coordinate-wise way until numerical convergence. More details about the algorithm can be found in Friedman et al. (2007, [23]). In addition, the shooting algorithm, proposed by Fu (1998, [26]), is also a special case of the coordinate descent algorithm. We will employ this shooting algorithm for the computation of the Lasso in this dissertation.

Adaptive Lasso

The selection consistency of the Lasso requires strong conditions, which can be easily violated in practice. To reduce the bias of the Lasso estimator, Zou (2006, [65]) proposed the adaptive Lasso, which adds a weight on the L_1 norm to adjust the penalties. It is defined as

$$\hat{\boldsymbol{\beta}}^{(ALasso)} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{n}||\mathbf{y} - \boldsymbol{X\beta}||_2^2 + \lambda \sum_{j=1}^{p} \hat{w}_j \beta_j,$$

where $\hat{\mathbf{w}} = (\hat{w_1}, \cdots, \hat{w_p})$ is a weight vector. Usually, if $\hat{\boldsymbol{\beta}}$ is a root-n-consistent estimator such as the ordinary least squares (OLS) estimator, one can define the weight as

$$\hat{\mathbf{w}} = \frac{1}{|\hat{\boldsymbol{\beta}}|^{\gamma}}, \text{ for some } \gamma > 0.$$

Note that if $\hat{\beta}_j = 0$, $\hat{\beta}_j^{(ALasso)} = 0$; and if $\hat{\beta}_j$ is large, the weight will be small, which means we apply a small penalty (shrinkage) on this coefficient. In high-dimensional problems, we can pick the Lasso or the ridge estimator with a small shrinkage parameter as the initial estimator.

Theoretically, if p is fixed in the model, the adaptive Lasso can achieve the selection consistency under weak conditions while the Lasso requires stronger conditions (See [65]). In other words, the adaptive Lasso can be consistent for cases where the Lasso is inconsistent.

The computation of the adaptive Lasso is the same as the Lasso after applying the following transformation:

$$\mathbf{X}^{(w)} = X \cdot W,$$

where W is a $p \times p$ diagonal matrix with diagonal elements $\hat{w_1}, \cdots, \hat{w_p}$. Then we can solve the following Lasso-type problem

$$\hat{\boldsymbol{\beta}}^{(w)} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{n} ||\mathbf{y} - \mathbf{X}^{(w)}\boldsymbol{\beta}||_2^2 + \lambda ||\boldsymbol{\beta}||_1,$$

and the solution for the adaptive Lasso is $\hat{\boldsymbol{\beta}}^{(ALasso)} = W \cdot \hat{\boldsymbol{\beta}}^{(w)}$.

It is worth mentioning that the adaptive Lasso does not always outperform the Lasso since the result of the adaptive Lasso heavily depends on the initial weight. When the OLS estimation (or the ridge estimation) is not highly accurate, the adaptive Lasso can have a worse result than the result of the Lasso.

SCAD

Fan and Li (2001, [16]) advocated that a good penalty function should satisfy the following three properties:

1. Unbiasedness: The resulting estimator is nearly unbiased especially for the large coefficients to reduce the modeling bias.

2. Sparsity: The resulting estimator should set small estimated coefficients to zero automatically on the purpose of variable selection.

3. Continuity: The resulting estimator should be continuous in the data to avoid instability in prediction.

It is shown that the L_q penalty with $0 \leq q < 1$ does not satisfy the continuous condition, the L_1 penalty (Lasso) does not satisfy the unbiasedness condition, and the L_q penalty with $q > 1$ does not satisfy the sparsity condition. Hence, none of the L_q penalties can satisfy these three conditions simultaneously.

According to these three properties, Fan and Li (2001, [16]) proposed the smoothly clipped absolute deviation (SCAD) penalty, which is a non-convex function corresponding to the quadratic spline with knots at λ and $a\lambda$. The first derivative of the SCAD penalty can be given as

$$J'_S(|z|) = I(|z| < \lambda) + \frac{(a - |z|/\lambda)_+}{a-1} I(|z| > \lambda),$$

where $a > 2$ is a constant which can be chosen by the cross validation. Fan and Li (2001, [16]) also suggested to set a as 3.7 for most problems.

The SCAD penalty satisfies the above three properties and theoretical results show that there exist some local SCAD minimizers which are selection consistent under some strong conditions on the design matrix (See [34] for more details).

Generally, the SCAD estimator can be computed by the local linear approximation (LLA, [67]) or the local quadratic approximation (LQA, [16]).

MCP

Zhang (2010, [61]) proposed a MC+ method, which contains two parts: a minimax concave penalty (MCP) and a penalized linear unbiased selection (PLUS) algorithm. The MCP provides the minimum non-convexity of the penalized loss given the level of bias, which can remedy the bias problem of the Lasso. Mathematically, the MCP is defined as

$$J_M(|z|) = \int_0^{|z|} (1 - \frac{x}{\gamma\lambda})\mathrm{d}x,$$

where $\gamma > 0$ is a regularization parameter. It minimizes the maximum concavity of the model subject to some unbiasedness conditions.

Zhang (2010, [61]) established the selection consistency of the MCP estimator under a strictly global convexity condition and the estimator can be efficiently computed by the PLUS algorithm.

Other Popular Penalties

In addition to those mentioned above, we also simply list some other popular ones as follows.

1. L_0 penalty (Classical Variable Selection):

$$\hat{\boldsymbol{\beta}}^{(L_0)} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{n}||\mathbf{y} - \boldsymbol{X}\boldsymbol{\beta}||_2^2 + \lambda \sum_{j=1}^{p} I(\beta_j \neq 0).$$

   Note that the L_0 penalty is equivalent to the traditional feature selection method: the best subset. Most classical model selection criteria can be casted into this framework, such as the Akaike information criterion (AIC) with $\lambda = 2\sigma^2/n$, the Bayesian information criterion (BIC) with $\lambda = \log(n)\sigma^2/n$, the risk inflation criterion (RIC) with $\lambda = 2\log(p)\sigma^2/n$; however, the estimation is computationally infeasible since the L_0 penalty is non-convex and discontinuous at 0. Normally we have to apply the exhaustive search in practice.

2. Truncated L_1 penalty (TLP, [48]):

$$\hat{\boldsymbol{\beta}}^{(TLP)} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{n}||\mathbf{y} - \boldsymbol{X}\boldsymbol{\beta}||_2^2 + \lambda \sum_{j=1}^{p} \min(\frac{|\beta_j|}{\tau}, 1),$$

where $\tau > 0$ is a threshold parameter. The TLP penalty actually is a surrogate of the L_0 penalty. It provides a good approximation of the L_0 penalty especially when $\tau \to 0^+$ with significant computational advantages due to its piecewise linearity.

3. L_q penalty:

$$\hat{\boldsymbol{\beta}}^{(L_q)} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{n}||\mathbf{y} - \boldsymbol{X}\boldsymbol{\beta}||_2^2 + \lambda \sum_{j=1}^{p} |\beta_j|^q.$$

The general L_q penalty with $0 < q < 2$ leads to a bridge regression ([22], [26]). Whereas, the estimator does not produce sparse solutions for $q > 1$.

4. Elastic net penalty ([66]):

$$\hat{\boldsymbol{\beta}}^{(Elastic)} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{n}||\mathbf{y} - \boldsymbol{X}\boldsymbol{\beta}||_2^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} |\beta_j|^2,$$

where λ_1 and λ_2 are two tuning parameters. The elastic net actually is a linear combination of the L_1 and the L_2 penalties, which is proposed to overcome the limitations of the L_1 penalty.

1.2.2 Selection of the Tuning Parameter

Another aspect of the penalized least squares approach involves the selection of the tuning parameter. In this subsection, we begin with the introduction of the traditional selection criteria such as cross validation, generalized cross validation, AIC and BIC. We further present the extended BIC ([11]), which adjusts BIC to produce a sparse model. Finally we introduce the stability selection, proposed by Meinshausen and Bühhmann (2010, [41]), which bases on the subsampling technique to avoid the problem of choosing the tuning parameter in the penalized least squares estimation.

Cross Validation

The cross validation (CV, [50]) is a useful technique to access the accuracy of a statistical model. Given a dataset, we can divide the dataset into a training set and a test set. We model the training data only to obtain the estimator, and then apply the estimator to predict the values in the test set. The best model chosen by the cross validation should have the best prediction performance among the candidate models.

In practice, the k-fold cross validation is widely used for model selection. In the k-fold cross validation, the dataset is split into k equal size sets. In these k sets, we will leave one set as the test set and use the remaining $k-1$ as the training set. By taking each of these k sets exactly once as the test set, we repeat the process k times and evaluate the average prediction error in these k results.

In the penalized least squares estimation, we denote the whole dataset as T. In the k-fold cross validation, we need repeat the process k times. Using the same notations as in Fan and Li (2001, [16]), we express the process as follows. At each time, we denote the training sets and test sets as $T - T^i$ and T^i respectively, where $i = 1, \cdots, k$. Given a number of tuning parameters, we compute the penalized least squares estimator as $\hat{\boldsymbol{\beta}}^{(i)}(\lambda)$ using the training set $T - T^i$ at each tuning parameter. Therefore, the tuning parameter selected by the k-fold cross validation will be the one with the smallest average prediction error, i.e.,

$$\lambda_{CV} = \underset{\lambda}{\operatorname{argmin}} \frac{1}{k} \sum_{i=1}^{k} ||\mathbf{y}_{\mathbf{T^i}} - \mathbf{X}_{\mathbf{T_i}} \hat{\boldsymbol{\beta}}^{(i)}(\lambda)||^2,$$

where $\mathbf{y}_{\mathbf{T^i}}$ and $\mathbf{X}_{\mathbf{T_i}}$ are the corresponding observations in the test set T^i.

It is worth mentioning that since the computation of the cross validation is much more expensive compared with the computation of the other criteria, we will apply the 2-fold cross validation in the dissertation.

Generalized Cross Validation

The generalized cross validation (GCV), firstly introduced by Craven et al. (1978,[13]), is considered as a weighted version of the n-fold cross validation but with significantly reduced computational burden.

In the penalized least squares estimation, given a number of tuning parameters, we can compute the penalized least squares estimator as $\hat{\boldsymbol{\beta}}(\lambda)$ at each tuning parameter. Then the tuning parameter selected by the GCV can be expressed as ([16])

$$\lambda_{GCV} = \underset{\lambda}{\operatorname{argmin}} \frac{||\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}(\lambda)||^2}{n(1 - \sum_{j=1}^{p} I(\hat{\beta}_j(\lambda) \neq 0)/n)^2}.$$

Akaike Information Criterion

The Akaike Information Criterion (AIC), proposed by Akaike (1974, [2]), is one of the most widely used variable selection criteria. Generally, given a statistical model of some data, we can compute the maximized value of the likelihood function, denoted as $\hat{L}$ and the number of the parameters in the model, denoted as $\hat{s}$. Thus the AIC value of this model is defined as

$$AIC = -2 \log \hat{L} + 2\hat{s}.$$

The model chosen from a collection of candidate models is the one with the minimum AIC value.

In the linear regression (1.1), we can compute the likelihood of $(\boldsymbol{\beta}^*, \sigma)$ and derive the AIC value as

$$AIC = n(\log(2\pi) + \log(||\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}||^2) - \log(n)) + 2(n + \hat{s} + 1),$$

where $\hat{\boldsymbol{\beta}}$ is the estimator of $\boldsymbol{\beta}^*$ in the model.

In the penalized least squares estimation, the tuning parameter selected by the AIC is

$$\lambda_{AIC} = \underset{\lambda}{\operatorname{argmin}}(n \log(||\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}(\lambda)||^2) + 2\sum_{j=1}^{p} I(\hat{\beta}_j(\lambda) \neq 0).$$

Bayesian Information Criterion

The Bayesian Information Criterion (BIC), another widely applied information criterion, was proposed by Schwarz (1978,[47]). The BIC is also based on the likelihood function of a model and is related with the AIC. Using the same notations, we can define the BIC value of a model as

$$BIC = -2 \log \hat{L} + \log(p)\hat{s}.$$

Obviously, the BIC penalizes the number of parameters in the model stronger than the AIC.

In the linear regression (1.1), under the assumption that the errors are generated from the normal distribution, the BIC value can be derived as

$$BIC = n(\log(||\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}||^2) - \log(n)) + \log(p)\hat{s}.$$

Similarly, in the penalized least squares estimation, the tuning parameter selected by the BIC is

$$\lambda_{BIC} = \underset{\lambda}{\operatorname{argmin}}(n \log(||\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}(\lambda)||^2) + \log(p)\sum_{j=1}^{p} I(\hat{\beta}_j(\lambda) \neq 0).$$

Extended Bayesian Information Criterion

Since the BIC is too conservative for handling variable selection problems in high dimensional data analysis, Chen and Chen (2008, [11]) proposed the extended Bayesian Information Criterion (EBIC), which emphasizes the model sparsity by adding an additional penalty term on the BIC for the growing number of parameters in the model. In general, the extended BIC can produce a sparser model than the ordinary BIC criterion.

In the penalized least squares estimation, the tuning parameter selected by the EBIC is

$$\lambda_{EBIC} = \underset{\lambda}{\text{argmin}}(n\log(||\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}(\lambda)||^2) + (2\log(p) + \log(n))\sum_{j=1}^{p} I(\hat{\beta}_j(\lambda) \neq 0).$$

Stability Selection

The stability selection, proposed by Meinshausen and Bühhmann (2010, [41]), applies the subsampling technique on the general variable selection methods. It can enhance and improve the selection accuracy of existing penalized likelihood approaches.

Under the framework of the penalized least squares estimation, the algorithm can be simply described as follows. At each tuning parameter λ_k, we draw a random subsampling of size $n/2$ without replacing many times. Then the probability of each variable $\mathbf{x}_j$ being selected in all the subsamples can be computed as $\hat{\Pi}_j^{(\lambda_k)}, j = 1, \cdots, p$. Given a cutoff value π_{thr} with $0 < \pi_{thr} < 1$, the relevant set selected by the stable selection is defined as

$$\hat{S_0}^{(stable)} = \{j : \max_k \hat{\Pi}_j^{(\lambda_k)} \geq \pi_{thr}\}.$$

Here the variables with higher selection probabilities can be kept in the model while those with lower probabilities will be dropped. It is suggested that the cutoff value π_{thr} can take a value in $(0.6, 0.9)$. The algorithm is easy to understand and can be combined with all the penalty functions which produce sparse solutions. Furthermore, the random subsampling method can alleviate the correlation effect among the original data set. Note that the stability selection algorithm does not require selecting the tuning parameter, we identify a variable as relevant if its selection probability at any tuning parameter exceeds the threshold. Hereafter, the algorithm avoids the selection issue of the tuning parameter.

1.3 Outline

The dissertation is organized as follows.

Chapter 1 mainly introduces the background and the motivation of the proposed method in the penalized linear regression. The related work by other statisticians on the penalty functions and the selection criteria are provided in the literature review section.

Chapter 2 presents the major contribution of the dissertation: *selection by partitioning the whole solution paths (SPSP)*. An example is employed to demonstrate the general process of the algorithm. Some further discussions on the penalized likelihood estimation are provided in the following sections.

Chapter 3 develops an original type of scores to rank the importance of the variables based on the results of the SPSP algorithm. We define the scores of the variables by the areas between their solution paths and the boundaries of the partitions. Hereafter, we refer these scores as *the area-out-of-zero-region importance scores (AIS).*

Chapter 4 provides the theoretical results on the selection consistency of the proposed SPSP algorithm. It is shown that the proposed SPSP estimators can achieve the selection consistency as long as the original estimator is estimation consistent. Furthermore, under the irrepresentable condition (Zhao and Yu, 2006, [64]), we prove that the SPSP procedure on the Lasso is selection consistent over the whole solution paths. Moreover, since the irrepresentable condition is difficult to satisfy in practice, we establish the selection consistencies under weaker conditions on both the Lasso and the general penalties at the end of this chapter.

In Chapter 5, we conduct extensive numerical studies to illustrate the advantage of the SPSP procedure and the AIS. Six simulation examples are presented, covering different scenarios in feature selection practices. Moreover, we also carry out a simulation study to illustrate the application of SPSP on Gaussian graphical modeling. The results show the proposed SPSP algorithm significantly outperforms the other methods in general.

In Chapter 6, two real applications (Boston housing data and glioblastomas gene expression data) are presented to further investigate the performance of the proposed method. Compared with other approaches, the models constructed by the proposed SPSP procedure are simpler with relatively smaller model errors.

Chapter 7 summarizes the main results in the dissertation. Some prospective future research and discussion are provided in the chapter as well.

CHAPTER 2 SELECTION BY PARTITIONING THE SOLUTION PATHS

In this chapter, we propose an approach which utilizes the whole solution paths to select the informative features in the model. Firstly, we develop a partition rule to divide the variables into two groups: relevant and irrelevant, at each tuning parameter. Based on the partitioning rule, we propose the main algorithm of the dissertation: *the selection by partitioning the whole solution paths (SPSP) algorithm*. The algorithm allows us to improve the selection accuracy by efficiently combining all the information across the whole solution paths, especially in cases when strong correlations among the variables are presented. Furthermore, we extend the SPSP procedure to the penalized likelihood estimation.

2.1 The SPSP Algorithm

2.1.1 Notations and the Partitioning Rule

Considering the penalized least squares problem:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{n}||\mathbf{y} - \boldsymbol{X}\boldsymbol{\beta}||^2 + \lambda \sum_{j=1}^{p} J(|\beta_j|), \tag{2.1}$$

where $\mathbf{y} = (y_1, \cdots, y_n)^T$ is the response and $\mathbf{X} = (\mathbf{x_1}, \cdots, \mathbf{x_p})$ is the design matrix in model (1.1). Here we also assume that $\mathbf{x}_j, j = 1, \cdots, p$ are standardized and the response $\mathbf{y}$ is centered. Further denote the vector of the true regression coefficients in model (1.1) as

$$\boldsymbol{\beta}^* = (\beta_1^*, \cdots, \beta_p^*),$$

such that $Ey = \mathbf{X}\boldsymbol{\beta}^*$. Suppose the index set for the true relevant (nonzero) variables is $S = \{j : \beta_j^* \neq 0\}$ with $s = |S|$, and the index set for irrelevant variables is $S^c = \{j : \beta_j^* = 0\}$. The goal in the variable selection is to correctly recover this sparsity pattern from the noisy observations in the model, and correctly estimate S.

Once we specify the penalty function $J(\cdot)$ in (2.1), a grid of the tuning parameters is required to compute the solution paths. Typically, we would pick the grid to be equi-distant on the log scale as follows:

$$\lambda_{\min} = \lambda_1 < \cdots < \lambda_K = \lambda_{\max},$$

where $\lambda_{\min} = 1/n$, λ_{max} is the smallest λ yielding $\hat{\boldsymbol{\beta}} = 0$. Hence we have

$$\lambda_{k+1} = \lambda_k \exp\left(\frac{\log(\lambda_{\max}) - \log(\lambda_{\min})}{K-1}\right), k = 1, \cdots, K-1.$$

Bühlmann and Van De Geer (2011,[8]), Shen et al. (2012, [48]) also suggested the same way to build the grid of the tuning parameters. Note that since the solution paths are usually continuous with respect to the tuning parameter, they vary little for the choice of the grid as long as enough tuning parameters are selected. In practice, we can pick almost $K = 100$ grid points, as suggested in Shen et al. (2012, [48]) and Sun et al. (2012, [51]).

For each λ_k, we obtain a vector of the penalized least squares estimators as

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{k}} = (\hat{\beta}_{k,1}, \cdots, \hat{\beta}_{k,p})^T.$$

A variable is more likely to be identified as relevant if its estimator is farther away from 0, regardless of the sign of the estimator. Therefore we take the absolute values of the estimators as

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{k}}^{(abs)} = (|\hat{\beta}_{k,1}|, \cdots, |\hat{\beta}_{k,p}|)^T.$$

In general, variables with a larger $|\hat{\beta}_{k,p}|$ are more likely to be important. Therefore, we are interested in finding a proper cutoff point $T_k = T(\lambda_k)$, such that the estimated relevant set $\hat{S}_k$ and irrelevant set $\hat{S}_k^c$ at $\lambda = \lambda_k$ are derived as

$$\hat{S}_k = \{j : |\hat{\beta}_{k,j}| > T_k\} \tag{2.2}$$

and

$$\hat{S}_k^c = \{j : |\hat{\beta}_{k,j}| \leq T_k\}. \tag{2.3}$$

To obtain $(T_k, \hat{S}_k, \hat{S}_k^c)$ for each λ_k, we sort the absolute values $|\hat{\beta}_{k,1}|, \cdots, |\hat{\beta}_{k,p}|$ in ascending order to obtain

$$\hat{\beta}_{k,(1)}^{(abs)} \leq \cdots \leq \hat{\beta}_{k,(p)}^{(abs)},$$

where $\hat{\beta}_{k,(j)}^{(abs)}$ is the jth order statistics of $|\hat{\beta}_{k,1}|, \cdots, |\hat{\beta}_{k,p}|$. Then we define the adjacent distances between these ordered values as

$$D_{k,j} = \hat{\beta}_{k,(j)}^{(abs)} - \hat{\beta}_{k,(j-1)}^{(abs)}, j = 1, \cdots, p.$$

Note that $D_{k,1}$ is the adjacent distance between $\hat{\beta}_{k,(1)}^{(abs)}$ and 0 as we define $\hat{\beta}_{k,(0)}^{(abs)} = 0$ for convenience.

Let $\hat{s}_k = |\hat{S}_k|$ be the number of variables in the estimated relevant set $\hat{S}_k$, then there are $p - \hat{s}_k$ variables in the estimated irrelevant set $\hat{S}_k^c$. Hereafter, by (2.2) and (2.3), the selected variables in $\hat{S}_k$ are those variables whose estimated coefficients correspond to the $\hat{s}_k$ largest values:

$$\hat{\beta}_{k,(p-\hat{s}_k+1)}^{(abs)}, \cdots, \hat{\beta}_{k,(p)}^{(abs)}$$

while the variables in $\hat{S}_k^c$ are those whose estimated coefficients correspond to the smallest $p - \hat{s}_k$ ordered values:

$$\hat{\beta}_{k,(1)}^{(abs)}, \cdots, \hat{\beta}_{k,(p-\hat{s}_k)}^{(abs)}.$$

We simply define the gap between $\hat{S}_k$ and $\hat{S}_k^c$ as the adjacent distance between $\hat{\beta}_{k,(p-\hat{s}_k)}^{(abs)}$ and $\hat{\beta}_{k,(p-\hat{s}_k+1)}^{(abs)}$, i.e.,

$$D(\hat{S}_k, \hat{S}_k^c) = D_{k,p-\hat{s}_k+1} = \hat{\beta}_{k,(p-\hat{s}_k+1)}^{(abs)} - \hat{\beta}_{k,(p-\hat{s}_k)}^{(abs)}.$$

Note that it suggests given the number of the variables in $\hat{S}_k$, one can compute the adjacent distance which separates $\hat{S}_k$ and $\hat{S}_k^c$ as $D_{k,p-\hat{s}_k+1}$.

In principle, $D(\hat{S}_k, \hat{S}_k^c)$, the gap between $\hat{S}_k$ and $\hat{S}_k^c$ should be sufficiently large to separate the irrelevant features from the important ones. We consider $D(\hat{S}_k, \hat{S}_k^c)$ large enough if it meets the following two criteria,

$$\frac{D_{\max}(\hat{S}_k)}{D(\hat{S}_k, \hat{S}_k^c)} < C, \tag{2.4}$$

$$\frac{D(\hat{S}_k, \hat{S}_k^c)}{D_{\max}(\hat{S}_k^c)} > C, \tag{2.5}$$

where $D_{\max}(\hat{S}_k) = \max\{D_{k,j} : j > p - \hat{s}_k + 1\}$ is the largest adjacent distance in $\hat{S}_k$, $D_{\max}(\hat{S}_k^c) = \max\{D_{k,j} : j < p - \hat{s}_k + 1\}$ is the largest adjacent distance in $\hat{S}_k^c$ and C is a certain constant. The criterion (2.4) ensures that the gap between the relevant set and irrelevant set should have the same order as the distances between the estimated nonzero coefficients, while (2.5) guarantees that $D(\hat{S}_k, \hat{S}_k^c)$ has a higher order than the distances between the estimators of the zero coefficients. The constant C is used to control the differences of the magnitudes between the estimators of the zero coefficients and those of the nonzero coefficients. The principles (2.4) and (2.5) here are equivalent to saying that the order of the estimators for the nonzero coefficients should be higher than those of the zero coefficients. Instead of comparing every pair of the estimators, we just use adjacent distances for simplicity of the calculation. Therefore, finding the proper T_k now transforms to finding an adjacent distance that is large enough—satisfies (2.4) and (2.5)—to be the gap between the estimators for the zero coefficients and those for the nonzero ones.

In order to introduce our proposed algorithm for partitioning the solution paths, we further define the largest adjacent distance under where $D_{\max}(\hat{S}_k^c)$ happens in $\hat{S}_k^c$ as

$$D_{\max 2}(\hat{S}_k^c) = \max\{D_{k,j} : j < j', D_{k,j'} = D_{\max}(\hat{S}_k^c)\}.$$

Then following the aforementioned principles, we develop the algorithm for partitioning the solution paths as follows.

***Selection by Partitioning the Solution Paths (SPSP)* Algorithm**

1 Set the initial values as $T_0 = \infty$, $\hat{S}_0 = \emptyset$, $\hat{S}_0^c = \{1, \ldots, p\}$, and proceed to λ_1.

2 At each λ_k, we estimate $T_k, \hat{S}_k, \hat{S}_k^c$ from $T_{k-1}, \hat{S}_{k-1}, \hat{S}_{k-1}^c$ and $\hat{\boldsymbol{\beta}}_{\boldsymbol{k}}^{(abs)}$.

2.1 Update $T_k = \max_{j \in \hat{S}_{k-1}^c} |\hat{\beta}_{k,j}|$, $\hat{S}_k = \{j : |\hat{\beta}_{k,j}| > T_k\}$, $\hat{S}_k^c = \{j : |\hat{\beta}_{k,j}| \leq T_k\}$;

2.2 Calculate $D_{k,1}, \ldots, D_{k,p}$. Further obtain $D_{\max}(\hat{S}_k^c)$, $D_{\max 2}(\hat{S}_k^c)$ and $D(\hat{S}_k, \hat{S}_k^c)$.

2.3 If $D(\hat{S}_k, \hat{S}_k^c) < C \times D_{\max}(\hat{S}_k^c)$ and $D_{\max}(\hat{S}_k^c) > C \times D_{\max 2}(\hat{S}_k^c)$, we update

$$T_k = \hat{\beta}_{k,(j'-1)}^{(abs)}, \hat{S}_k = \{j : |\hat{\beta}_{k,j}| > T_k\}, \hat{S}_k^c = \{j : |\hat{\beta}_{k,j}| \leq T_k\}.$$

Otherwise $T_k, \hat{S}_k, \hat{S}_k^c$ remain unchanged as in Step 2.1.

3 Proceed to λ_{k+1} and repeat Step 2 until $k = K$.

4 Identify the union of all $\hat{S}_k$ as the index set for our selected relevant variables, i.e., $\hat{S} = \bigcup_{k=1}^{K} \hat{S}_k$.

At each λ_k, we find in Step 2 the cutoff point T_k, the location of the gap that distinguishes the relevant and irrelevant variables, based on the results from λ_{k-1}. This not only simplifies the computation process, but also makes the boundary line $T_k = T(\lambda_k)$ relatively more smooth to avoid unstable selection results. Specifically, in Step 2.1, we first use the largest estimated coefficients among those identified as "zero-coefficients" for λ_{k-1} as the current boundary. This could take care of the case where some coefficients in $\hat{S}_{k-1}$ becomes small and enters into the zero region at λ_k. At Step 2.2 and Step 2.3, we decided on whether any adjacent distances within $\hat{S}_k^c$ is large enough to be considered as the new gap between the zero and nonzero coefficients. This manages the scenario that there are too few variables in $\hat{S}_k$ so that a "large" gap still exists.

For λ_1, we use the initial values set in Step 1, where all the variables are considered "irrelevant", which means we are conservative in identifying the relevant variables at the start of this process. This is because we implement an aggressive selection strategy at Step 4 to use the union of all $\hat{S}_k$'s

as our estimated index set for the relevant variables, allowing us to minimize the false positive rate at each λ_k. Another reason is that the estimation at the small λ_k's are usually unstable, because the design matrix corresponding to nonzero coefficients is usually ill-conditioned. Therefore it is better to select fewer relevant variables, rather than taking the high risks of committing false positive errors.

Here the choice of the constant C for the partitioning rule is data-adaptive. In practice, we first obtain the estimator with a small value of λ, then take the absolute value and compute the adjacent distances of the sorted values. We then choose the constant C as the ratio of the maximal adjacent distance to the second maximal adjacent distance. Simulation studies confirm that the strategy is practically effective. In fact, both the simulations and our theoretical results show that our final results are not sensitive to the choice of C.

Once we identify the index set of all the relevant variables $\hat{S}$, we estimate the regression parameters $\hat{\beta}_{\hat{S}}$ a model that only includes the features that has been selected,

$$\mathbf{y} = \mathbf{X}_{\hat{S}}\boldsymbol{\beta}_{\hat{S}} + \boldsymbol{\varepsilon},$$

where $\mathbf{X}_{\hat{S}} = (\mathbf{x_j})_{j\in\hat{S}}$, $\boldsymbol{\beta}_{\hat{S}} = (\beta_j)^T_{j\in\hat{S}}$. In most cases, the number of features in $\hat{S}$ is smaller than the sample size, we just use the least squares estimator as $\hat{\beta}_{\hat{S}}$. If the number of selected variables are larger than the sample size, we could use a ridge regression with a small shrinkage factor.

To gain more insights for the approach, we demonstrate the general process of the SPSP algorithm by the following example.

Suppose Figure 2.1 is the plot of the solution paths of the penalized least squares estimators over 4 tuning parameters. There are also 4 lines in the plot, where each one represents a variable in the model. To better describe the process, we name these variables as "1", "2", "3" and "4" based on the β values from top to bottom at λ_1.

In the SPSP approach, the threshold value, the estimated relevant set and irrelevant set are ini-

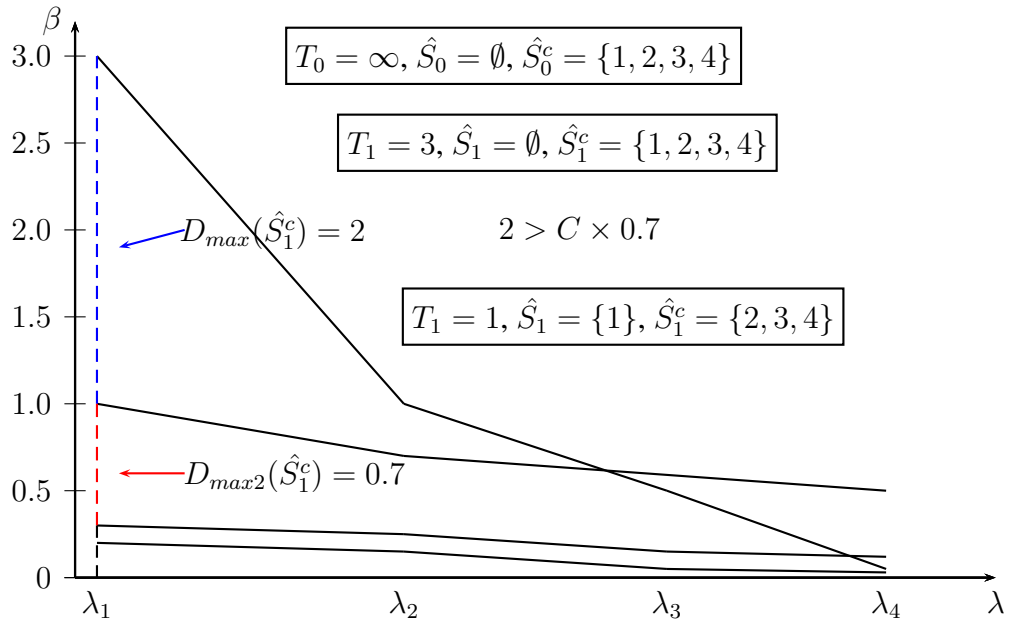

Figure 2.1: Demonstration of the SPSP approach on a simple example: Partitions at λ_1.

tialized as

$$T_0 = \infty, \hat{S}_0 = \emptyset, \hat{S}_0^c = \{1,2,3,4\}.$$

At the first tuning parameter λ_1, we firstly update the threshold value as the largest value of the irrelevant variables in the previous step. Since the initial estimated irrelevant set contains all the variables, we have

$$T_1 = 3, \hat{S}_1 = \emptyset, \hat{S}_1^c = \{1,2,3,4\}.$$

We recommend the constant C can be chosen as the ratio of the largest adjacent distance and the second adjacent distance at a small tuning parameter ($\lambda = 0.01$). For instance, suppose we pick $C = 2$ for this example. Note that currently the estimated relevant set is an empty set, therefore the adjacent distance in $\hat{S}_1$ is defined to be 0. According to the partitioning rule, we update the adjacent

distance 2 as the new gap between the relevant set and irrelevant set, i.e.,

$$T_1 = 1, \hat{S}_1 = \{1\}, \hat{S}_1^c = \{2, 3, 4\}.$$

Similarly, we follow the same steps for the remaining tuning parameters. Figure (2.2), (2.3) demonstrate the partitioning processes at the tuning parameters λ_2 and λ_3. In particular, at λ_2, the maximal adjacent distance 0.45 in the irrelevant set is large enough to satisfy the partitioning rule

$$0.3 < C \times 0.45 \text{ and } 0.45 > C \times 0.1.$$

Therefore we update the adjacent distance 0.45 as the new gap, then we have

$$T_2 = 0.25, \hat{S}_2 = \{1, 2\}, \hat{S}_2^c = \{3, 4\}.$$

At λ_3, the maximal adjacent distance 0.10 in the irrelevant set does not satisfy the second partitioning rule, where $0.10 \leq C \times 0.05$, thus the estimated relevant and irrelevant sets remain unchanged as

$$T_3 = 0.15, \hat{S}_3 = \{1, 2\}, \hat{S}_3^c = \{3, 4\}.$$

Obviously, we can easily perform the SPSP algorithm for all the remaining tuning parameters and finally union all the relevant sets as our final result. In this example, based on the first 3 tuning parameters, the final relevant set is

$$\hat{S} = \hat{S}_1 \bigcup \hat{S}_2 \bigcup \hat{S}_3 = \{1, 2\},$$

which is reasonable based on the solution paths of these variables since the estimators of the variables "1" and "2" are remarkably larger than the remaining two variables.

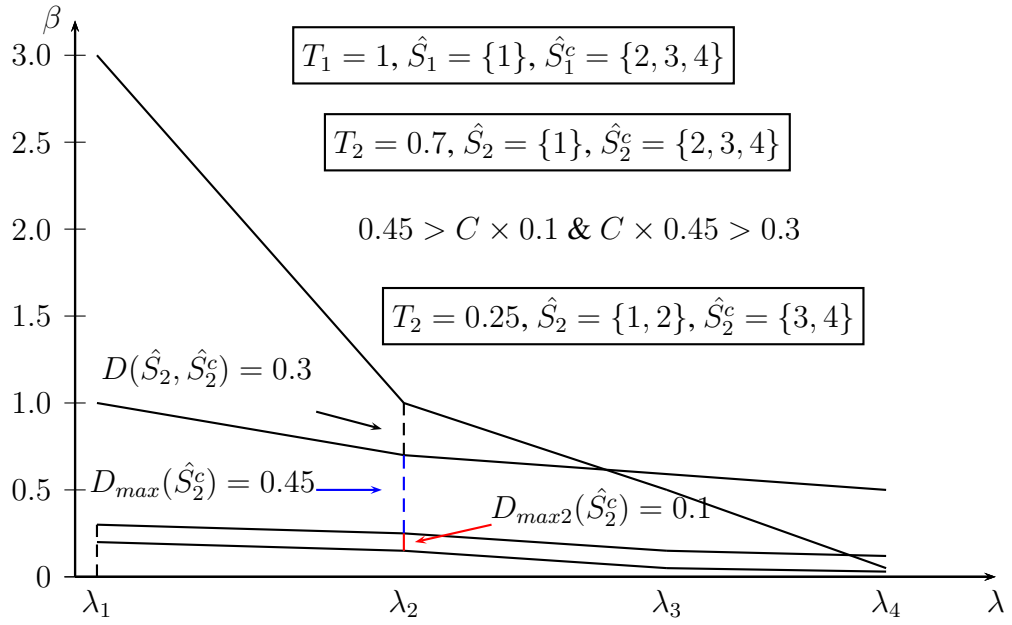

Figure 2.2: Demonstration of the SPSP approach on a simple example: Partitions at λ_2.

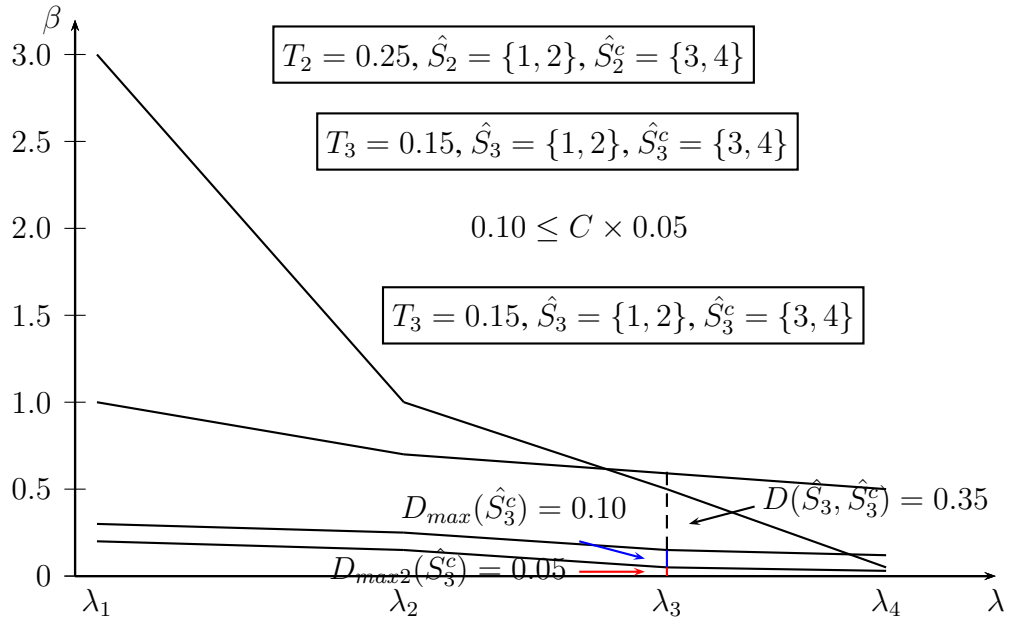

Figure 2.3: Demonstration of the SPSP approach on a simple example: Partitions at λ_3.

2.1.2 The SPSP Algorithm on the Ridge

One advantage of the proposed SPSP procedure is that it can not only be applied to the penalties like the Lasso, the adaptive Lasso, the SCAD, the MCP, but also it can be applied for the penalties which cannot produce sparse solutions, such as the ridge penalty. As a result, the SPSP algorithm on the ridge can greatly reduce the computation complexity for feature selection problems since strictly convex penalties like ridge are easier to solve.

Particularly, the ridge estimator in the linear regression model (1.1), which refers to the L_2 penalized least squares estimator, is defined as

$$\hat{\boldsymbol{\beta}}^{(ridge)} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{n}||\mathbf{y} - \boldsymbol{X}\boldsymbol{\beta}||_2^2 + \lambda||\boldsymbol{\beta}||_2^2,$$

where $||\boldsymbol{\beta}||_2^2 = \sum_{j=1}^{p} \beta_j^2$ is the L_2 norm of the parameters.

Compared with other penalties, the advantage of the ridge estimator is that it has a closed solution:

$$\hat{\boldsymbol{\beta}}^{(ridge)} = (\mathbf{X}^{\mathbf{T}}\mathbf{X} + n\lambda\mathbf{I})^{-1}\mathbf{X}^{\mathbf{T}}\mathbf{y},$$

which simplifies the computation compared with the other penalized least squares estimators. Nonetheless, it is well known the ridge estimator can not perform the variable selection since it does not automatically shrink the estimates of the regression coefficients to zero. To remedy the problem, we can apply the proposed SPSP algorithm on the ridge estimator for the variable selection. The simulation studies in Chapter 5 show the selection accuracy is quite satisfactory.

Additionally, for some special problems, the SPSP algorithm on the ridge has a remarkably good performance. Take Example 5 in Wang et al. (2011, [57]) for instance, there are 120 variables and 50 observations in the example. The first 60 coefficients are generated from $N(3, 0.5)$ while the remaining 60 coefficients are zero. The design matrix $\mathbf{X}$ is generated from a multivariate normal

distribution with zero mean and the covariance matrix

$$\begin{pmatrix} \Sigma_0 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Sigma_0 & 0.2\mathbf{J} & \mathbf{0} \\ \mathbf{0} & 0.2\mathbf{J} & \Sigma_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \Sigma_0 \end{pmatrix},$$

where Σ_0 is a 30×30 matrix with diagonal elements 1 and off-diagonal elements 0.7, $\mathbf{J}$ is a 30×30 matrix with all elements 1 and the error $\varepsilon_i \sim N(0,1), i = 1, \cdots, n$.

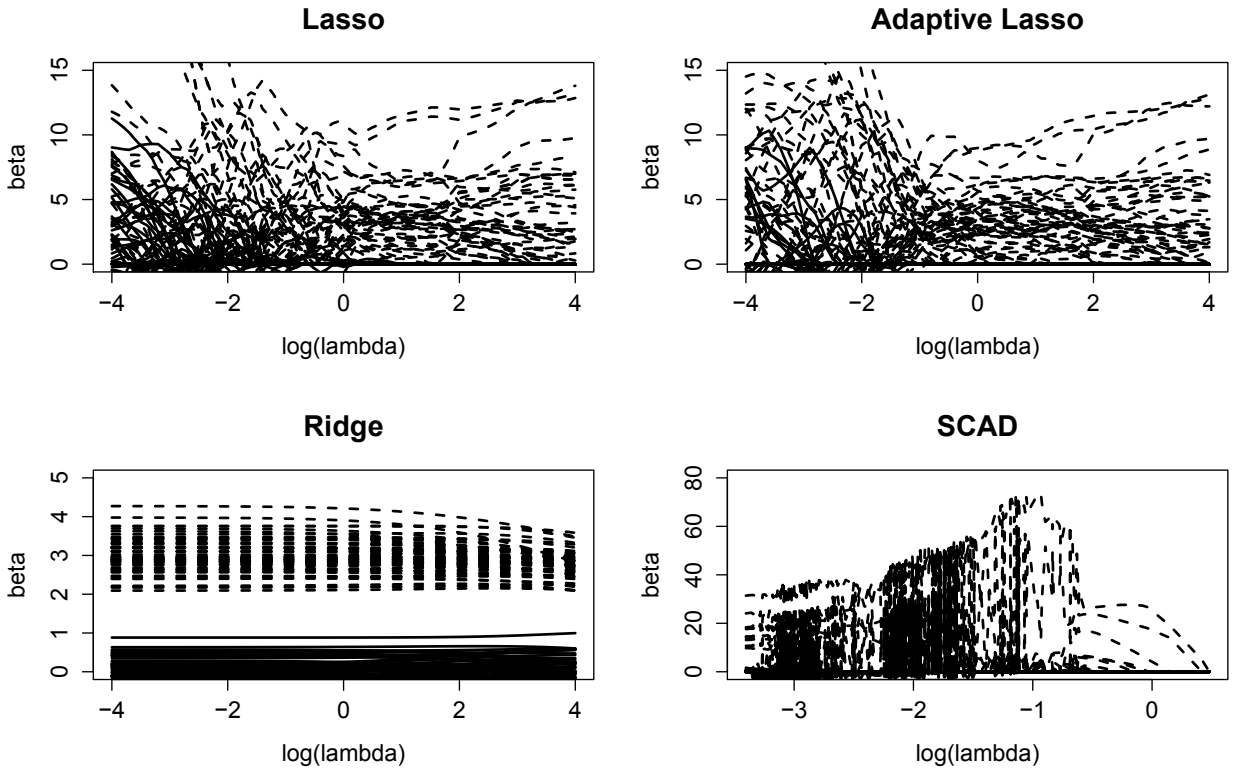

Figure 2.4: The solution paths for the Lasso, the adaptive Lasso, the ridge and the SCAD of a simulated example.

Figure 2.4 shows the solution paths of the Lasso, the adaptive Lasso, the ridge and the SCAD for the same simulated example. The dashed lines represent those 60 nonzero variables and the black

lines represent the remaining 60 zero variables. We implement the Lasso and the adaptive Lasso by the **R** package *lassoshooting* and the SCAD by the **R** package *plus*. It is observed that for this high dimensional problem, the solution paths of the Lasso, the adaptive Lasso and the SCAD are chaotic, which hardly distinguish the relevant coefficients from the irrelevant ones. Whereas, the solution paths of the ridge regression can perfectly divide the relevant and irrelevant coefficients into two clusters. The gap between the relevant set and the irrelevant set at each tuning parameter is so large that one can identify the nonzero variables intuitively. Obviously, the SPSP algorithm on the ridge can be perfectly applied on this example. For more details, we evaluate the average performance of the SPSP algorithm on these penalties over 100 replicates. It turns out the solution paths of the ridge are similar and the SPSP approach on the ridge has the best performance.

In general, the SPSP algorithm on the ridge can have wide applications in practice due to its computational feasibility. We now have the power to replace the L_1 penalties and all the non-convex penalties with a strictly convex penalty function like the L_2 penalties. More simulation studies can be found in Chapter 5 of the dissertation.

2.2 The SPSP Algorithm in the Penalized Likelihood Estimation

The penalized least squares estimation can be easily extended to handle feature selection problems in many other models such as graphical modeling, generalized linear models, Cox's proportional hazards models. In these models, we usually apply the penalized likelihood approach, which obtains a sparse estimate by solving an objective function consisting of likelihood and a penalty function. It is well known that the penalized least squares estimation in the general linear regression model is equivalent to a transformation of the penalized likelihood estimation. As a result, we can apply the SPSP algorithm on penalized likelihood estimators with a similar fashion.

Let $f(\mathbf{V}, \boldsymbol{\beta})$ be the likelihood for a random vector $\mathbf{V} = (V_1, \cdots, V_n)$, then the penalized likelihood estimator can be obtained by maximizing the following function:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmax}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \log f(V_i, \boldsymbol{\beta}) - \lambda \sum_{j=1}^{p} J(|\beta_j|) \right\},$$

where $J(\cdot)$ is a penalty function, and $\lambda > 0$ is the tuning parameter. Obviously, the problem has an equivalent form as

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ -\frac{1}{n} \sum_{i=1}^{n} \log f(V_i, \boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} J(|\beta_j|) \right\},$$

which is similar to the penalized least squares estimation in (1.2).

Fan and Li (2004, [20]) established the theoretical results for the non-concave penalized likelihood estimators under the infinite parameter setting. It is shown that under several regularity conditions, the penalized likelihood estimators enjoy the oracle properties that the estimation is as accurate as if we knew nonzero coefficients and applying MLE on the submodes which contains the true nonzero coefficients only. See [16] ,[20] for more asymptotic results of the penalized likelihood estimators.

The computation of the penalized likelihood estimation often requires some numerical algorithms. The coordinate descent algorithm along with the shooting algorithm can be applied for the computation of convex penalties (See [26]). For the computation of nonconcave penalties, Fan and Li (2001, [16]) proposed an effective local quadratic approximation (LQA) algorithm.

Using the same notations, we select a sequence of the tuning parameters $\lambda_1 < \cdots < \lambda_K$. At each tuning parameter, the penalized likelihood estimator is denoted as $\hat{\boldsymbol{\beta}}_{\boldsymbol{k}} = (\hat{\beta}_{k,1}, \cdots, \hat{\beta}_{k,p})^T$. Therefore, the same SPSP approach can be applied on these estimators.

Note that the choice of the constant C in the SPSP algorithm for the penalized likelihood estimation is similar as mentioned above. We can obtain an initial estimator as the MLE firstly:

$$\hat{\boldsymbol{\beta}}^{(MLE)} = \underset{\boldsymbol{\beta}}{\operatorname{argmax}} \frac{1}{n} \sum_{i=1}^{n} \log f(V_i, \boldsymbol{\beta}),$$

and then compute the adjacent distances of the sorted absolute values. The constant C can also be chosen as the ratio of the maximal adjacent distance to the second maximal adjacent distance in practice. In case MLE does not exist due to the high dimensionality of $\boldsymbol{\beta}$, we can apply the penalized

likelihood estimator with a small shrinkage parameter as the initial estimator.

After we obtain the relevant index set $\hat{S}$, we perform the estimation as

$$\hat{\boldsymbol{\beta}}^{(SPSP)} = \underset{\boldsymbol{\beta}}{\operatorname{argmax}} \frac{1}{n} \sum_{i=1}^{n} \log f(V_i, \boldsymbol{\beta}_{\hat{S},\hat{S}^c}),$$

where $\boldsymbol{\beta}_{\hat{S},\hat{S}^c} = (\boldsymbol{\beta}_{\hat{S}}, \mathbf{0}_{\hat{S}^c})^T$.

2.2.1 Graphical Modeling

Graphical models use graphs to represent the conditional dependencies between random variables. In Gaussian graphical models, suppose there are n multivariate normal observations of dimension d, with mean μ and covariance matrix Σ. We are interested in inferring the conditional dependencies between these d variables $(\mathbf{x}_1, \cdots, \mathbf{x}_d)$. Denote the precision matrix as the inverse of the covariance matrix $\Theta = \Sigma^{-1}$. Note that two variables $\mathbf{x}_j$ and $\mathbf{x}_k$ are conditionally dependent given all the other variables if and only if $\Theta_{jk} \neq 0$. Therefore the problem is equivalent to the estimation of the precision matrix Θ. To discover the sparsity pattern of the precision matrix, the graphical Lasso, proposed by Friedman et al. (2008, [25]), has been a popular approach in the graphical models. Generally, the graphical Lasso is a penalized likelihood approach which can be defined as follows,

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmin}} \left\{ -\log(det(\Theta)) + tr(S\Theta) + \lambda \sum_{j<k} |\Theta_{jk}| \right\}$$

over all the non-negative definite matrices, where $det(A)$ denotes the determinant of A, $tr(A)$ denotes the trace of A and $\lambda > 0$ is the tuning parameter. A simple and fast algorithm to compute the graphical Lasso is also proposed in Friedman et al. (2008, [25]).

If $\Theta_{jk} \neq 0$ which indicates the conditional dependency between $\mathbf{x}_j$ and $\mathbf{x}_k$, we draw an edge between nodes j and k in a corresponding graph (Lauritzen, 1996, [36]). Therefore, the number of the parameters is $p = \binom{d}{2}$ and the SPSP algorithm can be applied to select the nonzero values in this Gaussian graphical model.

2.2.2 Generalized Linear Models

In the generalized linear models, we simply assume that conditioning on $\mathbf{x_{(i)}}$, the density of y_i is $f_{\boldsymbol{\beta}}(y_i|g^{-1}(\mathbf{x_{(i)}}))$, where g^{-1} is a real-valued, known inverse link function. For instance, in logistic regression, the conditional distribution of $y_i|\mathbf{x_{(i)}}$ is $Binomial(1, \pi(\mathbf{x_{(i)}}))$ and the link function is $g(\pi) = \log\left(\frac{\pi}{1-\pi}\right)$. That is,

$$\log\left(\frac{\pi(\mathbf{x_{(i)}})}{1-\pi(\mathbf{x_{(i)}})}\right) = \mathbf{x_{(i)}}^T\boldsymbol{\beta}.$$

Generally, we can compute the penalized likelihood estimators in generalized linear models as

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ -\frac{1}{n}\sum_{i=1}^{n} \log f_{\boldsymbol{\beta}}(y_i|g^{-1}(\mathbf{x_{(i)}})) + \lambda \sum_{j=1}^{p} J(|\beta_j|) \right\}.$$

Several fast algorithms have been developed for the estimation of the generalized linear models with convex penalties, such as the path following algorithm (Park and Hastie, 2007, [43]) and the cyclical coordinate descent algorithm (Friedman et al., 2010, [24]). We can commonly pick the Lasso and the adaptive Lasso as the penalty functions and then apply the proposed SPSP algorithm for the variable selection.

2.2.3 Proportional Hazards Models

Tibshirani (1997, [53]) firstly proposed the Lasso for variable selection in the Cox's proportional hazards model. Fan and Li (2002, [17]) then proposed the penalized partial likelihood approach for the Cox's proportional hazards models. Using the same notations in their paper, we denote T, C and $\mathbf{x}$ be the survival time, the censoring time and the associated variables respectively. Additionally, let $Z = \min(T, C)$ be the observed time and $\delta = I(T \leq C)$ be the censoring indicator. Assume the observed data $(\mathbf{x}_i, Z_i, \delta_i), i = 1 \cdots, n$ is an independently and identically distributed random sample from a certain population. The hazard function of T given $\mathbf{x}$ in the model is $h(t|\mathbf{x}) = h_0(t)exp(\mathbf{x}^T\boldsymbol{\beta})$, where $h_0(t)$ is an baseline hazard function and $\boldsymbol{\beta}$ is the vector of the parameters in

the model.

More notations are needed for the likelihood in the model. The observed failure times can be sorted as $t_1^0 < \cdots < t_N^0$ and the variables with the corresponding N failures are $\mathbf{x}_{(1)}, \cdots, \mathbf{x}_{(N)}$. Let R_j be the risk set right before the time t_j^0, that is $R_j = \{i : Z_i \geq t_j^0\}$. Then the penalized partial likelihood estimator can be defined as

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ -\frac{1}{n} \sum_{i=1}^{N} \left[\mathbf{x}_{(j)}^T \boldsymbol{\beta} - \log \sum_{i \in R_j} \exp(\mathbf{x}_i^T \boldsymbol{\beta}) \right] + \lambda \sum_{j=1}^{p} J(|\beta_j|) \right\}.$$

More details can be found in Fan and Li (2002, [17]). Note that we have to use the partial likelihood in the model due to the censoring issue in the survival data. Based on the partial penalized likelihood estimation, we can easily perform the variable selection by the proposed SPSP algorithm. Since the estimators are computed from the incomplete likelihood, one can expect the selection based on the whole solution paths should be more stable than the selection by a single tuning parameter.

CHAPTER 3 AREA-OUT-OF-ZERO-REGION IMPORTANCE SCORES

The SPSP procedure automatically identifies the important variables. However, it is possible that one may want to keep the flexibility for tuning the model complexities in actual applications. Traditional approaches such as forward/backward selection ranks the variables by their order of entering/leaving the model. While typical lasso type of approaches rank the variables by the order of the tuning parameters at which the estimates of their coefficients are shrunk to zero. In other words, if the estimate for the coefficient of variable A hits zero at a smaller value of the tuning parameter compared to variable B, then it is impossible to include A in the model without B, even if A is more important than B.

We propose to assess the importance of each variable using their solution paths. Here we develop an original type of scores, noted as area-out-of-zero-region importance scores (AIS), to rank the importance of the features. The scores we define also incorporate the information of the whole solution paths, and is based on the SPSP procedure described in Chapter 2.

Specifically, we rank the importance of the variables by the areas between their solution paths and the boundary of the partitions, which can be computed as the integral of the difference of the variable curve and the boundary, i.e.

$$A(j) = \int_{\log(\lambda_1)}^{\log(\lambda_K)} [f_j(\log(\lambda)) - c(\log(\lambda))] d\log(\lambda),$$

where $f_j(\log(\lambda))$ represents the curve of solution path for β_j, and $c(\log(\lambda))$ represents the boundary of the partitions. Here we use the log transformation of the tuning parameter as the independent parameter because we commonly pick the tuning parameters equidistant on the log scale. In this sense, the variable with the solution path which mostly stays in nonzero region is more likely to be chosen in the model.

We use the trapezoid rule to approximate the integral. Since we choose λ in the log scale, let

$$\Delta_k = \log(\lambda_{k+1}) - \log(\lambda_k), k = 1, \cdots, K-1,$$

and we summarize how to calculate $A(j)$ and rank the importance of the variables as follows.

***Area-out-zero-region (AIS)* Algorithm**

1. Input the absolute values of the estimators $\hat{\boldsymbol{\beta}}_{\boldsymbol{k}}^{(abs)}$, the cutoff points $T_k, k = 1, \cdots, K$, the final relevant set $\hat{S}$ obtained from the SPSP algorithm.

2. For each $j = 1, \cdots, p$, compute

$$A(j) = \frac{1}{2} \sum_{k=1}^{K-1} [(|\hat{\beta}_{k,j}| - T_k) + (|\hat{\beta}_{k+1,j}| - T_{k+1})](\Delta_k).$$

3. Sort the index set $\hat{S}$ by $A(j), j \in \hat{S}$ decreasingly and also the index set $\hat{S}^c$ by $A(j), j \in \hat{S}^c$ decreasingly.

The general idea of the aforementioned algorithm is that if a variable is more important, the estimate of its coefficients tends to be larger. The AIS is a measurement of the average magnitude of these estimates across the whole solution paths. Compared to the other existing approaches, the AIS directly targets on the importance of the variables, minimizing the possibilities of less important variables coming into the model before more important ones. Therefore, models yielded from AIS tend to have a lower false positive rate, as evidenced by our simulation studies.

CHAPTER 4 LARGE SAMPLE THEORIES

This chapter derives the large sample theories of the proposed SPSP estimator on the general penalized least squares estimators. Basically, there are two types of consistencies in the variable selection. An estimator $\hat{\boldsymbol{\beta}}$ is called estimation consistent if

$$P(\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}^*) \to 1 \text{ as } n \to \infty,$$

and called selection consistent if

$$P(\hat{S}(\hat{\boldsymbol{\beta}}) = S) \to 1 \text{ as } n \to \infty,$$

where $\hat{S}(\hat{\boldsymbol{\beta}}) = \{j : \hat{\boldsymbol{\beta}}_j \neq 0\}$.

For most penalties, the estimators can satisfy either selection consistency or estimation consistency or even both when the tuning parameter λ_n is appropriately chosen. In particular, the Lasso penalty is selection consistent under some conditions when $\sqrt{n}\lambda_n \to \infty$ and $\lambda_n \to 0$ ([64]); the ridge penalty is estimation consistent when $\lambda_n \to 0$; the adaptive Lasso penalty is selection consistent when $\sqrt{n}\lambda_n \to \infty$ and $\sqrt{n}\lambda \to 0$ ([65]); the SCAD penalty is selection consistent when $\sqrt{n}\lambda_n \to \infty$ and $\lambda_n \to 0$ ([16]).

As mentioned in Chapter 2, the estimation of the SPSP algorithm is the OLS estimation based on the selected relevant variables while the OLS estimators are estimation consistent under the general assumptions in (1.1). Hence, the estimation consistency of the SPSP estimator can be obtained directly if the selection consistency is satisfied. Hereafter, we will focus on the selection consistency of the SPSP estimator in the chapter.

We firstly show the SPSP estimator can possess the selection consistency built on the solution paths over the interval where the original estimator is either selection consistent or estimation consistent. In the sense, although the ridge estimator cannot produce sparse solutions, the SPSP estimator

on the ridge can still be selection consistent. Then we establish the selection consistency of the SPSP estimator on the Lasso over the whole solution paths. We mainly apply the irrepresentable condition, proposed by Zhao and Yu (2006, [64]), to control the estimation behavior of the Lasso and then discuss the consistency under some weaker conditions. Finally we extend the consistency properties of the SPSP estimator to the general penalized least squares estimation.

4.1 Selection Consistency

The SPSP algorithm bases on the idea that the gap between the estimated relevant and irrelevant coefficients should be significantly large, which actually means that the distances of the relevant estimators are one or several orders of magnitude larger than the distances of the irrelevant estimators. According to this partitioning rule, the selection consistency of the proposed SPSP estimator can be established as follows.

Let $\hat{S}(\lambda_n)$ be the estimated relevant set at the tuning parameter λ_n, then the relevant set of the SPSP algorithm built on the solution paths over the interval I should be

$$\hat{S} = \bigcup_{\lambda_n \in I} \hat{S}(\lambda_n).$$

We establish the following property as our first result for the SPSP estimator.

Theorem 1. *For any penalized least squares estimator which is either selection consistent or estimation consistent when $\lambda_n \in (\lambda_{n,min}, \lambda_{n,max})$, where $\lambda_{n,min} \to 0$ as $n \to \infty$, then the SPSP estimator built on the solution path over $(\lambda_{n,min}, \lambda_{n,max})$ is selection consistent. That is*

$$\mathbb{P}(\hat{S} = S) \to 0 \text{ as } n \to \infty.$$

Proof. Suppose the penalized least squares estimator $\hat{\boldsymbol{\beta}}$ is estimation consistent when $\lambda_n \in (\lambda_{n,min}, \lambda_{n,max})$, that is $\hat{\boldsymbol{\beta}} \xrightarrow{p} \boldsymbol{\beta}^*$, thus we have

$$|\hat{\beta}_j| \xrightarrow{p} |\beta_j^*|, j \in S \text{ and } |\hat{\beta}_j| \xrightarrow{p} 0, j \in S^c.$$

Hence with probability 1, we have for any $j_1, j_2 \in S^c, j_3, j_4 \in S$,

$$\frac{\left||\hat{\beta}_{j_3}| - |\hat{\beta}_{j_4}|\right|}{\left||\hat{\beta}_{j_2}| - |\hat{\beta}_{j_3}|\right|} \to \frac{\left||\beta^*_{j_3}| - |\beta^*_{j_4}|\right|}{\left||\beta^*_{j_2}| - |\beta^*_{j_3}|\right|} = \frac{\left||\beta^*_{j_3}| - |\beta^*_{j_4}|\right|}{|\beta^*_{j_3}|} \leq C^*,$$
$$\frac{\left||\hat{\beta}_{j_2}| - |\hat{\beta}_{j_3}|\right|}{\left||\hat{\beta}_{j_1}| - |\hat{\beta}_{j_2}|\right|} \to \frac{\left||\beta^*_{j_2}| - |\beta^*_{j_3}|\right|}{\left||\beta^*_{j_1}| - |\beta^*_{j_2}|\right|} = \frac{|\beta^*_{j_2}| - 0}{0 - 0} = \infty,$$

where $C^* = \dfrac{\max\limits_{j \in S} |\beta^*_j| - \min\limits_{j \in S} |\beta^*_j|}{\min\limits_{j \in S} |\beta^*_j|}$ is a constant. Therefore the partitioning rule for the true relevant and irrelevant sets has been verified at $\lambda_n \in (\lambda_{n,min}, \lambda_{n,max})$, that is, with probability 1,

$$\begin{aligned} \frac{D_{\max}(\hat{S})}{D(\hat{S}, \hat{S}^c)} &\leq C^*, \\ \frac{D(\hat{S}, \hat{S}^c)}{D_{\max}(\hat{S}^c)} &\to \infty. \end{aligned}$$

In the SPSP algorithm, we begin with all the variables unimportant, i.e., $\hat{S}_0 = \emptyset$. Since the true relevant set and irrelevant set satisfies the partitioning rule over $(\lambda_{n,min}, \lambda_{n,max})$, there exists at least one $\lambda_n \in (\lambda_{n,min}, \lambda_{n,max})$ such that $\hat{S}(\lambda_n) = S$ for a large enough constant C in the SPSP algorithm.

Then we have

$$\hat{S} = \bigcup_{\lambda_n} \hat{S}(\lambda_n) \xrightarrow{p} S.$$

The proof is similar for the selection consistent case since the selection consistency of the penalized least squares estimation requires stronger conditions than the estimation consistency, then we still have with probability 1

$$|\hat{\beta}_j| > \hat{C} > 0, j \in S \text{ and } |\hat{\beta}_j| = 0, j \in S^c,$$

where $\hat{C}$ is a positive constant. □

It is well known that the existing penalized least squares estimators all enjoy either estimation consistent or selection consistent as $\lambda_n \to 0$. For instance, the Lasso, the adaptive Lasso, the SCAD,

the MCP all enjoy the selection consistency under some strong conditions while the ridge holds the estimation consistency. Hence, Theorem 1 can establish the selection consistency of the SPSP estimator on a wide class of the penalty functions. Since the selection consistency of the penalized least squares estimator is usually more difficult to obtain, applying the SPSP algorithm on the estimators which possess the estimation consistency is adequate to obtain the selection consistency. Moreover, Theorem 1 can be easily extended to other applications. Take the penalized likelihood estimation as an example, Fan and Peng (2004, [20]) established the asymptotic properties of the penalized likelihood estimators with diverging number of the parameters. One important conclusion in their paper is that under several regularity conditions on the penalty functions and the likelihood functions, in a certain range of λ_n, there exists a local penalized likelihood estimator $\hat{\boldsymbol{\beta}}$ such that

$$||\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^{*}||_1 = O_p(\sqrt{p}(n^{-1/2} + a_n)),$$

where $a_n = \max_{j \in S} \lambda_n J(|\beta_j^*|)$. In other words, the general penalized likelihood estimators can enjoy a root-(n/p)-estimation-consistent property in a certain range of λ_n. Therefore the SPSP algorithm on the penalized likelihood estimators can possess the selection consistency over the solution paths on this range.

4.2 Lasso

Generally, the Lasso estimator is defined as

$$\hat{\boldsymbol{\beta}}^{(Lasso)} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{n}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2 + \lambda_n||\boldsymbol{\beta}||_1, \tag{4.1}$$

where λ_n is the tuning parameter.

Lots of work has been done on the selection consistency of the Lasso estimator. It was shown the selection consistency requires a strong condition on the design matrix, called neighborhood stability ([40]) or irrepresentable condition ([64]). In this section, we will focus on the selection consistency of the SPSP estimator on the Lasso under the irrepresentable condition firstly and then discuss the

consistency under a weaker condition - the restricted eigenvalue condition ([5]).

4.2.1 Irrepresentable Condition

Theorem 1 established the consistency over the solution paths on $(\lambda_{n,min}, \lambda_{n,max})$. But actually we anticipate the proposed SPSP estimator should be selection consistent over the whole solution paths. In other words, even if we select the correct model over a part of the solution paths, we still need to ensure that the irrelevant features won't be selected over the remaining solution paths. On this purpose, we introduce the following property of the Lasso estimator firstly.

Lemma 2. *Let $\hat{\boldsymbol{\beta}} = (\hat{\boldsymbol{\beta}}_1, \mathbf{0})$ be the solution of the Lasso at the tuning parameter λ. Let X_1 be the sub matrix of $\mathbf{X}$ that consists of the columns corresponding to $\hat{\boldsymbol{\beta}}_1$ and X_2 be the sub matrix with the remaining columns. If $|(\frac{1}{n}X_2^T X_1)(\frac{1}{n}X_1^T X_1)^{-1}| < \mathbf{1}$ where $\mathbf{1}$ is a vector of 1's, then there exists a vector $\boldsymbol{\Delta} > 0$ of the same dimension as $\hat{\boldsymbol{\beta}}_1$, such that $(\hat{\boldsymbol{\beta}}_1 - SIGN(\hat{\boldsymbol{\beta}}_1)\boldsymbol{\Delta}, 0)$ is a solution for the Lasso with a larger tuning parameter $\lambda + l$, where SIGN$(\hat{\boldsymbol{\beta}}_1)$ is a diagonal matrix with the diagonal vector sign$(\hat{\boldsymbol{\beta}}_1)$ and $l > 0$ is any positive number.*

Proof. By the KKT condition, the sufficient and necessary condition for $\hat{\boldsymbol{\beta}}$ to be the solution of the Lasso at λ is

$$\frac{1}{n}X_1^T\varepsilon + \frac{1}{n}X_1^T X(\boldsymbol{\beta}^* - \hat{\boldsymbol{\beta}}) = \lambda \text{ sign}(\hat{\boldsymbol{\beta}}_1)$$
$$\left|\frac{1}{n}X_2^T\varepsilon + \frac{1}{n}X_1^T X(\boldsymbol{\beta}^* - \hat{\boldsymbol{\beta}})\right| \leq \lambda\mathbf{1}.$$

Same equations can be derived for $\hat{\boldsymbol{\beta}}' = (\hat{\boldsymbol{\beta}}_1 - \text{SIGN}(\hat{\boldsymbol{\beta}}_1)\boldsymbol{\Delta}, 0)$. That is, if $\hat{\boldsymbol{\beta}}'$ is the solution at $\lambda + l$, then we need to make sure

$$\frac{1}{n}X_1^T\varepsilon + \frac{1}{n}X_1^T X(\boldsymbol{\beta}^* - \hat{\boldsymbol{\beta}}') = (\lambda + l) \text{ sign}(\hat{\boldsymbol{\beta}}_1)$$
$$\left|\frac{1}{n}X_2^T\varepsilon + \frac{1}{n}X_1^T X(\boldsymbol{\beta}^* - \hat{\boldsymbol{\beta}}')\right| \leq (\lambda + l)\mathbf{1}.$$

Taking the differences of the above two sets of equations, we only need the following to be true

in order for $\hat{\boldsymbol{\beta}}'$ to be the solution at $\lambda + l$,

$$\frac{1}{n} X_1^T X_1 \text{SIGN}(\hat{\boldsymbol{\beta}}_1)\boldsymbol{\Delta} = l \ \text{sign}(\hat{\boldsymbol{\beta}}_1)$$
$$\left| \frac{1}{n} X_2^T X_1 \text{SIGN}(\hat{\boldsymbol{\beta}}_1)\boldsymbol{\Delta} \right| \leq l\mathbf{1}.$$

It follows from the condition $|(\frac{1}{n}X_2^T X_1)(\frac{1}{n}X_1^T X_1)^{-1}| < \mathbf{1}$ immediately that $\boldsymbol{\Delta} = l(\frac{1}{n}X_1^T X_1)^{-1}\mathbf{1}$ satisfies both the above equation and inequality. □

The lemma actually suggests that once the coefficient of a certain feature has been shrunk to zero, it will not jump back to a non-zero value for a larger tuning parameter. The lemma is quite useful to control the behavior of the irrelevant variables in the model.

Before we establish the selection consistency of the SPSP algorithm on the Lasso over the whole solution paths, it is natural to assume some regularity conditions on the design matrix. Zhao and Yu (2006, [64]) proposed some regularity conditions as follows.

There exist $0 \leq c_1 < c_2 \leq 1$ and $M_1, M_2 > 0$ such that

$$\alpha^T C_{11}^n \alpha \geq M_1, \text{ for } \forall ||\alpha||_2^2 = 1, \tag{4.2}$$
$$p = O(n^{c_1}), \tag{4.3}$$
$$s = O(e^{n^{c_3}}), 0 \leq c_3 < c_2 - c_1, \tag{4.4}$$
$$n^{\frac{1-c_2}{2}} \min_{j \in S} |\beta_j^*| \geq M_2, \tag{4.5}$$

where s is the number of the relevant variables in the model, $C_{11}^n = \frac{1}{n}(X_1^*)^T X_1^*$ and X_1^* is the sub matrix of $\mathbf{X}$ that consists of columns corresponding to the relevant features. Note that these conditions are commonly assumed in the theoretical studies of the penalized least squares estimation. See [64] and [39] for more details on the conditions.

Now we can establish the following result:

Theorem 3. *Under the regularity conditions* (4.2)-(4.5)*, there exists* $\lambda_{n,min}$ *where* $\lambda_{n,min} \to 0$ *as*

$n \to \infty$, such that the SPSP estimator on the Lasso built on the solution paths over $(\lambda_{n,min}, \infty)$ is selection consistent under the following irrepresentable condition ([64])

$$|C_{21}^n (C_{11})^{-1}| \leq 1 - \eta,$$

where η is a positive constant, $C_{11}^n = \frac{1}{n}(X_1^*)^T X_1^*$, $C_{21}^n = \frac{1}{n}(X_2^*)^T X_1^*$, and X_1^* is the sub matrix of $\mathbf{X}$ that consists of columns corresponding to the relevant features, X_2^* is the sub matrix with the irrelevant columns.

Proof. For $\lambda_n \propto n^{\frac{c_4-1}{2}}$ with $0 \leq c_4 < 1$, the Lasso estimator is selection consistent based on the Theorem 4 in [64], therefore the selection consistency of the SPSP estimator over $(\lambda_{n,min}, \lambda_{n,max})$ can be obtained directly from the previous Theorem 1, where $\lambda_{n,min}, \lambda_{n,max} \propto n^{\frac{c_4-1}{2}}$. That is, for $\lambda_n \in (\lambda_{n,min}, \lambda_{n,max})$,

$$\hat{S}(\lambda_n) \xrightarrow{p} S \text{ as } n \to \infty.$$

As $\lambda_n > \lambda_{n,max}$, according to Lemma 2, under the strong irrepresentable condition, the estimated zero coefficients in $(\lambda_{n,min}, \lambda_{n,max})$ remain to be zero, which indicates that these variables will be identified as relevant at $\lambda_n > \lambda_{n,max}$. Thus we have for $\lambda_n > \lambda_{n,max}$,

$$\hat{S}(\lambda_n) \subseteq S.$$

Especially, when $\lambda_n \to \infty$, we have $|\hat{\beta}_j| \xrightarrow{p} 0, j = 1, \cdots, p$, which also suggests $\hat{S}(\lambda_n) \subseteq S$ with probability 1. Since the final relevant set of the SPSP approach is $\bigcup_{\lambda_n} \hat{S}(\lambda_n) \xrightarrow{p} S$, the selection consistency of the SPSP estimator on the Lasso built on the solution paths over the $(\lambda_{n,min}, \infty)$ has been proved.

□

4.2.2 Restricted Eigenvalue Condition

We established the selection consistency of the SPSP estimator on the Lasso over the whole solution paths in Theorem 3; however, one issue on the concern of the conditions in Theorem 3 is that the irrepresentable condition is too strong to fulfill. Much work has shown that if the condition is violated, the Lasso may perform poorly in selecting the correct model ([42]). Hence, it will be better if we can seek some weak assumptions for the SPSP algorithm on the Lasso.

For controlling the prediction error of the Lasso in high dimensional cases, Bickel et al. (2009, [5]) proposed the following restricted eigenvalue condition.

Assumption 1. *Restricted Eigenvalue Condition $RE(s_0, c_0)$ ([5]) For some integer s_0 such that $1 \leq s_0 \leq p$, and a positive number c_0, the following condition holds:*

$$\kappa(s_0, c_0) = \min_{\substack{S \subseteq \{1,\cdots,p\}, \\ |S| \leq s_0}} \min_{\substack{\boldsymbol{\delta} \neq \mathbf{0}, \\ ||\boldsymbol{\delta}_{S^c}||_1 \leq c_0 ||\boldsymbol{\delta}_S||_1}} \frac{||\mathbf{X}\boldsymbol{\delta}||_2}{\sqrt{n}||\boldsymbol{\delta}_S||_2} > 0. \tag{4.6}$$

The number c_0 indicates that this condition is valid only for the vectors satisfying

$$||\boldsymbol{\delta}_{S^c}||_1 \leq c_0 ||\boldsymbol{\delta}_S||_1,$$

and the integer s_0 plays the role of an upper bound on the sparsity of the vector. The general idea is that the minimum eigenvalue of the Gram matrix $\frac{1}{n}\mathbf{X}^T\mathbf{X}$ should be greater than 0 over a restricted subset of the vectors instead of all the vectors. While for the Lasso residuals $\boldsymbol{\delta} = \hat{\boldsymbol{\beta}}^{(Lasso)} - \boldsymbol{\beta}^*$, we have with probability close to 1,

$$||\boldsymbol{\delta}_{S^c}||_1 \leq 3||\boldsymbol{\delta}_S||_1,$$

hence we only require $c_0 = 3$ for the Lasso in the assumption (See [5]).

It is known that the restricted eigenvalue condition (REC) is weaker than the irrepresentable condition ([8]). Note that there are some other related assumptions with the restricted eigenvalue condition such as the compatibility condition ([55]), the restricted isometry condition ([10]), and the

coherence condition ([9]).

Under the REC assumption, we can control the prediction error of the Lasso in a non-asymptotic form and then we have the following result:

Theorem 4. *Suppose that $RE(s, c_0)$ is satisfied with $c_0 = 3$ and κ. Consider the Lasso estimator $\hat{\boldsymbol{\beta}} = (\hat{\beta}_1, \cdots, \hat{\beta}_p)^T$ defined in* (4.1) *with $\lambda_n = A\sigma\sqrt{\frac{\log p}{n}}$, where A is a constant and $A > 4\sqrt{2}$. Suppose $\min_{j\in S} |\beta_j^*| - \frac{8A\sigma}{\kappa^2} s\sqrt{\frac{\log p}{n}} \geq C_1 > 0$, where C_1 is a constant. If $s\sqrt{\log p} \leq \sqrt{n}$, then*

$$\mathbb{P}\left(\frac{\min_{j_2\in S^c, j_3\in S}\left||\hat{\beta}_{j_2}| - |\hat{\beta}_{j_3}|\right|}{\max_{j_1,j_2\in S^c}\left||\hat{\beta}_{j_1}| - |\hat{\beta}_{j_2}|\right|} \to \infty\right) = 1 \text{ as } n \to \infty, \tag{4.7}$$

$$\mathbb{P}\left(\frac{\max_{j_3,j_4\in S}\left||\hat{\beta}_{j_3}| - |\hat{\beta}_{j_4}|\right|}{\min_{j_2\in S^c, j_3\in S}\left||\hat{\beta}_{j_2}| - |\hat{\beta}_{j_3}|\right|} \leq C^*\right) \to 1 \text{ as } n \to \infty, \tag{4.8}$$

where $C^* = \dfrac{\max_{j\in S}|\beta_j^*| - \min_{j\in S}|\beta_j^*|}{\min_{j\in S}|\beta_j^*|}$.

Proof. Under the Restricted Eigenvalue Condition $RE(s_0, 3)$, if $\lambda_n = A\sigma\sqrt{\frac{\log p}{n}}$, then with probability at least $1 - p^{1-A^2/32}$ (See [5]),

$$||\boldsymbol{\delta}_S||_1 \leq \frac{2A\sigma}{\kappa^2} s\sqrt{\frac{\log p}{n}}, ||\boldsymbol{\delta}_{S^c}||_1 \leq \frac{6A\sigma}{\kappa^2} s\sqrt{\frac{\log p}{n}}.$$

To prove (4.7), it suffices to see that

$$\max_{j_1,j_2\in S^c}\left||\hat{\beta}_{j_1}| - |\hat{\beta}_{j_2}|\right| \leq \max_{j\in S^c}|\beta_j| \leq ||\boldsymbol{\delta}_{S^c}||_1,$$

and for any $j_2 \in S^c, j_3 \in S$,

$$\begin{aligned}\left|\left|\hat{\beta}_{j_2}\right| - \left|\hat{\beta}_{j_3}\right|\right| &\geq \left|\beta^*_{j_3}\right| - \left|\left|\hat{\beta}_{j_3}\right| - \left|\beta^*_{j_3}\right|\right| - \left|\hat{\beta}_{j_2}\right| \\ &\geq \min_{j\in S}\left|\beta^*_j\right| - \left|\delta_{j_3}\right| - \left|\delta_{j_2}\right| \\ &\geq \min_{j\in S}\left|\beta^*_j\right| - ||\boldsymbol{\delta}||_1.\end{aligned}$$

Then with probability at least $1 - p^{1-A^2/32}$,

$$\begin{aligned}\frac{\min_{j_2\in S^c, j_3\in S}\left|\left|\hat{\beta}_{j_2}\right| - \left|\hat{\beta}_{j_3}\right|\right|}{\max_{j_1,j_2\in S^c}\left|\left|\hat{\beta}_{j_1}\right| - \left|\hat{\beta}_{j_2}\right|\right|} &\geq \frac{\min_{j\in S}\left|\beta^*_j\right| - ||\boldsymbol{\delta}||_1}{||\boldsymbol{\delta}_{S^c}||_1} \geq \frac{C_1\kappa^2}{6A\sigma}\frac{\sqrt{n}}{s\sqrt{\log p}}, \\ &\to \infty \text{ as } n \to \infty.\end{aligned}$$

The proof of (4.8) is analogous. We can see for any $j_3, j_4 \in S$

$$\begin{aligned}\left|\left|\hat{\beta}_{j_3}\right| - \left|\hat{\beta}_{j_4}\right|\right| &\leq \left|\left|\beta^*_{j_3}\right| - \left|\beta^*_{j_4}\right|\right| + \left|\left|\hat{\beta}_{j_3}\right| - \left|\beta^*_{j_3}\right|\right| + \left|\left|\hat{\beta}_{j_4}\right| - \left|\beta^*_{j_4}\right|\right| \\ &\leq \max_{j\in S}\left|\beta^*_j\right| - \min_{j\in S}\left|\beta^*_j\right| + ||\boldsymbol{\delta}_S||_1.\end{aligned}$$

Then with probability at least $1 - p^{1-A^2/32}$,

$$\begin{aligned}\frac{\max_{j_3,j_4\in S}\left|\left|\hat{\beta}_{j_3}\right| - \left|\hat{\beta}_{j_4}\right|\right|}{\min_{j_2\in S^c, j_3\in S}\left|\left|\hat{\beta}_{j_2}\right| - \left|\hat{\beta}_{j_3}\right|\right|} &\leq \frac{\max_{j\in S}\left|\beta^*_j\right| - \min_{j\in S}\left|\beta^*_j\right| + ||\boldsymbol{\delta}_S||_1}{\min_{j\in S}\left|\beta^*_j\right| - ||\boldsymbol{\delta}||_1}, \\ &\to C^* \text{ as } n \to \infty.\end{aligned}$$

□

The beta-min condition here is a common condition in the selection consistency which requires the true nonzero coefficients to be sufficiently large. Theorem 4 states that in a certain range of the tuning parameters, the distances between the irrelevant coefficients of the Lasso estimator are much smaller than the distance between the relevant coefficient and the irrelevant coefficient while

the distances between the relevant coefficients and the irrelevant coefficients should be at least the same order of magnitude as the distances between the relevant coefficients. Consequently, we can apply this property in the SPSP algorithm to distinguish the relevant variables from the irrelevant variables.

Note that Theorem 4 only shows the selection consistency of the SPSP estimator on the Lasso over part of the solution paths. For the consistency over the whole solution paths especially at small tuning parameters, we still need to add some stability conditions to control the estimation behavior.

4.3 General Case

In this section, we will generalize the theoretical results of the Lasso to the general penalty functions using a similar restricted eigenvalue condition. Firstly, We define the general penalized least squares estimator as

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{n}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2 + \lambda_n \sum_{j=1}^{p} J(|\beta_j|), \tag{4.9}$$

where $J(\cdot)$ is a general penalty function.

Due to the high dimensionality of the data, we must impose some uniform conditions on the penalty function. Under the framework of the least squares (L_2) loss, we have to control the penalty terms on the purpose of the estimation accuracy. A higher-order derivative restriction on the penalty function is a simple way to bound the penalty terms.

Assumption 2. *There is a large enough open subset Ω of $\mathbb{R}^p$ which contains the true parameter $\boldsymbol{\beta}^*$ and all the possible penalized least squares estimators in* (4.9)*, such that for almost all $\bar{\boldsymbol{\beta}}$ in Ω, all $J''(\bar{\beta}_j), j = 1, \cdots, p$ exist. Furthermore the supremum of the absolute values of the second derivatives exists, denoted as*

$$D = \sup\{|J''(\bar{\beta}_j)|, j = 1, \cdots, p, \bar{\boldsymbol{\beta}} \in \Omega\}.$$

Under the assumption, the second derivative of the penalty function is bounded, which is necessary for controlling the error bounds in the estimation. Clearly, the Lasso and the adaptive Lasso penalties both satisfy this assumption with $D = 0$. For the ridge penalty, we have $D = 2$. Moreover, for the SCAD penalty,

$$J'_{SCAD}(\theta) = I(\theta \leq \lambda_n) + \frac{(a\lambda_n - \theta)_+}{(a-1)\lambda_n} I(\theta > \lambda_n)$$

for some $a > 2$ and $\theta > 0$ and the hard thresholding penalty function

$$J_{HT}(\theta) = \lambda_n - \frac{(|\theta| - \lambda_n)^2}{\lambda_n} I(|\theta| < \lambda_n),$$

the second derivatives are bounded as long as we restrict the range of the tuning parameter λ_n.

Define the general residual as $\boldsymbol{\delta} = \hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*$ and the maximal value of the first derivative of the true nonzero coefficients as $f_n = \max_{j \in S} J'(|\beta_j^*|)$, we present a basic inequality for the general penalized least squares estimator as follows.

Lemma 5. *Consider the linear model* (1.1)*, let* $\hat{\boldsymbol{\beta}}$ *be the estimator in* (4.9) *with* $\lambda_n = A\sigma\sqrt{\frac{\log p}{n}}$*, where* A *is a constant and* $A > 4\sqrt{2}$*, under Assumption 2, then with probability at least* $1 - p^{1-A^2/32}$ *, we have*

$$\frac{1}{n}||\mathbf{X}\boldsymbol{\delta}||_2^2 + \lambda_n \sum_{j \in S^c} J(|\hat{\delta}_j|) \leq \frac{\lambda_n}{2}||\boldsymbol{\delta}||_1 + \lambda_n(f_n||\boldsymbol{\delta}_S||_1 + \frac{D}{2}||\boldsymbol{\delta}_S||_2^2).$$

Proof. The first part of proof is similar with the proof of Lemma B.1 in [5]. For the completeness, we begin from the following basic inequality

$$\frac{1}{n}||\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}||_2^2 + \lambda_n \sum_{j=1}^{p} J(|\hat{\beta}_j|) \leq \frac{1}{n}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^*||_2^2 + \lambda_n \sum_{j=1}^{p} J(|\beta_j^*|),$$

which is equivalent to

$$\frac{1}{n}||\mathbf{X}\boldsymbol{\delta}||_2^2 + \lambda_n \sum_{j\in S^c} J(|\hat{\beta}_j|) \leq \lambda_n(\sum_{j\in S} J(|\beta_j^*|) - \sum_{j\in S} J(|\hat{\beta}_j|)) + 2\sum_{j=1}^{p}(\hat{\beta}_j - \beta_j^*)(\frac{\sum_{i=1}^n x_{ij}\epsilon_i}{n}).$$

Define $V_j = \frac{\sum_{i=1}^n x_{ij}\epsilon_i}{n}, 1 \leq j \leq p$, then $V_j \sim \mathcal{N}(0, \frac{\sigma^2}{n})$ and consider the event

$$\mathscr{A} = \bigcap_{j=1}^{p} \left\{ |V_j| \leq \frac{\lambda}{4} \right\},$$

thus

$$\mathbb{P}(\mathscr{A}^c) \leq p \cdot \mathbb{P}\left(\frac{\sqrt{n}}{\sigma}|V_j| > \frac{\sqrt{n}\lambda_n}{4\sigma} \right) \leq p \exp\left(-\frac{n\lambda_n^2}{32\sigma^2} \right) = p^{1-A^2/32}.$$

Hence on event $\mathscr{A}$,

$$\frac{1}{n}||\mathbf{X}\boldsymbol{\delta}||_2^2 + \lambda_n \sum_{j\in S^c} J(|\hat{\beta}_j|) \leq \frac{\lambda_n}{2}||\boldsymbol{\delta}||_1 - \lambda_n(\sum_{j\in S} J(|\hat{\beta}_j|) - \sum_{j\in S} J(|\beta_j^*|)). \tag{4.10}$$

Consider $J(|\hat{\beta}_j|) - J(|\beta_j^*|), j \in S$, by Taylor Expansion,

$$J(|\hat{\beta}_j|) - J(|\beta_j^*|) = J'(|\beta_j^*|)\delta_j \text{sign}(\beta_j^*) + \frac{J''(|\bar{\beta}_j|)}{2}\delta_j^2,$$

where $|\bar{\beta}_j|$ lies between $|\beta_j^*|$ and $|\hat{\beta}_j|$. Thus

$$|J(|\hat{\beta}_j|) - J(|\beta_j^*|)| \leq f_n|\delta_j| + \frac{D}{2}\delta_j^2,$$
$$\sum_{j\in S} |J(|\hat{\beta}_j|) - J(|\beta_j^*|)| \leq f_n||\boldsymbol{\delta}_S||_1 + \frac{D}{2}||\boldsymbol{\delta}_S||_2^2$$

Then the result can be derived from (4.10) directly. □

This lemma suggests one interesting property on the general residual, that is, with probability

close to 1, the general residual must satisfy

$$\sum_{j\in S^c} J(|\delta_j|) - \frac{1}{2}||\boldsymbol{\delta}_{S^c}||_1 \leq \frac{1}{2}||\boldsymbol{\delta}_S||_1 + f_n||\boldsymbol{\delta}_S||_1 + \frac{D}{2}||\boldsymbol{\delta}_S||_2^2.$$

Similarly, in the general high-dimensional scenario, we impose some conditions on the design matrix in order to obtain the error bounds for the general penalized least squares estimator.

Assumption 3. *(Generalized Restricted Eigenvalue Condition $RE(s_0, \Psi)$) For some integer s_0 such that $1 \leq s_0 \leq p$, and a function Ψ, the following condition holds:*

$$\kappa(s_0, \Psi) = \min_{\substack{S\subseteq\{1,\cdots,p\},\\ |S|\leq s_0}} \min_{\substack{\boldsymbol{\delta}\neq\mathbf{0},\\ \boldsymbol{\delta}\in\Psi(S)}} \frac{||\mathbf{X}\boldsymbol{\delta}||_2}{\sqrt{n}||\boldsymbol{\delta}||_1} > 0,$$

where $\Psi(S) = \left\{\boldsymbol{\delta} \in \mathbb{R}^p : \sum_{j\in S^c} J(|\delta_j|) - \frac{1}{2}||\boldsymbol{\delta}_{S^c}||_1 \leq \frac{1}{2}||\boldsymbol{\delta}_S||_1 + f_n||\boldsymbol{\delta}_S||_1 + \frac{D}{2}||\boldsymbol{\delta}_S||_2^2\right\}$.

The assumption is similar as the restricted eigenvalue condition (4.6). We also require that the minimum restricted eigenvalue of the design matrix over a subset of the vectors is positive. Note that we use the L_1 norm in the denominator to facilitate the derivation in the general case. In particular, we can specify the subset once we choose the penalty function.

Theorem 6. *Let the conditions of Lemma 5 be satisfied, then under the condition $RE(s, \Psi)$ with κ, if $A < \frac{2\kappa^2}{D\sigma}\sqrt{\frac{n}{\log p}}$, with probability at least $1 - p^{1-A^2/32}$, we have*

$$||\boldsymbol{\delta}||_1 \leq \frac{(1+2f_n)A\sigma\sqrt{\log p}}{2\sqrt{n}\kappa^2 - AD\sigma\sqrt{\log p}}.$$

Proof. Under Assumption 3, the general penalized least squares residual satisfies

$$||X\boldsymbol{\delta}||_2 > \sqrt{n}\kappa||\boldsymbol{\delta}||_1.$$

And in Lemma 5, we have

$$\begin{aligned}\frac{1}{n}||\mathbf{X}\boldsymbol{\delta}||_2^2 &\leq \frac{\lambda_n}{2}||\boldsymbol{\delta}||_1 + \lambda_n(f_n||\boldsymbol{\delta}_S||_1 + \frac{D}{2}||\boldsymbol{\delta}_S||_2^2), \\ &\leq (\frac{1}{2}+f_n)\lambda_n||\boldsymbol{\delta}||_1 + \frac{D}{2}\lambda_n||\boldsymbol{\delta}||_2^2, \\ &\leq (\frac{1}{2}+f_n)\lambda_n||\boldsymbol{\delta}||_1 + \frac{D}{2}\lambda_n||\boldsymbol{\delta}||_1^2,\end{aligned} \tag{4.11}$$

Hence

$$\begin{aligned}\kappa^2||\boldsymbol{\delta}||_1^2 &\leq (\frac{1}{2}+f_n)\lambda_n||\boldsymbol{\delta}||_1 + \frac{D}{2}\lambda_n||\boldsymbol{\delta}||_1^2, \\ (\kappa^2 - \frac{D}{2}\lambda_n)||\boldsymbol{\delta}||_1^2 &\leq (\frac{1}{2}+f_n)\lambda_n||\boldsymbol{\delta}||_1, \\ ||\boldsymbol{\delta}||_1 &\leq \frac{(1+2f_n)A\sigma\sqrt{p\log p}}{2\sqrt{n}\kappa^2 - AD\sigma\sqrt{\log p}}.\end{aligned}$$

Thus the inequality in Theorem 6 holds. □

Theorem 6 provides the bound of the estimation error for the general penalized least squares estimator. Moreover, the prediction error $\frac{1}{n}||\mathbf{X}\boldsymbol{\delta}||_2^2$ can also be derived from (4.11). The upper bound on the constant A controls the range of the tuning parameter λ_n. Particularly, the upper bound is infinity for the case $D = 0$ such as the Lasso and the adaptive Lasso. In general, in the asymptotic sense, assuming $\sqrt{\log p} < \sqrt{n}$, the upper bound can be sufficiently large.

Now we can summarize the following result for the SPSP algorithm on the general least squares estimator.

Theorem 7. *Consider the general penalized least squares estimator* $\hat{\boldsymbol{\beta}} = (\hat{\beta}_1, \cdots, \hat{\beta}_p)^T$ *defined in* (4.9) *with* $\lambda_n = A\sigma\sqrt{\frac{\log p}{n}}$*, where* A *is a constant and* $A > 4\sqrt{2}$*. Under the same conditions in Theorem 6, suppose* $\min_{j\in S}|\beta_j^*| - \frac{(1+2f_n)A\sigma\sqrt{\log p}}{2\sqrt{n}\kappa^2 - AD\sigma\sqrt{\log p}} \geq C_2 > 0$*, where* C_2 *is a constant. If*

$\sqrt{\log p} \leq \sqrt{n}$, *then*

$$\mathbb{P}\left(\frac{\min\limits_{j_2\in S^c, j_3\in S}\left||\hat{\beta}_{j_2}|-|\hat{\beta}_{j_3}|\right|}{\max\limits_{j_1,j_2\in S^c}\left||\hat{\beta}_{j_1}|-|\hat{\beta}_{j_2}|\right|}\to\infty\right)=1 \textit{ as } n\to\infty,$$

$$\mathbb{P}\left(\frac{\max\limits_{j_3,j_4\in S}\left||\hat{\beta}_{j_3}|-|\hat{\beta}_{j_4}|\right|}{\min\limits_{j_2\in S^c, j_3\in S}\left||\hat{\beta}_{j_2}|-|\hat{\beta}_{j_3}|\right|}\leq C^*\right)\to 1 \textit{ as } n\to\infty,$$

where $C^*=\dfrac{\max\limits_{j\in S}|\beta_j^*|-\min\limits_{j\in S}|\beta_j^*|}{\min\limits_{j\in S}|\beta_j^*|}$.

Proof. The proof is analogous to the proof of Theorem 4. □

Here Theorem 7 provides the similar results as Theorem 4. It proves the partitioning rule for the SPSP algorithm on the general penalty functions over a certain range of the tuning parameters. It is worth mentioning that the conditions in Theorem 7 can be relaxed once we specify the penalty function in (4.9). The further theoretical properties of the SPSP estimators over the whole solution paths will be left as the future work.

CHAPTER 5 SIMULATION STUDIES

This chapter examines the selection accuracy of the SPSP algorithm through six simulated examples. Specifically, we apply four popular penalty functions in the simulation studies: the Lasso, the adaptive Lasso, the SCAD and the MCP. For each penalty, we compare the performance of the proposed algorithm with the criteria for selecting the tuning parameter including the CV, the GCV, the AIC, the BIC and the EBIC. Moreover, we also present the comparison between the AIS algorithm and two other stepwise selection approaches: the forward selection and the LARS algorithm. Finally, we provide a simple example of the SPSP algorithm in the graphical modeling. All of the simulation studies in the chapter are implemented in the $\mathbf{R}$ environment.

5.1 SPSP for variable selection

We carry out extensive simulation studies to compare the SPSP procedure and the AIS algorithm with the current existing approaches. For the examples either with small number of the variables or with simple correlation structures (such as Example 1 and 2 in [52]), the simulation studies show that both the proposed SPSP algorithm and some of the existing approaches have good performance in selecting the correct model. The results of these examples are available upon request.

5.1.1 Simulation Settings

We mainly propose the SPSP algorithm for the variable selection problems with high correlations in the data sets, especially for the high dimensional data problems. Therefore, we present six simulation examples with relatively complicated correlation structures here. The first simulation illustrates one low dimensional case, where the relevant variables are highly correlated. The other five simulation settings all have high dimensional features ($p > n$). In the second simulation, the relevant variables are highly correlated with different signs. To examine the performances of the proposed SPSP algorithm over different noise levels, we increase the noise level of the second simulation for the third simulation setting. For the following two simulation examples, the true models

both contain 3 relevant variables, while the number of irrelevant variables are 97 and 997 respectively. We can evaluate the performance of the proposed algorithm for sparse models through these two examples. The last simulation is also from a high dimensional linear regression, but the true model is not contained in the model space. The simulation is used to examine the performance of the proposed algorithm in the misspecified models.

All the following simulations are generated from model (1.1) with

$$x_{ij} \sim N(0,1), i = 1, \cdots, n, j = 1, \cdots, p$$

and $\epsilon_i \sim N(0, \sigma^2)$. The details of the simulations are described as follows.

(1) Let

$$\boldsymbol{\beta}^* = (3, 3, 3, 3, 3, -2, -2, -2, -2, -2, 0, \cdots, 0),$$

where the first 10 coefficients are nonzero and the remaining 30 coefficients are zero. The pairwise correlation between the first 10 variables is 0.9. The remaining 30 variables are independent with each other, and are also independent with the first 10 variables. We set $\sigma = 3$ and $n = 50$.

(2) Let $\boldsymbol{\beta}^* = (3, 3, -2, 3, 3, -2, 0, \cdots, 0)$, where the first 6 coefficients are nonzero and the remaining 94 coefficients are zero. The pairwise correlation between the first 3 variables is 0.9, the pairwise correlation between the second 3 variables is also 0.9 and the remaining 94 variables are independent with each other. Furthermore, the first 3 variables, the second 3 variables, and the remaining 94 variables are independent with each other. We set $\sigma = 1$ and $n = 50$.

(3) The same as Model (2) except that we set $\sigma = 3$.

(4) There are 100 variables in the model. Let $\beta_1^* = 3, \beta_2^* = 1.5, \beta_5^* = 2$, and the remaining 97 coefficients are zero. The correlation between x_{j1} and x_{j2} is $0.5^{|j_1 - j_2|}$. We set $\sigma = 2$ and

$n = 50$.

(5) The same as Model (4) except that we set 1000 variables in the model.

(6) The true coefficients in the model is

$$\boldsymbol{\beta}^* = (1, -1.25, 0.75, -0.95, 1.5, 0, \cdots, 0),$$

where among these 100 coefficients, the first 10 are nonzero and the remaining 95 are zero. All the variables are independent of each other, i.e, the rows of $\mathbf{X}$ is generated from $N(\mathbf{0}, \mathbf{I}_p)$. We consider $n = 50$ and $\sigma = 0.25$ in the model. Here the response y is obtained from

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta}^* + \mathbf{x}_{p+1} + \boldsymbol{\varepsilon},$$

where $\mathbf{x}_{p+1} = \mathbf{x}_1 \circ \mathbf{x}_2$ is an interaction term which is the product of the first two variables. However, we still apply the model (1.1): $\mathbf{y} = \mathbf{X}\boldsymbol{\beta}^* + \boldsymbol{\varepsilon}$ to estimate the coefficients. Therefore, the true model cannot be correctly specified in this simulation study.

The first three examples are similar with the examples in the random lasso paper by Wang et al. (2011, [57]), which are designed to investigate the performance of the existing methods for the data with complicated correlation structures. The following two examples are selected from the truncated L_1 penalty paper by Shen et al. (2012, [48]). The last misspecified model is taken from Lv and Liu (2014, [37]). We aim at examining the performance of the proposed methods through these six examples with different correlation structures.

To compute the solution paths for Lasso and adaptive Lasso, we utilized the $\mathbf{R}$ package *lassoshooting* ([1]), and while the initial weight of the adaptive Lasso is obtained by the ridge estimator with a small shrinkage factor ($\lambda = 0.01$). For SCAD and MCP, $\mathbf{R}$ package *plus* ([62]) is implemented to obtain the solution paths.

For computing the solution paths of the penalized least squares estimators, we need to initially

select a grid of tuning parameters λ. For the Lasso and the adaptive Lasso, as discussed before, we select the grid to be equi-distant on the log scale as follows:

$$\lambda_{\min} = \lambda_1 < \cdots < \lambda_K = \lambda_{\max},$$

where $\lambda_{\min} = 1/n$, λ_{max} is the smallest λ yielding $\hat{\boldsymbol{\beta}} = 0$. That is,

$$\lambda_{k+1} = \lambda_k \exp\left(\frac{\log(\lambda_{\max}) - \log(\lambda_{\min})}{K-1}\right), k = 1, \cdots, K-1.$$

We apply $K = 100$ in all the simulations. For the ridge, we apply the same grid as the Lasso. For the SCAD and the MCP, the grid of tuning parameters are generated automatically by the *plus* function.

In the SPSP algorithm, the selection of the constant C is data-adaptive. Generally, for each penalty, we obtain an estimator with a small shrinkage factor ($\lambda = 0.01$), and then compute the adjacent distances of the sorted absolute values of the coefficients. Thus, the constant C can be computed as

$$C = \frac{\text{Maximal adjacent distance}}{\text{Second maximal adjacent distance}}. \tag{5.1}$$

Considering the criteria for selecting the tuning parameter λ, we apply the two-fold CV to reduce the computational cost. And for the GCV, the AIC, the BIC, the EBIC, as introduced in Chapter 1, the formulas are given as follows,

$$\begin{aligned}
\lambda_{GCV} &= \underset{\lambda_k}{\operatorname{argmin}} \left(\frac{SSE(\lambda_k)}{n(1-\hat{s}_k/n)^2}\right), \\
\lambda_{AIC} &= \underset{\lambda_k}{\operatorname{argmin}}(n\log(SSE(\lambda_k)) + 2\hat{s}_k), \\
\lambda_{BIC} &= \underset{\lambda_k}{\operatorname{argmin}}(n\log(SSE(\lambda_k)) + \hat{s}_k \log n), \\
\lambda_{EBIC} &= \underset{\lambda_k}{\operatorname{argmin}}(n\log(SSE(\lambda_k)) + \hat{s}_k(2logp + \log n)),
\end{aligned}$$

where $SSE(\lambda_k) = ||\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_{\boldsymbol{k}}||_2^2$, $\hat{s}_k$ is the number of the nonzero variables at the tuning parameter

λ_k, i.e.,

$$\hat{s}_k = \sum_{j=1}^{p} I(\hat{\beta}_{k,j} \neq 0).$$

For each simulation, we record the following measures: FP (False Positive, the number of zero variables incorrectly identified as nonzero), FN (False Negative, the number of nonzero variables incorrectly identified as zero), PS (FP+FN, the total number of the incorrect selections), and ME (Model Error=$(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)^T \hat{\boldsymbol{\Sigma}} (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)/\sigma^2$). Note that in each simulation, the true model is known, therefore we also record the result of the "oracle" selection in a sense that we select the "best" tuning parameter which minimizes the number of the incorrect selections (i.e., the smallest "PS" value).

We report the mean and the standard error (standard deviation$/\sqrt{B}$) of all these values over $B = 100$ replicates for these six examples. We also tried $B = 200$ and the results are similar.

5.1.2 The Main Results

The results for these six examples are summarized in Table 5.1-5.6.

From Table 5.1, we can observe that the CV, the GCV, the AIC and the BIC on all the penalties tend to select too many variables in the model, which lead to large FP values. For the EBIC, the FP value is almost 0 but the FN value is large. It indicates that this criterion excludes most of the informative variables for producing an over-sparse model. It is also seen that the SPSP on the Lasso, the adaptive Lasso, the SCAD and the MCP all perform well in excluding the irrelevant variables in the model (small FP values). Meanwhile, the SPSP on all the penalties exclude almost half of the relevant variables in the model, which is better than the results of the EBIC. Finally, we also notice that the model errors of the SPSP are smaller than the other selection criteria for all the penalties.

Table 5.2-5.6 show the results for high dimensional examples. In Table 5.2, we can see that the FN values of all the approaches are close on the same penalty; however, the SPSP algorithm on the penalty has much smaller FP values than the other selection criteria. It indicates that the proposed algorithm can remarkably reduce the number of the irrelevant variables for this high dimensional example. Even when the noise level increases in Example 3, the SPSP Algorithm still performs well in reducing the FP values compared with the other criteria.

Table 5.1: Simulation Results of Example 1 (low-dimension) over 100 replicates for the Lasso, the ALasso (Adaptive Lasso), the SCAD, the MCP (Standard Error in the parentheses).

		SPSP	2-CV	GCV	AIC	BIC	EBIC	Oracle
Lasso	FP	0.63	8.31	10.42	26.12	5.03	0.77	0.82
		(0.24)	(1.61)	(1.28)	(0.77)	(1.32)	(0.13)	(0.13)
	FN	5.11	5.58	5.79	1.69	6.27	7.03	6.95
		(0.28)	(0.39)	(0.32)	(0.27)	(0.33)	(0.13)	(0.12)
	PS	5.74	13.89	16.21	27.81	11.30	7.80	7.77
		(0.33)	(1.27)	(1.03)	(0.60)	(1.05)	(0.20)	(0.19)
	ME	0.44	0.59	0.53	0.72	0.59	0.57	0.57
		(0.02)	(0.02)	(0.02)	(0.03)	(0.02)	(0.02)	(0.02)
ALasso	FP	0.65	5.63	10.62	22.59	4.12	0.19	0.37
		(0.20)	(1.44)	(0.91)	(1.01)	(0.97)	(0.07)	(0.09)
	FN	4.03	6.57	4.55	1.40	6.73	8.09	7.19
		(0.29)	(0.43)	(0.38)	(0.24)	(0.33)	(0.11)	(0.18)
	PS	4.68	11.93	15.17	23.99	10.85	8.28	7.56
		(0.30)	(1.05)	(0.69)	(0.84)	(0.71)	(0.13)	(0.15)
	ME	0.40	0.80	0.51	0.69	0.55	0.63	0.55
		(0.02)	(0.04)	(0.02)	(0.03)	(0.02)	(0.03)	(0.02)
SCAD	FP	1.09	0.86	12.21	15.03	5.07	0.00	0.24
		(0.37)	(0.28)	(0.68)	(0.61)	(0.88)	(0.00)	(0.08)
	FN	5.86	8.34	6.25	5.84	7.10	8.94	6.90
		(0.17)	(0.13)	(0.16)	(0.18)	(0.20)	(0.05)	(0.16)
	PS	6.95	9.20	18.46	20.87	12.17	8.94	7.14
		(0.40)	(0.27)	(0.72)	(0.68)	(0.82)	(0.05)	(0.14)
	ME	0.47	0.61	0.66	0.71	0.57	0.61	0.66
		(0.02)	(0.03)	(0.03)	(0.03)	(0.03)	(0.02)	(0.04)
MCP	FP	0.96	0.61	12.04	15.38	6.92	0.00	0.97
		(0.35)	(0.24)	(0.63)	(0.66)	(0.82)	(0.00)	(0.17)
	FN	5.31	8.86	5.79	5.39	6.47	8.94	6.88
		(0.20)	(0.07)	(0.18)	(0.17)	(0.21)	(0.05)	(0.22)
	PS	6.27	9.47	17.83	20.77	13.39	8.94	7.85
		(0.39)	(0.23)	(0.67)	(0.68)	(0.74)	(0.05)	(0.17)
	ME	0.45	0.61	0.68	0.73	0.61	0.60	0.48
		(0.02)	(0.02)	(0.03)	(0.03)	(0.03)	(0.02)	(0.02)

Table 5.2: Simulation Results of Example 2 (high-dimension) over 100 replicates for the Lasso, the ALasso (Adaptive Lasso), the SCAD, the MCP (Standard Error in the parentheses).

		SPSP	2-CV	GCV	AIC	BIC	EBIC	Oracle
Lasso	FP	0.04	24.60	49.74	49.48	49.27	8.89	0.01
		(0.03)	(1.66)	(0.29)	(0.28)	(0.28)	(2.58)	(0.01)
	FN	2.11	1.97	1.68	1.68	1.68	2.00	2.01
		(0.06)	(0.02)	(0.08)	(0.08)	(0.08)	(0.03)	(0.01)
	PS	2.15	26.57	51.42	51.16	50.95	10.89	2.02
		(0.07)	(1.66)	(0.30)	(0.29)	(0.30)	(2.57)	(0.02)
	ME	1.54	1.21	1.35	1.35	1.35	2.09	2.54
		(0.07)	(0.07)	(0.05)	(0.05)	(0.05)	(0.15)	(0.17)
ALasso	FP	0.02	9.20	40.47	42.95	42.86	0.1	0.08
		(0.02)	(1.18)	(0.44)	(0.19)	(0.20)	(0.05)	(0.04)
	FN	1.05	1.05	0.24	0.22	0.22	2.1	1.92
		(0.13)	(0.12)	(0.06)	(0.06)	(0.06)	(0.06)	(0.05)
	PS	1.07	10.25	40.71	43.17	43.08	2.2	2.00
		(0.14)	(1.13)	(0.46)	(0.22)	(0.23)	(0.07)	(0.03)
	ME	1.00	1.14	1.31	1.33	1.33	1.8	1.82
		(0.09)	(0.07)	(0.05)	(0.05)	(0.05)	(0.12)	(0.11)
SCAD	FP	0.02	1.52	39.08	40.03	39.69	20.02	0.01
		(0.02)	(0.43)	(1.06)	(0.98)	(1.03)	(3.28)	(0.01)
	FN	3.42	3.89	3.68	3.68	3.68	3.81	2.46
		(0.11)	(0.05)	(0.09)	(0.09)	(0.09)	(0.08)	(0.09)
	PS	3.44	5.41	42.76	43.71	43.37	23.83	2.47
		(0.11)	(0.42)	(1.08)	(1.01)	(1.05)	(3.27)	(0.09)
	ME	2.27	2.48	1.36	1.36	1.36	2.07	6.40
		(0.09)	(0.11)	(0.05)	(0.05)	(0.05)	(0.13)	(0.38)
MCP	FP	0.01	0.85	46.47	46.54	46.48	42.74	0.03
		(0.01)	(0.24)	(0.67)	(0.54)	(0.55)	(2.03)	(0.02)
	FN	3.59	3.96	3.68	3.76	3.78	3.80	3.68
		(0.10)	(0.03)	(0.10)	(0.09)	(0.08)	(0.08)	(0.08)
	PS	3.60	4.81	50.15	50.30	50.26	46.54	3.71
		(0.10)	(0.25)	(0.68)	(0.56)	(0.58)	(2.03)	(0.08)
	ME	2.48	2.76	1.35	1.36	1.36	1.50	2.62
		(0.10)	(0.11)	(0.55)	(0.05)	(0.05)	(0.08)	(0.10)

Table 5.3: Simulation Results of Example 3 (high-dimension with large noise) over 100 replicates for the Lasso, the ALasso (Adaptive Lasso), the SCAD, the MCP (Standard Error in the parentheses).

		SPSP	2-CV	GCV	AIC	BIC	EBIC	Oracle
Lasso	FP	0.87	33.45	51.57	51.39	51.37	38.91	1.75
		(0.28)	(0.72)	(0.31)	(0.31)	(0.31)	(2.81)	(0.20)
	FN	2.96	2.25	1.96	1.95	1.95	2.13	2.31
		(0.13)	(0.10)	(0.10)	(0.10)	(0.10)	(0.10)	(0.09)
	PS	3.83	35.70	53.53	53.34	53.32	41.04	4.06
		(0.28)	(0.73)	(0.34)	(0.34)	(0.34)	(2.78)	(0.23)
	ME	0.39	0.66	1.02	1.02	1.02	0.94	0.51
		(0.04)	(0.03)	(0.03)	(0.03)	(0.03)	(0.04)	(0.03)
ALasso	FP	1.36	22.84	43.06	44.42	44.19	0.19	0.30
		(0.47)	(1.15)	(0.31)	(0.19)	(0.20)	(0.07)	(0.07)
	FN	2.61	1.87	1.21	1.11	1.10	3.02	2.51
		(0.15)	(0.13)	(0.13)	(0.12)	(0.12)	(0.10)	(0.10)
	PS	3.97	24.71	44.27	45.53	45.29	3.21	2.81
		(0.45)	(1.15)	(0.37)	(0.28)	(0.28)	(0.12)	(0.12)
	ME	0.39	0.58	1.01	1.02	1.02	0.53	0.44
		(0.05)	(0.03)	(0.03)	(0.03)	(0.03)	(0.05)	(0.04)
SCAD	FP	1.37	3.36	39.23	40.04	39.74	21.81	0.02
		(0.57)	(0.54)	(1.12)	(1.06)	(1.09)	(3.28)	(0.02)
	FN	3.66	3.91	3.94	3.94	3.94	3.99	2.76
		(0.08)	(0.04)	(0.04)	(0.04)	(0.04)	(0.04)	(0.10)
	PS	5.03	7.27	43.17	43.98	43.68	25.80	2.78
		(0.58)	(0.55)	(1.12)	(1.06)	(1.09)	(3.27)	(0.10)
	ME	0.40	0.42	1.02	1.02	1.02	0.74	0.88
		(0.03)	(0.03)	(0.03)	(0.03)	(0.03)	(0.07)	(0.05)
MCP	FP	1.69	1.90	45.37	45.93	45.81	39.64	0.02
		(0.70)	(0.43)	(0.90)	(0.79)	(0.85)	(2.56)	(0.02)
	FN	3.95	4.02	3.93	3.94	3.94	3.96	3.96
		(0.03)	(0.03)	(0.04)	(0.03)	(0.03)	(0.03)	(0.03)
	PS	5.64	5.92	49.30	49.87	49.75	43.60	3.98
		(0.70)	(0.43)	(0.90)	(0.80)	(0.86)	(2.55)	(0.02)
	ME	0.45	0.46	1.02	1.02	1.02	0.94	0.46
		(0.03)	(0.04)	(0.03)	(0.03)	(0.03)	(0.04)	(0.03)

Table 5.4: Simulation Results of Example 4 (high-dimensional sparse model, $p = 100$) over 100 replicates for the Lasso, the ALasso (Adaptive Lasso), the SCAD, the MCP (Standard Error in the parentheses).

		SPSP	2-CV	GCV	AIC	BIC	EBIC	Oracle
Lasso	FP	0.23	22.04	49.82	49.87	49.87	11.72	0.37
		(0.08)	(1.49)	(0.29)	(0.26)	(0.26)	(2.88)	(0.08)
	FN	0.34	0.00	0.02	0.02	0.02	0.01	0.00
		(0.09)	(0.00)	(0.02)	(0.02)	(0.02)	(0.01)	(0.00)
	PS	0.57	22.04	49.84	49.89	49.89	11.73	0.37
		(0.11)	(1.49)	(0.30)	(0.26)	(0.26)	(2.88)	(0.08)
	ME	0.57	0.43	1.02	1.02	1.02	0.62	0.50
		(0.06)	(0.04)	(0.03)	(0.03)	(0.03)	(0.05)	(0.04)
ALasso	FP	0.86	13.28	42.95	45.55	45.46	0.16	0.10
		(0.25)	(1.28)	(0.78)	(0.15)	(0.15)	(0.06)	(0.04)
	FN	0.21	0.00	0.01	0.00	0.01	0.10	0.00
		(0.07)	(0.00)	(0.01)	(0.00)	(0.01)	(0.04)	(0.00)
	PS	1.07	13.28	42.96	45.55	45.47	0.26	0.10
		(0.25)	(1.28)	(0.78)	(0.15)	(0.15)	(0.07)	(0.04)
	ME	0.23	0.31	0.98	1.01	1.01	0.23	0.24
		(0.04)	(0.03)	(0.03)	(0.03)	(0.03)	(0.03)	(0.03)
SCAD	FP	2.85	2.84	31.98	33.41	32.58	9.68	0.11
		(0.79)	(0.45)	(1.23)	(1.14)	(1.23)	(2.59)	(0.04)
	FN	0.32	0.24	0.11	0.11	0.11	0.37	0.13
		(0.08)	(0.06)	(0.04)	(0.04)	(0.04)	(0.07)	(0.05)
	PS	3.17	3.08	32.09	33.52	32.69	10.05	0.24
		(0.77)	(0.44)	(1.22)	(1.14)	(1.23)	(2.56)	(0.06)
	ME	0.42	0.35	1.00	1.00	1.00	0.53	0.52
		(0.06)	(0.05)	(0.03)	(0.03)	(0.03)	(0.05)	(0.06)
MCP	FP	1.38	1.47	42.56	43.94	43.84	29.13	0.16
		(0.43)	(0.41)	(0.94)	(0.78)	(0.80)	(3.24)	(0.05)
	FN	0.37	0.40	0.12	0.12	0.12	0.25	0.13
		(0.08)	(0.07)	(0.05)	(0.05)	(0.05)	(0.06)	(0.05)
	PS	1.75	1.87	42.68	44.06	43.96	29.38	0.29
		(0.42)	(0.41)	(0.93)	(0.77)	(0.79)	(3.21)	(0.07)
	ME	0.45	0.46	1.02	1.02	1.02	0.94	0.46
		(0.06)	(0.06)	(0.03)	(0.03)	(0.03)	(0.06)	(0.04)

Table 5.5: Simulation Results of Example 5 (high-dimensional sparse model, $p = 1000$) over 100 replicates for the Lasso, the ALasso (Adaptive Lasso), the SCAD, the MCP (Standard Error in the parentheses).

		SPSP	2-CV	GCV	AIC	BIC	EBIC	Oracle
Lasso	FP	0.04	41.55	56.85	56.56	56.03	0.11	0.14
		(0.03)	(2.29)	(0.42)	(0.41)	(0.39)	(0.04)	(0.05)
	FN	0.75	0.00	0.00	0.00	0.00	0.69	0.07
		(0.12)	(0.00)	(0.00)	(0.00)	(0.00)	(0.14)	(0.04)
	PS	0.79	41.55	56.85	56.56	56.03	0.80	0.21
		(0.12)	(2.29)	(0.42)	(0.41)	(0.39)	(0.13)	(0.06)
	ME	0.61	0.82	1.05	1.05	1.05	1.50	1.09
		(0.09)	(0.05)	(0.03)	(0.03)	(0.03)	(0.17)	(0.08)
ALasso	FP	0.13	20.08	41.70	44.78	44.78	0.11	0.09
		(0.06)	(1.92)	(0.87)	(0.29)	(0.29)	(0.04)	(0.04)
	FN	0.57	0.01	0.00	0.00	0.00	0.51	0.12
		(0.11)	(0.01)	(0.00)	(0.00)	(0.00)	(0.12)	(0.05)
	PS	0.70	20.09	41.70	44.78	44.78	0.62	0.21
		(0.11)	(1.92)	(0.87)	(0.29)	(0.29)	(0.12)	(0.06)
	ME	0.49	0.53	0.98	1.02	1.02	0.79	0.61
		(0.08)	(0.04)	(0.04)	(0.03)	(0.03)	(0.13)	(0.06)
SCAD	FP	1.95	6.20	29.56	31.61	30.70	15.45	0.07
		(0.43)	(0.77)	(0.60)	(0.70)	(0.69)	(2.19)	(0.04)
	FN	0.50	0.27	0.37	0.37	0.37	0.59	0.13
		(0.09)	(0.06)	(0.07)	(0.07)	(0.07)	(0.12)	(0.05)
	PS	2.45	6.47	29.93	31.98	31.07	16.04	0.20
		(0.42)	(0.77)	(0.60)	(0.70)	(0.69)	(2.16)	(0.06)
	ME	0.58	0.57	1.04	1.04	1.04	1.18	0.95
		(0.06)	(0.07)	(0.03)	(0.03)	(0.03)	(0.15)	(0.09)
MCP	FP	0.66	2.25	46.06	46.07	46.02	44.81	0.03
		(0.25)	(0.46)	(0.54)	(0.49)	(0.53)	(1.31)	(0.02)
	FN	0.69	0.44	0.30	0.30	0.30	0.30	0.30
		(0.10)	(0.08)	(0.07)	(0.07)	(0.07)	(0.07)	(0.07)
	PS	1.35	2.69	46.36	46.37	46.32	45.11	0.33
		(0.25)	(0.46)	(0.54)	(0.50)	(0.53)	(1.30)	(0.07)
	ME	0.58	0.58	1.04	1.04	1.04	1.03	0.48
		(0.07)	(0.06)	(0.03)	(0.03)	(0.03)	(0.03)	(0.04)

Table 5.6: Simulation Results of Example 6 (misspecified model) over 100 replicates for the Lasso, the ALasso (Adaptive Lasso), the SCAD, the MCP (Standard Error in the parentheses).

		SPSP	2-CV	GCV	AIC	BIC	EBIC	Oracle
Lasso	FP	0.39	27.20	44.72	45.47	45.47	23.35	0.66
		(0.13)	(1.65)	(0.34)	(0.17)	(0.17)	(3.14)	(0.13)
	FN	0.95	0.06	0.03	0.03	0.03	0.83	0.38
		(0.16)	(0.06)	(0.02)	(0.02)	(0.02)	(0.21)	(0.09)
	PS	1.34	27.76	44.75	45.50	44.50	24.18	1.04
		(0.18)	(1.64)	(0.35)	(0.17)	(0.17)	(3.02)	(0.14)
	ME	14.66	13.71	18.77	18.88	18.88	27.21	28.91
		(1.97)	(1.16)	(1.04)	(1.04)	(1.04)	(2.75)	(1.81)
ALasso	FP	0.67	19.09	36.91	42.18	42.15	0.30	0.37
		(0.17)	(1.59)	(1.14)	(0.25)	(0.25)	(0.08)	(0.08)
	FN	0.77	0.20	0.11	0.08	0.08	1.55	0.36
		(0.14)	(0.08)	(0.04)	(0.04)	(0.04)	(0.26)	(0.08)
	PS	1.44	19.29	37.02	42.26	42.23	1.85	0.73
		(0.19)	(1.57)	(1.14)	(0.26)	(0.26)	(0.24)	(0.12)
	ME	12.94	12.86	17.21	18.29	18.29	26.99	16.14
		(1.78)	(1.32)	(1.03)	(1.02)	(1.02)	(3.70)	(1.28)
SCAD	FP	5.28	4.00	37.86	38.67	38.47	25.71	0.45
		(1.17)	(0.73)	(1.07)	(0.97)	(1.01)	(3.01)	(0.10)
	FN	0.48	0.44	0.14	0.14	0.14	0.58	0.16
		(0.14)	(0.15)	(0.06)	(0.06)	(0.06)	(0.19)	(0.06)
	PS	5.76	4.44	38.00	38.81	38.61	26.29	0.61
		(1.15)	(0.71)	(1.08)	(0.98)	(1.01)	(2.94)	(0.11)
	ME	14.77	18.13	18.82	18.84	18.83	22.65	16.68
		(2.13)	(2.36)	(1.03)	(1.03)	(1.03)	(3.33)	(1.70)
MCP	FP	2.50	1.96	44.00	44.30	44.11	40.96	0.16
		(0.75)	(0.46)	(0.58)	(0.43)	(0.56)	(1.84)	(0.07)
	FN	0.57	0.65	0.11	0.11	0.11	0.23	0.15
		(0.15)	(0.16)	(0.05)	(0.05)	(0.05)	(0.11)	(0.07)
	PS	3.07	2.61	44.11	44.41	44.22	41.19	0.31
		(0.74)	(0.44)	(0.58)	(0.44)	(0.57)	(1.79)	(0.09)
	ME	13.96	16.97	18.90	18.93	18.90	19.37	10.69
		(2.23)	(2.46)	(1.04)	(1.04)	(1.04)	(1.61)	(1.16)

Table 5.4-5.5 presents the simulation results for Example 4 and Example 5. Note that in these two examples, the true models are highly sparse and the relevant variables and the irrelevant variables are all correlated. We observe that the SPSP procedure can remarkably improves the selection accuracy by selecting fewer irrelevant variables on the existing penalties in general. From these two tables, we also notice that the adaptive Lasso with the EBIC also has a competitive performance in selection due to the over-sparsity patterns of the true models but the model errors are much larger than those of the SPSP approaches especially in Example 5.

The results of Example 6 are shown in Table 5.6. We observe that for this example where models are misspecified, the SPSP algorithm on all the four penalties all have a good performance in both selection and estimation accuracy. Note that the adaptive Lasso with the EBIC, the SCAD with CV, the MCP with CV all perform similarly with SPSP in terms of FP values. It is worth mentioning that the ME values are all relatively large since we divide σ^2 in computing ME and the noise level $\sigma = 0.25$ is quite small in this simulation study.

To show the overall patterns of the results over these 100 replicates, we also report the box plots of the "PS" values of all the methods for these six examples in Figure 5.1-5.6. It is observed that the SPSP algorithm yields smaller medians of "PS" values than the other methods almost across all cases . In addition, we notice that the SPSP algorithm delivers some outliers of "PS" values for some simulations, especially as shown in Figure 5.3 and 5.4. This is because the SPSP algorithm may select some irrelevant variables at the beginning of the process in these simulations, which leads to high "PS" values. As discussed before, we may need a conservative modification to control the number of the selected irrelevant variables at the beginning of the SPSP algorithm.

In summary, from these simulation examples, we can see that the SPSP approach can improve the selection accuracy of the penalized least squares estimations in general, especially when high correlations among the variables are presented. It can well balance the trade-off between the model fitting and the model sparsity compared with the other criteria. Compared with the other criteria, the proposed algorithm can select fewer irrelevant variables without excluding more relevant variables for high dimensional data problems.

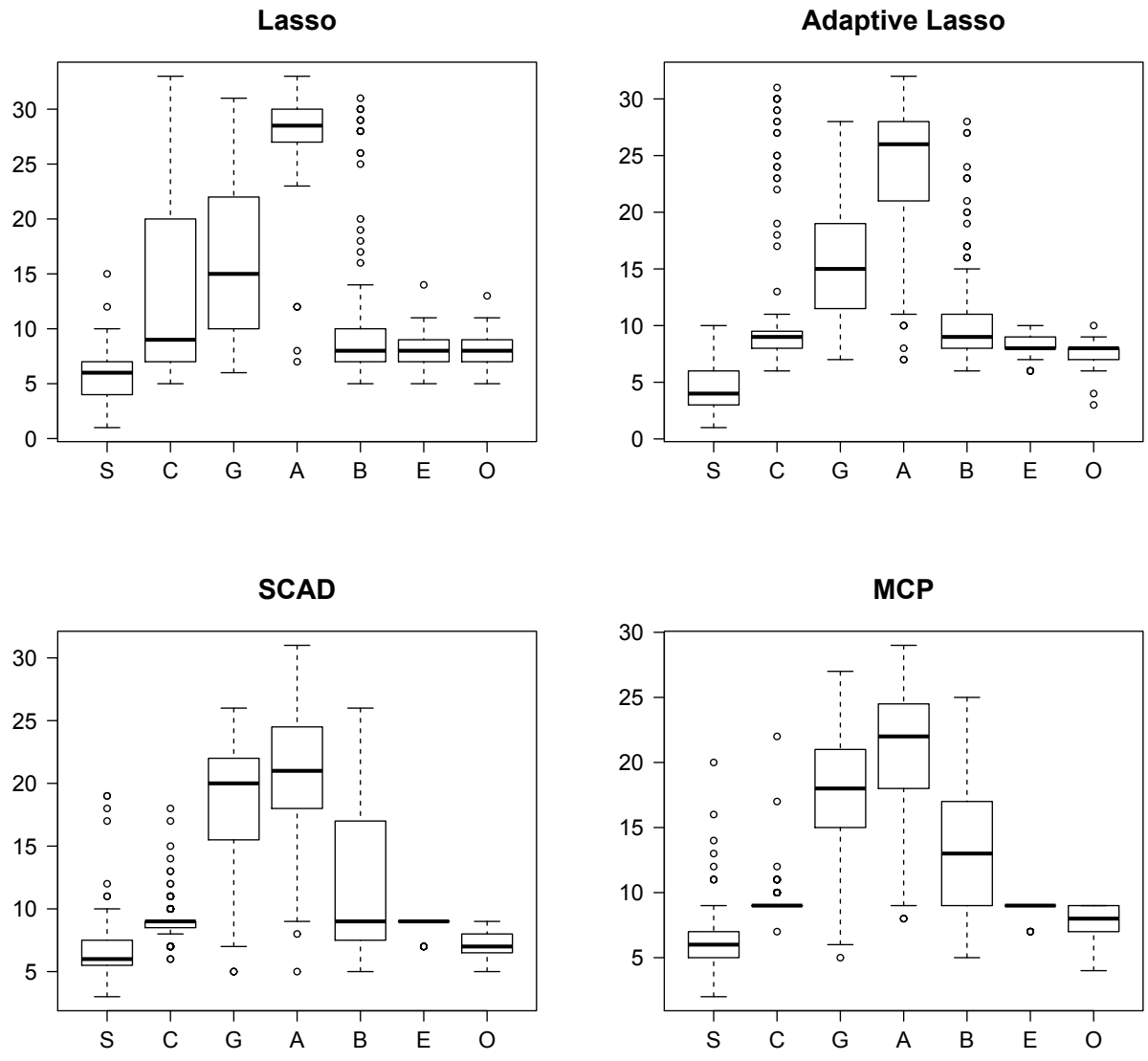

Figure 5.1: The box plots of PS values for Example 1: SPSP(S), 2-CV(C), GCV(G), AIC(A), BIC(B), EBIC(E), Oracle(O).

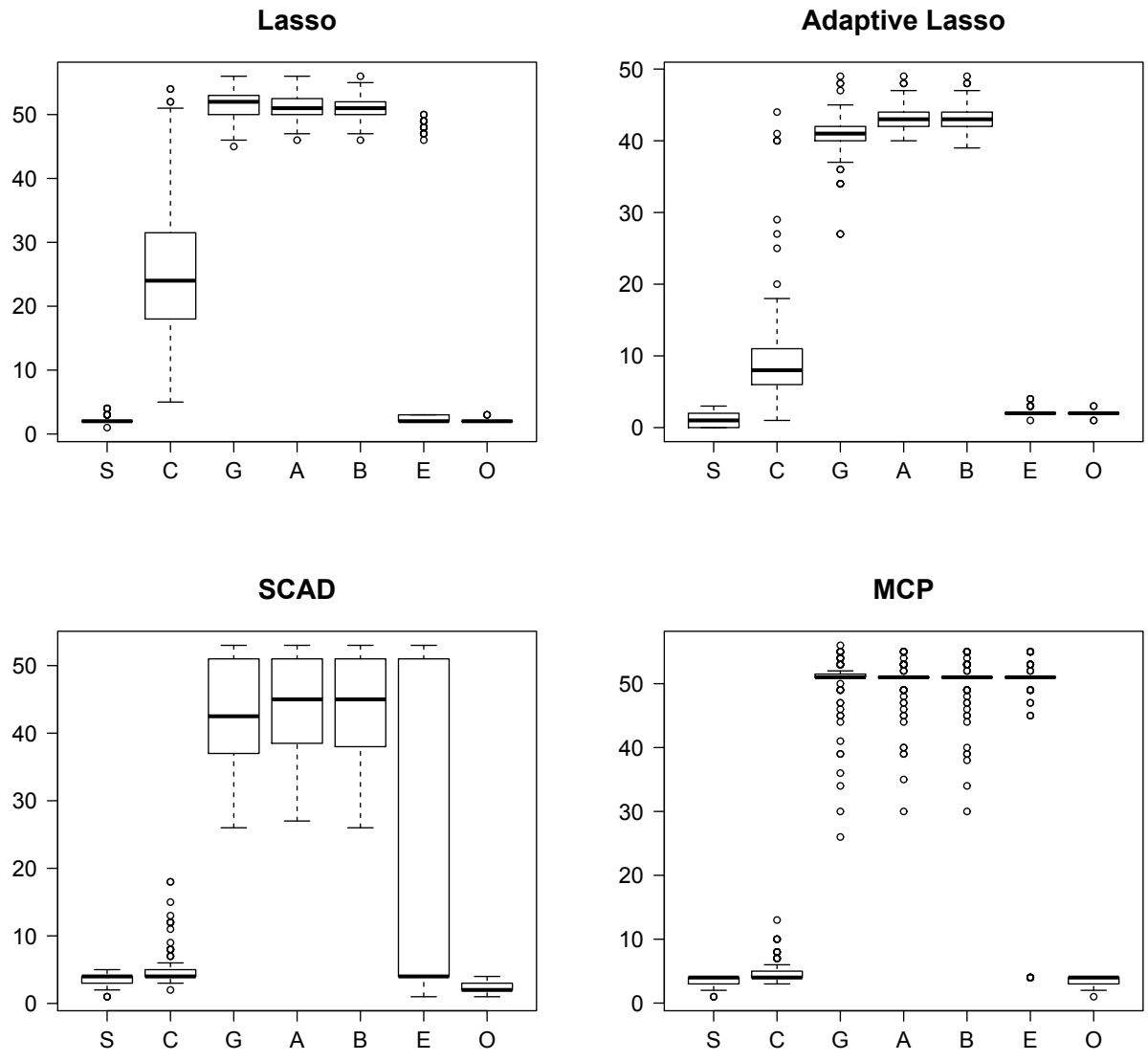

Figure 5.2: The box plots of PS values for Example 2: SPSP(S), 2-CV(C), GCV(G), AIC(A), BIC(B), EBIC(E), Oracle(O).

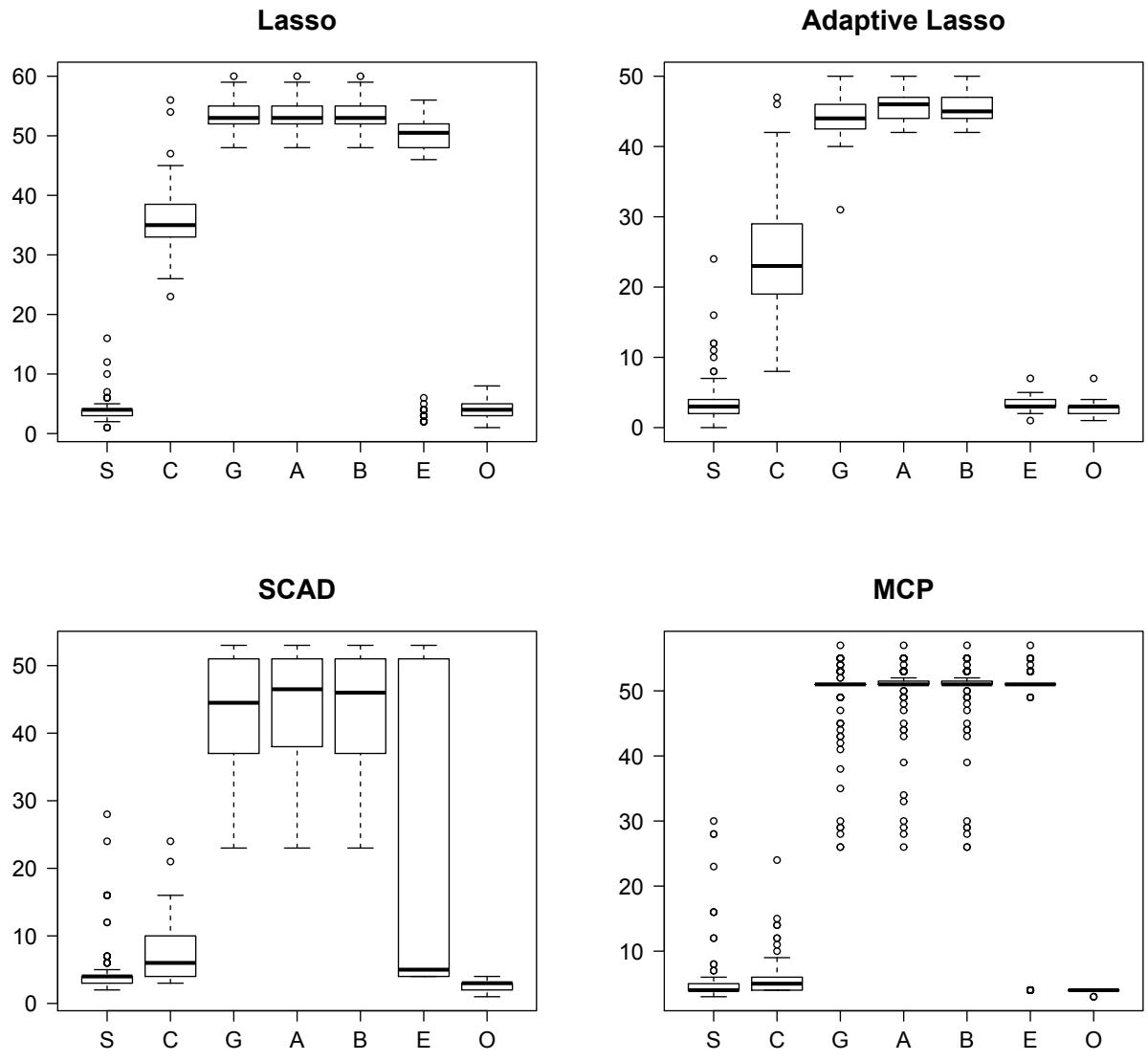

Figure 5.3: The box plots of PS values for Example 3: SPSP(S), 2-CV(C), GCV(G), AIC(A), BIC(B), EBIC(E), Oracle(O).

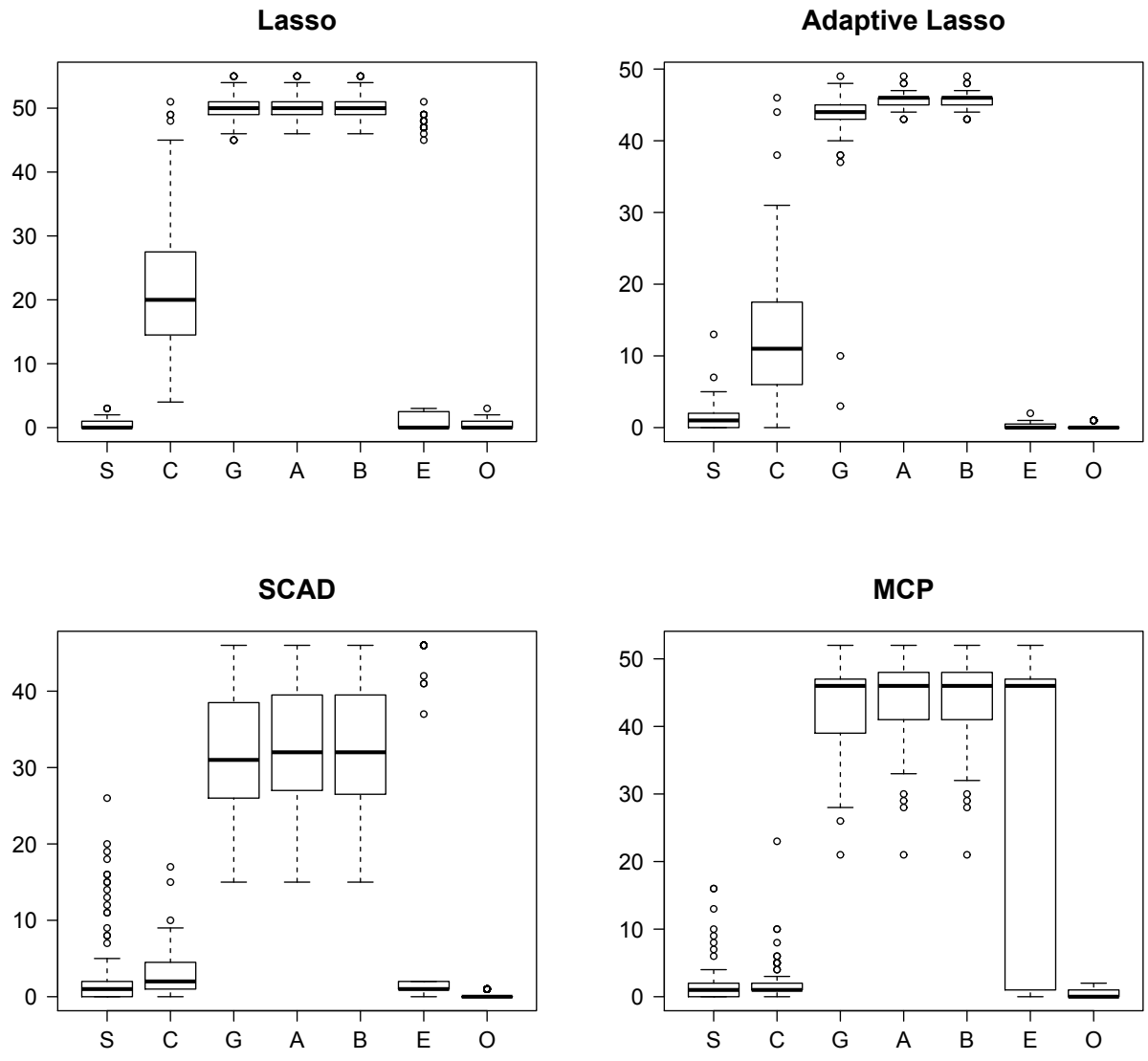

Figure 5.4: The box plots of PS values for Example 4: SPSP(S), 2-CV(C), GCV(G), AIC(A), BIC(B), EBIC(E), Oracle(O).

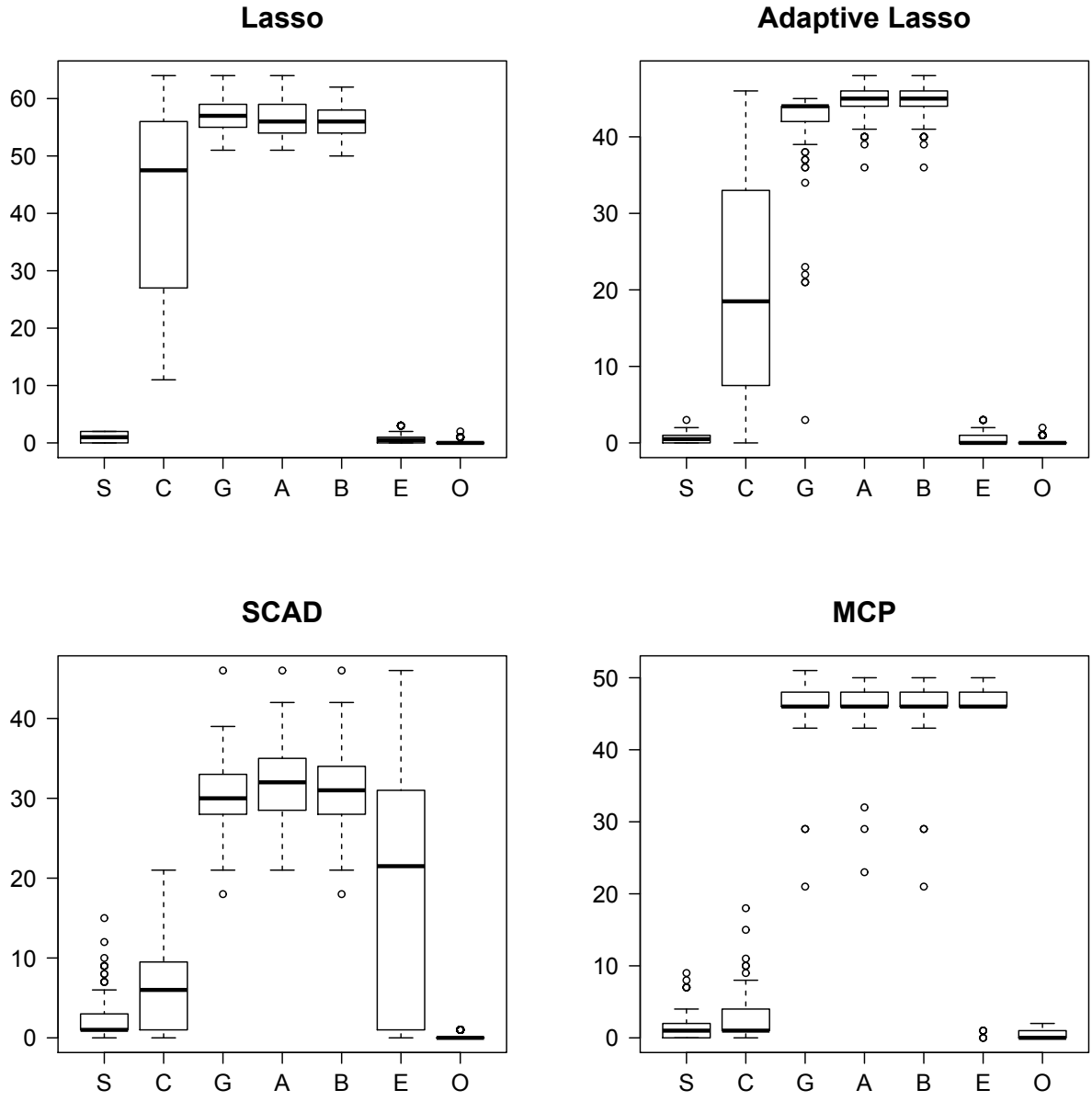

Figure 5.5: The box plots of PS values for Example 5: SPSP(S), 2-CV(C), GCV(G), AIC(A), BIC(B), EBIC(E), Oracle(O).

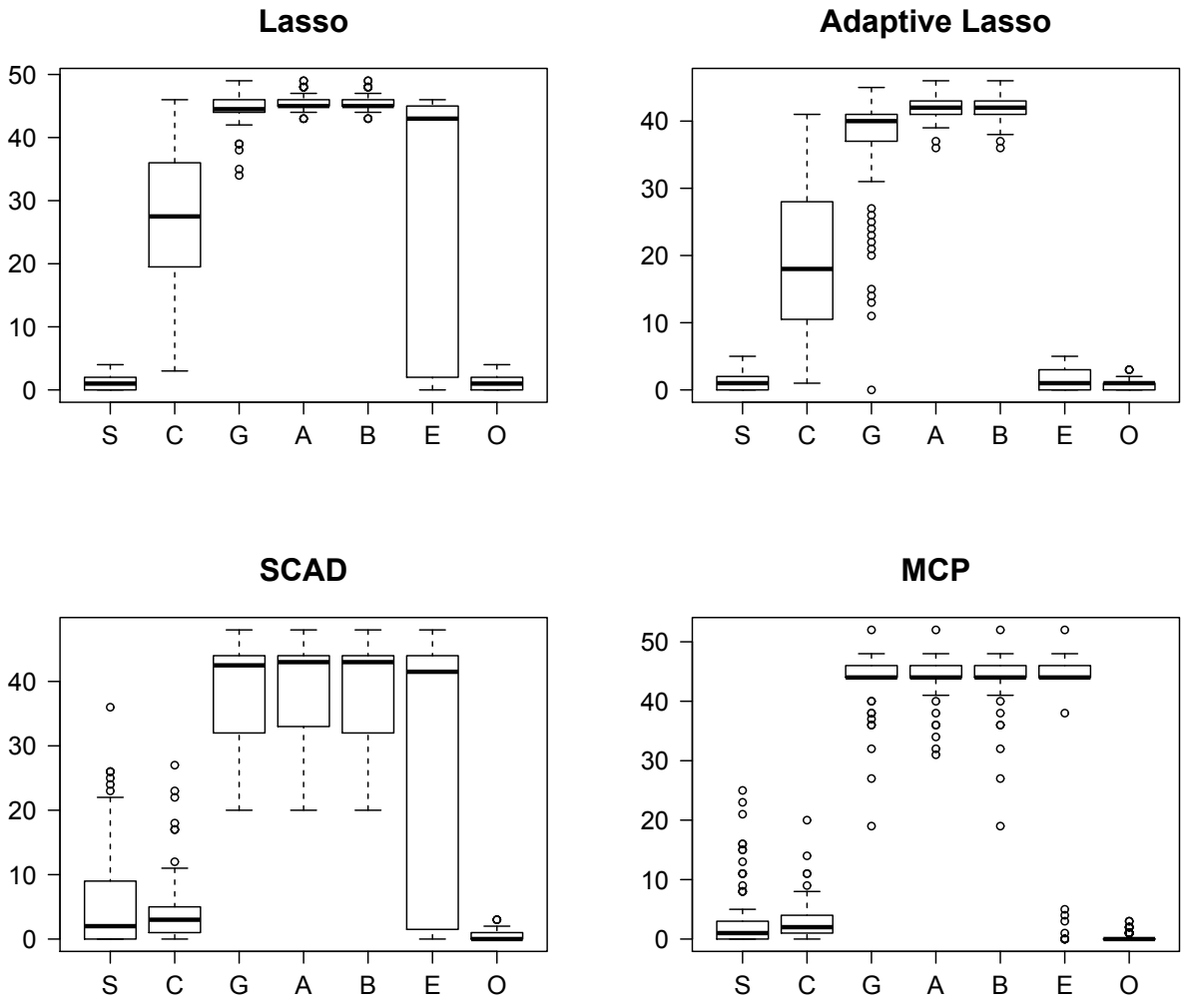

Figure 5.6: The box plots of PS values for Example 6: SPSP(S), 2-CV(C), GCV(G), AIC(A), BIC(B), EBIC(E), Oracle(O).

5.1.3 The SPSP Algorithm on the Ridge

One compelling advantage of the proposed SPSP procedure is that it can be applied for the penalties which cannot produce sparse solutions, such as the ridge penalty. Here we also implement the SPSP algorithm on the ridge for these examples. The results are presented in Table 5.7. For the convenience of comparison, we also report the ranking of the performance of the ridge among all the five penalties (the Lasso, the adaptive Lasso, the SCAD, the MCP and the ridge) with the SPSP procedure. For instance, if rank equals 3, it means this result of ridge with SPSP is the third smallest among these five penalties. In two of the examples, the ridge actually returns the best results in terms of selection accuracy. Although the ME numbers of ridge are generally large compared to other penalties, the differences between the ME values of ridge and those of other penalties are relatively small.

Table 5.7: Simulation results of the SPSP approach on the ridge (Standard Error in parentheses), ranking among all the five penalties in the third row.

		Ex(1)	Ex(2)	Ex(3)	Ex(4)	Ex(5)	Ex(6)
Ridge	FP	4.32	0.52	2.33	0.36	0.18	0.37
		(0.81)	(0.16)	(0.61)	(0.11)	(0.07)	(0.11)
		5	5	5	2	3	1
	FN	3.38	0.48	1.50	0.61	0.96	2.19
		(0.32)	(0.10)	(0.21)	(0.10)	(0.10)	(0.18)
		1	1	1	5	5	5
	PS	7.70	1.00	3.83	0.97	1.14	2.56
		(0.70)	(0.19)	(0.60)	(0.12)	(0.11)	(0.15)
		5	1	1	2	3	3
	ME	0.49	0.77	0.53	0.55	0.86	31.33
		(0.03)	(0.08)	(0.08)	(0.08)	(0.08)	(2.77)
		5	1	5	4	5	5

5.1.4 The Comparison with the Stability Selection

As introduced in Chapter 1, the stability selection (SS) approach, proposed by Meinshausen and Bühhmann (2010, [41]), also avoids the selection issue of the tuning parameter by using the subsampling techniques. Hence, we compare the performance of the SPSP procedure with the SS

approach for the first four examples and the last example. Here we omit Example 5 because of the huge computational cost for stability selection. As suggested in Meinshausen and Bühlmann (2010, [41]), we evaluate the selection probabilities over 100 subsamples and choose the threshold value as 0.9. The results of the SS algorithm can be found in Table 5.8 and Table 5.9. It can be seen that generally the SPSP algorithm has better performance than SS in terms od both selection accuracy and model errors. Particularly, the SS approach tends to exclude many relevant variables. In addition, we also notice that the computational cost of the SS algorithm is dramatically higher than that of the SPSP procedure since its process involves resampling and solution paths for all the 100 subsamples to be computed.

Table 5.8: Results of the SS algorithm and the SPSP on the Lasso.

		Ex(1)	Ex(2)	Ex(3)	Ex(4)	Ex(6)
SS	FP	0.35	0.01	0.01	0.10	0.00
		(0.64)	(0.10)	(0.10)	(0.32)	(0.00)
	FN	9.47	2.37	4.79	0.30	2.30
		(0.56)	(0.54)	(0.73)	(0.48)	(0.82)
	PS	9.82	2.38	4.80	0.40	2.30
		(0.89)	(0.56)	(0.74)	(0.52)	(0.82)
	ME	1.93	1.68	1.64	0.26	28.02
		(1.31)	(0.57)	(0.97)	(0.34)	(9.93)
SPSP	FP	0.63	0.87	0.01	0.23	0.39
		(0.24)	(0.03)	(0.28)	(0.08)	(0.13)
	FN	5.11	2.11	2.96	0.34	0.95
		(0.28)	(0.54)	(0.09)	(0.48)	(0.16)
	PS	5.74	2.15	3.83	0.57	1.34
		(0.33)	(0.07)	(0.11)	(0.52)	(0.18)
	ME	0.44	1.54	0.39	0.57	14.66
		(0.02)	(0.07)	(0.04)	0.06	(1.97)

Table 5.9: The average time for computing the SS and the SPSP estimators (in seconds)

	Ex(1)	Ex(2)	Ex(3)	Ex(4)	Ex(6)
SS	41.23s	88.79s	121.16s	199.29s	162.39s
SPSP	0.44s	1.55s	0.39s	3.89s	2.90s

5.2 Ranking by the AIS

In this section, we will investigate the performance of the proposed AIS algorithm and compare the results with the other stepwise selection methods: the forward selection, the Least Angle Regression (LARS) algorithm ([15]) and the stepwise selections by the original penalized least squares estimators..

In details, the AIS algorithm is based on the results from the SPSP procedure. We will compute the AIS on the Lasso, the adaptive Lasso, the ridge, the SCAD and the MCP for all these six examples. The forward selection and the LARS algorithm are implemented by the $\mathbf{R}$ package *lars* ([29]). The stepwise selections by the original estimators are ranked by the order of the tuning parameters at which the estimators of their coefficients are shrunk to zeros.

For all the examples, we compare the False Positive Rate (FPR) of the AIS method with the other two methods at each step. Particularly, in these stepwise selection methods, we select q variables after q steps, $q = 1, \cdots, p$, then FPR is computed as

$$FPR = \frac{FP(q)}{q},$$

where $FP(q)$ =the number of irrelevant variables in these q variables. Clearly, a higher FPR of one method indicates that the method mistakenly choose more irrelevant variables at this stage.

The average FPRs of all the methods over 100 replicates are shown in the Figure 5.7-5.12.

Figure 5.7 shows that the FPRs of the AIS on all the penalties are smaller than the other methods for Example 1. Even at the beginning, the FPR curves of the AIS methods are much lower than the other FPR curves. For Example 2 and Example 3, the AIS approaches and the LARS algorithm are remarkably better than the Forward Selection. It is also noticed that the AIS on the ridge penalty have the best performance for most steps in these two examples. Among these three examples, we also notice that the results of AIS algorithm on each penalty are much better than those of the original penalty, which suggests that the AIS algorithm can improve the selection accuracy of the

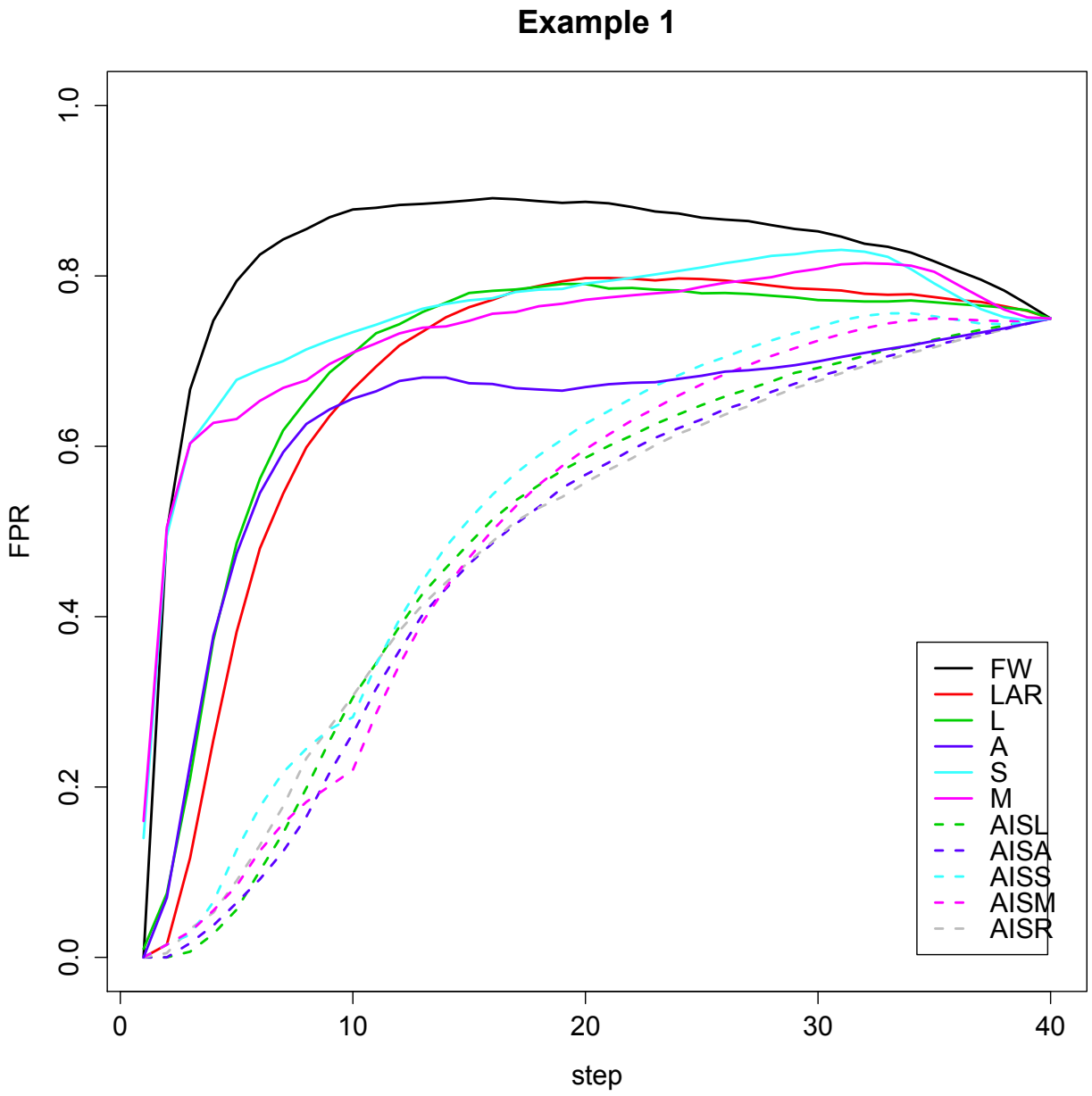

Figure 5.7: The mean of FPRs over 100 replicates for Example 1: the Forward Selection (FW), the Least Angle Regression (LAR), the original Lasso (L), the original adaptive Lasso (A), the original SCAD (S), the original MCP (M), the AIS on the Lasso (AISL), the AIS on the adaptive Lasso (AISA), the AIS on the SCAD (AISS), the AIS on the MCP (AISM), the AIS on the ridge (AISR).

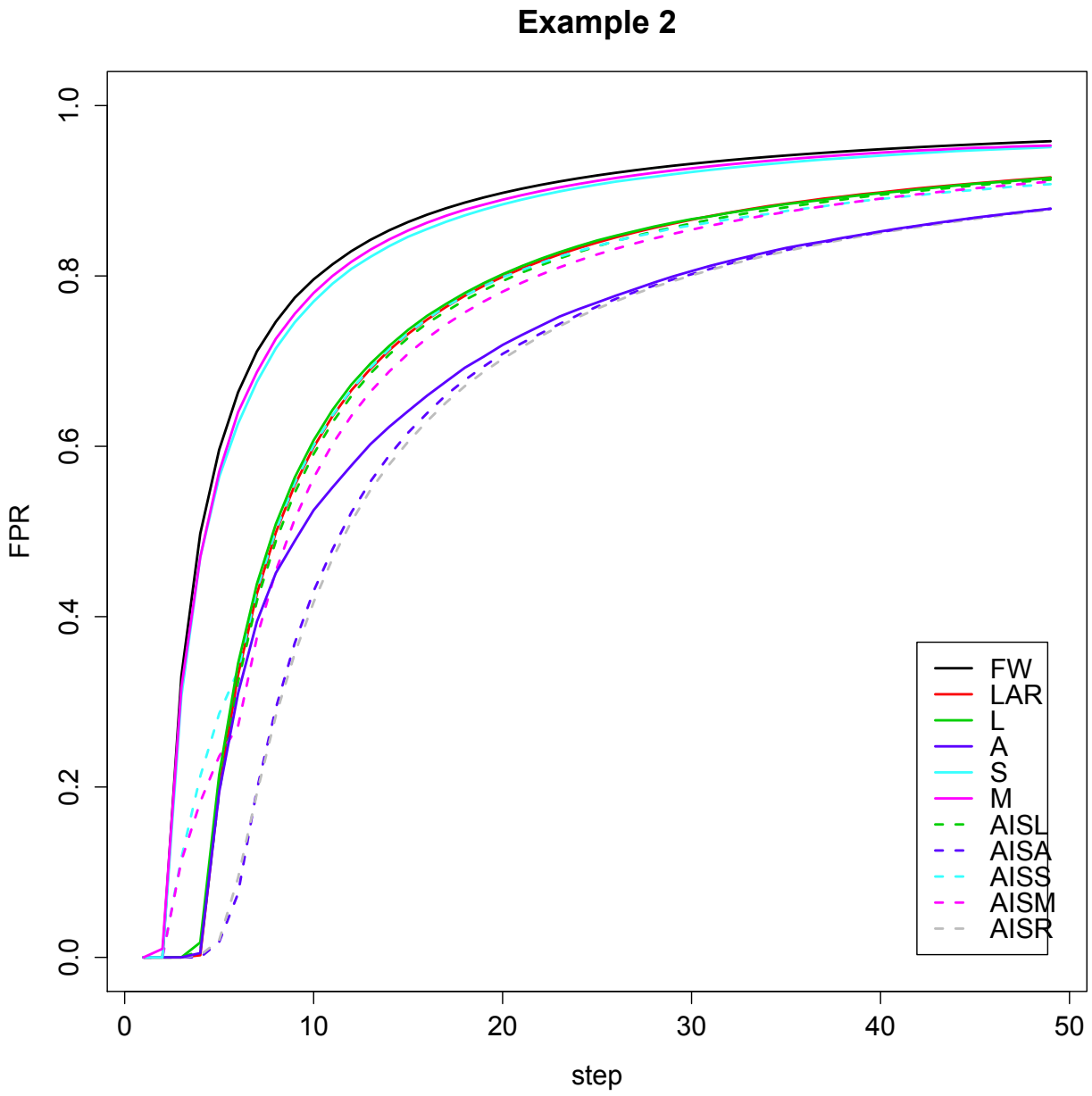

Figure 5.8: The mean of FPRs over 100 replicates for Example 2: the Forward Selection (FW), the Least Angle Regression (LAR), the original Lasso (L), the original adaptive Lasso (A), the original SCAD (S), the original MCP (M), the AIS on the Lasso (AISL), the AIS on the adaptive Lasso (AISA), the AIS on the SCAD (AISS), the AIS on the MCP (AISM), the AIS on the ridge (AISR).

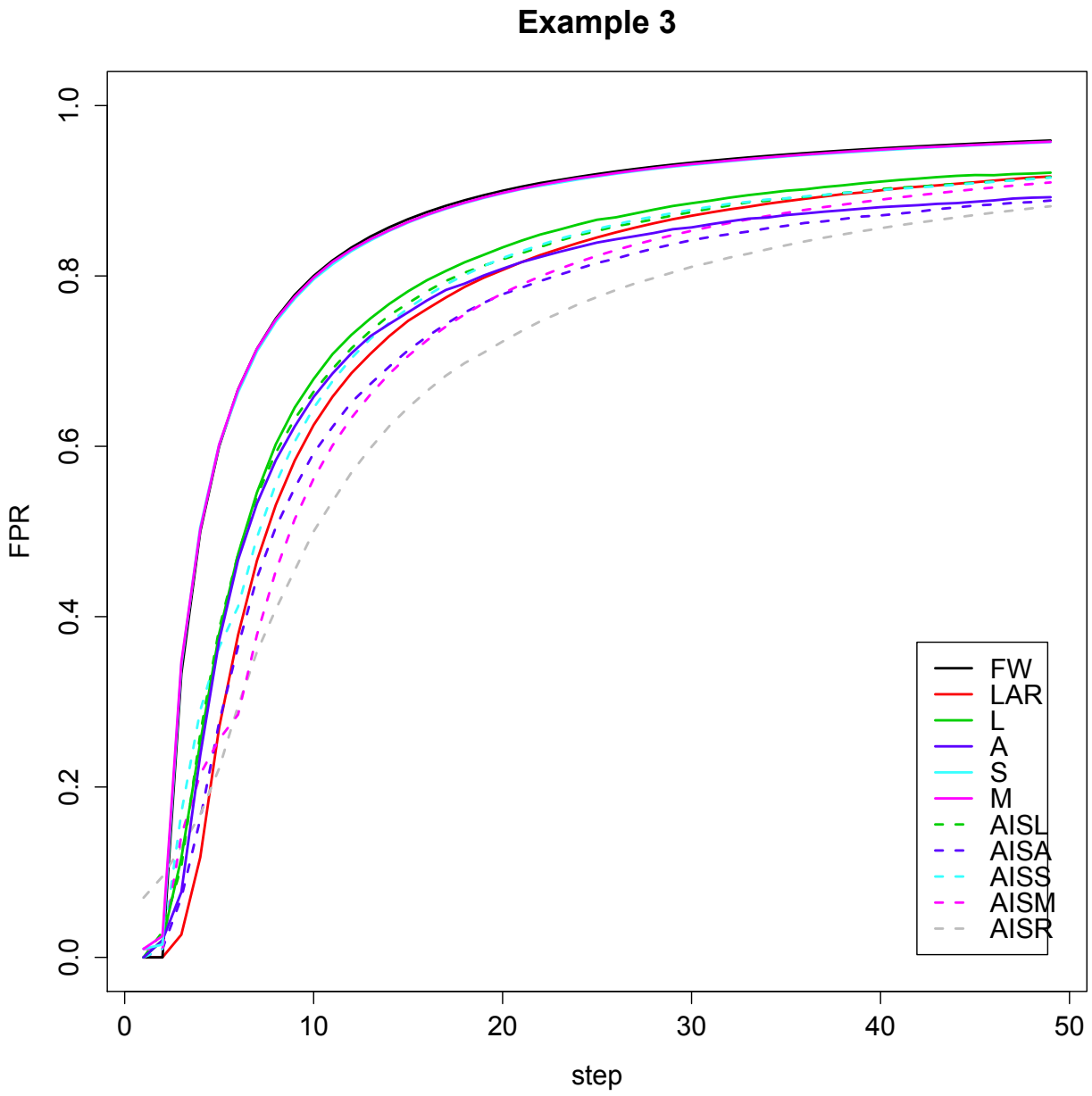

Figure 5.9: The mean of FPRs over 100 replicates for Example 3: the Forward Selection (FW), the Least Angle Regression (LAR), the original Lasso (L), the original adaptive Lasso (A), the original SCAD (S), the original MCP (M), the AIS on the Lasso (AISL), the AIS on the adaptive Lasso (AISA), the AIS on the SCAD (AISS), the AIS on the MCP (AISM), the AIS on the ridge (AISR).

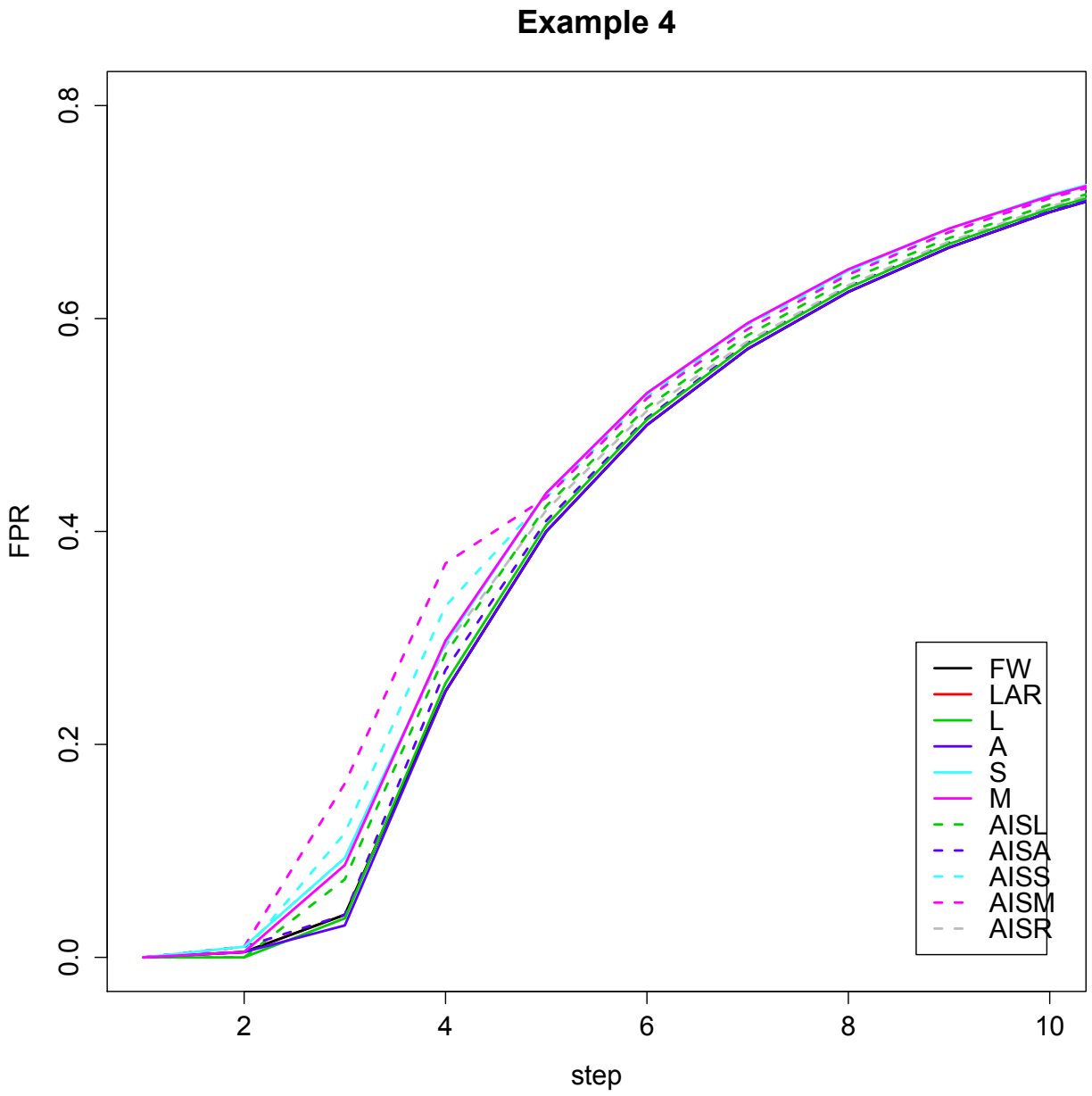

Figure 5.10: The mean of FPRs over 100 replicates for Example 4: the Forward Selection (FW), the Least Angle Regression (LAR), the original Lasso (L), the original adaptive Lasso (A), the original SCAD (S), the original MCP (M), the AIS on the Lasso (AISL), the AIS on the adaptive Lasso (AISA), the AIS on the SCAD (AISS), the AIS on the MCP (AISM), the AIS on the ridge (AISR).

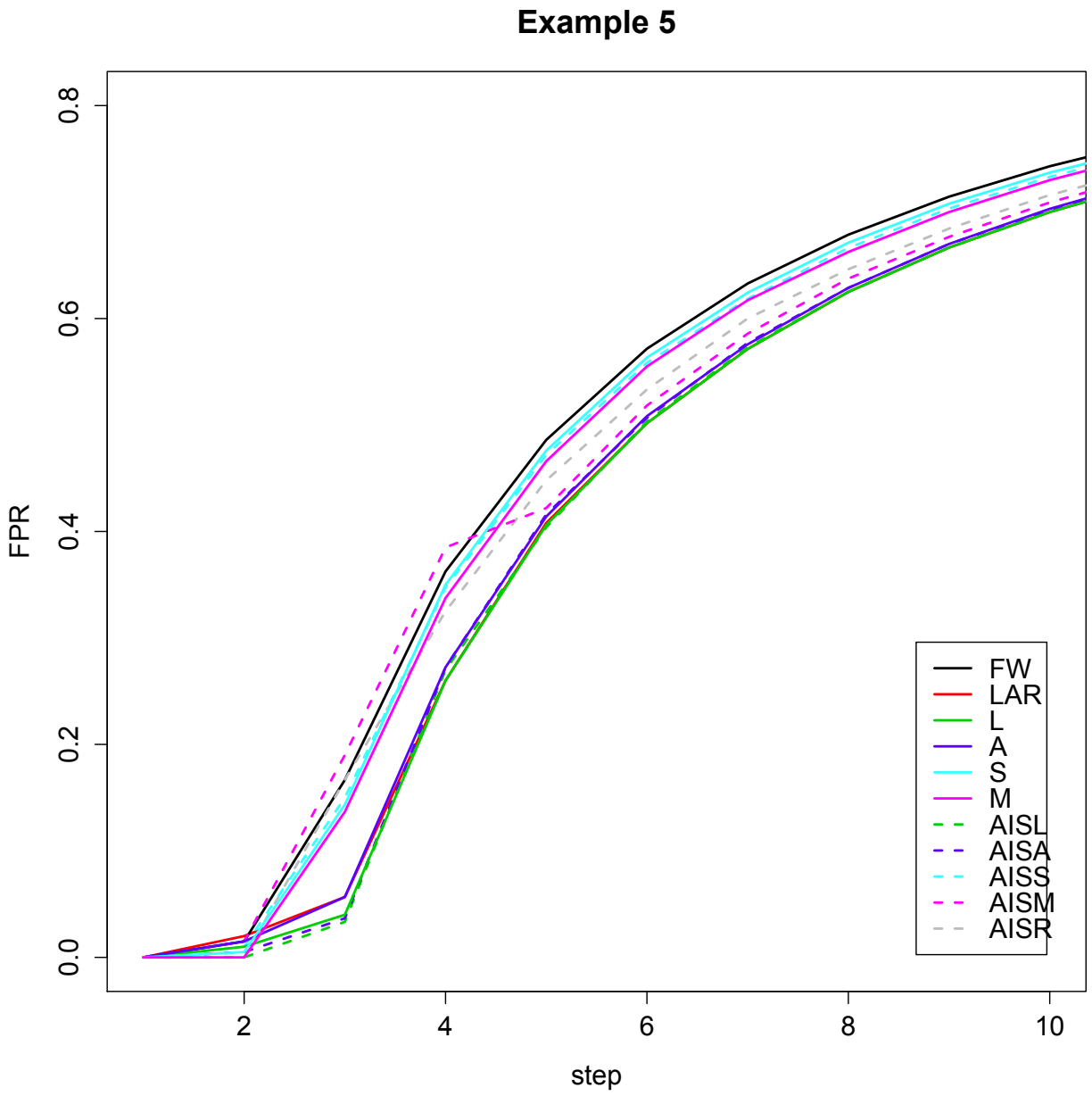

Figure 5.11: The mean of FPRs over 100 replicates for Example 5: the Forward Selection (FW), the Least Angle Regression (LAR), the original Lasso (L), the original adaptive Lasso (A), the original SCAD (S), the original MCP (M), the AIS on the Lasso (AISL), the AIS on the adaptive Lasso (AISA), the AIS on the SCAD (AISS), the AIS on the MCP (AISM), the AIS on the ridge (AISR).

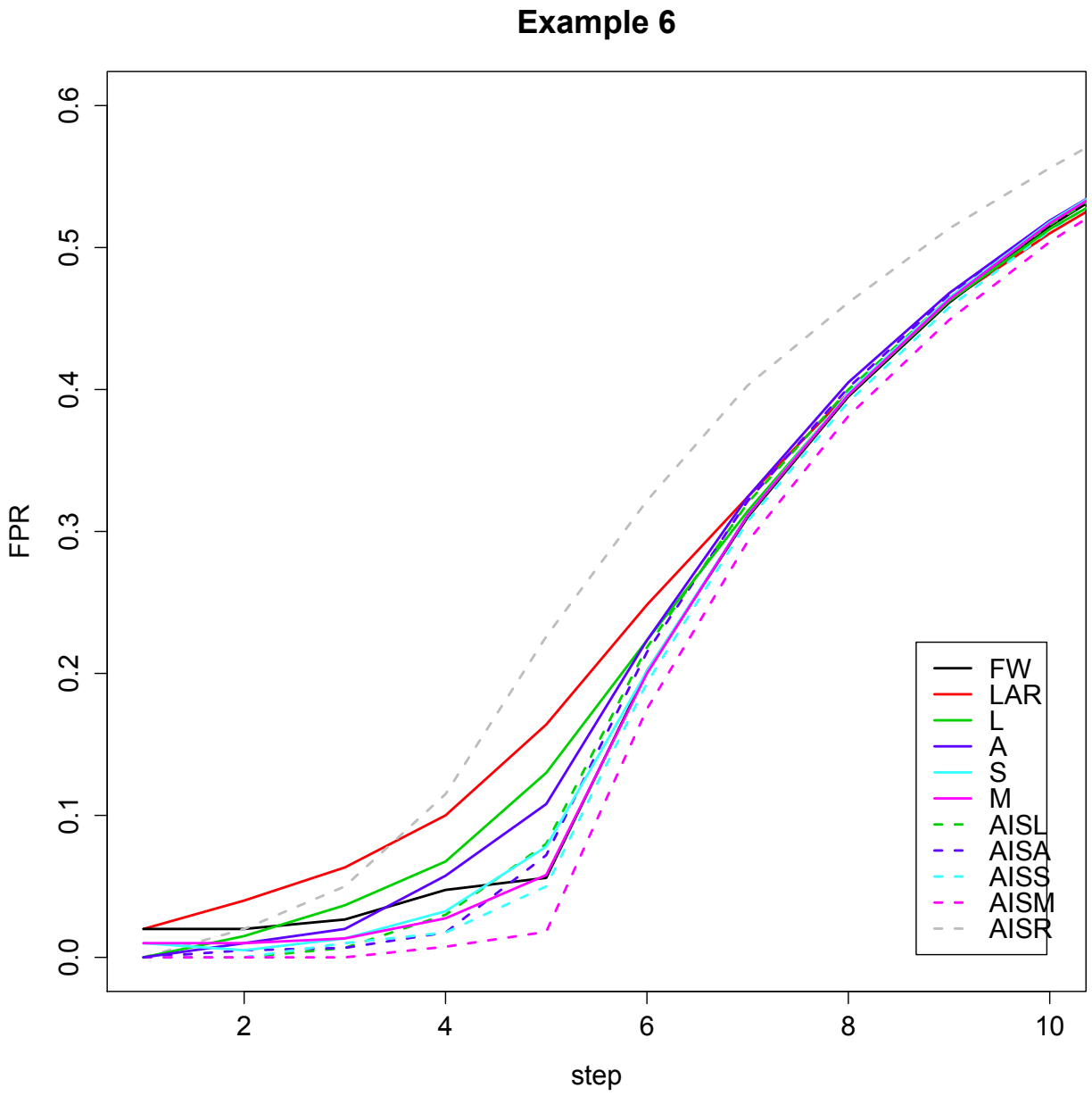

Figure 5.12: The mean of FPRs over 100 replicates for Example 6: the Forward Selection (FW), the Least Angle Regression (LAR), the original Lasso (L), the original adaptive Lasso (A), the original SCAD (S), the original MCP (M), the AIS on the Lasso (AISL), the AIS on the adaptive Lasso (AISA), the AIS on the SCAD (AISS), the AIS on the MCP (AISM), the AIS on the ridge (AISR).

original approach in these examples. For the remaining three sparse high dimensional examples, we only report the results over the first 10 steps since all the approaches can select the relevant variables in the first steps. For Example 4, the AIS methods are a little worse than the other approaches while in Example 5, the AIS algorithm on the Lasso has the smallest FPRs among all the methods. Figure 5.12 shows the FPRs of the misspecified model for Example 6. It is seen that the AIS on MCP has the best performance.

5.3 SPSP in Gaussian Graphical Modeling

In this section, we investigate the performance of the SPSP algorithm in the Gaussian graphical models through a simple simulation study.

The data is simulated from the following sparse scenario, and then we can perform the selection methods to select the nonzero entries in the inverse covariance matrix. In details, the Gaussian data is simulated as $\mathbf{X} \sim N_p(\mathbf{0}, \mathbf{\Sigma})$, where the inverse of the covariance matrix is set as

$$(\mathbf{\Sigma}^{-11})_{j,j} = 1, (\mathbf{\Sigma}^{-11})_{j,j+1} = 0.5, (\mathbf{\Sigma}^{-11})_{j+1,j} = 0.5, j = 1, \cdots, p/4,$$

and zero otherwise. We set $p = 100$ and $n = 50$ in the simulation.

This simulation study is similar with the example of the AR(1) model in Yuan and Lin (2007, [60]), which has also been used by Friedman et al. (2008, [25]) for the numerical study of the graphical Lasso. Considering all the $100(100-1)/2 = 4950$ conditional dependencies among the 100 variables, the number of the nonzero dependencies is $100/4 = 25$ and the number of the zero dependencies is $100(100-1)/2 - 25 = 4925$.

We will compare the performance of the proposed SPSP algorithm with the graphical models selected by BIC and the EBIC. Here the details about using BIC and the EBIC to choose the tuning parameter in the graphical models are described in Foygel and Drton (2010, [21]). We apply the **R** package *glasso* to solve the graphical Lasso estimators and apply the package *qgraph* to select the graphical Lasso models by BIC and the EBIC. Note that the grid of the tuning parameters in the

Table 5.10: The mean of FP, FN values of the SPSP algorithm, BIC, and the EBIC over 100 replicates (Standard Error in the parentheses). The true model has 25 nonzero dependencies and 4925 zero dependencies.

	SPSP	BIC	EBIC
FP	19.31	116.56	0
	(2.48)	(3.2)	(0)
FN	2.50	0	25
	(0.80)	(0.0)	(0)

simulation is generated automatically by the function *glassopath* and all the graphs are drawn by the function *qgraph*.

We report the mean and the standard error of the number of the false positives (FP), the number of the false negatives (FN) of the SPSP algorithm, BIC and the EBIC over 100 replicates in Table 5.10. It is observed that the BIC tends to include too many zero dependencies (high FP value) while the EBIC missed all the nonzero dependencies (high FN value) in the model. Compared with the results of these two criteria, the SPSP algorithm has a much better performance, which selects most of the nonzero dependencies without adding many zero dependencies in the model.

To infer the nonzero entries in the inverse of the covariance matrix, we draw the graphs of one replicate in this simulation study. The true model, the model obtained from the SPSP algorithm, the model selected by BIC, the model selected by the EBIC are shown in Figure 5.13. In all the graphs, we draw an edge between nodes j and k if the entry in the j-th row and k-th column of the estimated inverse covariance matrix is nonzero.

As can be seen from Figure 5.13, the SPSP algorithm correctly selects most of the edges where the corresponding true dependencies are nonzero. Meanwhile, it only includes a small number of the edges without dependencies in the true model. However, the models selected by BIC contains too many edges in the graph and the model selected selected by the EBIC does not have any edge in the graph.

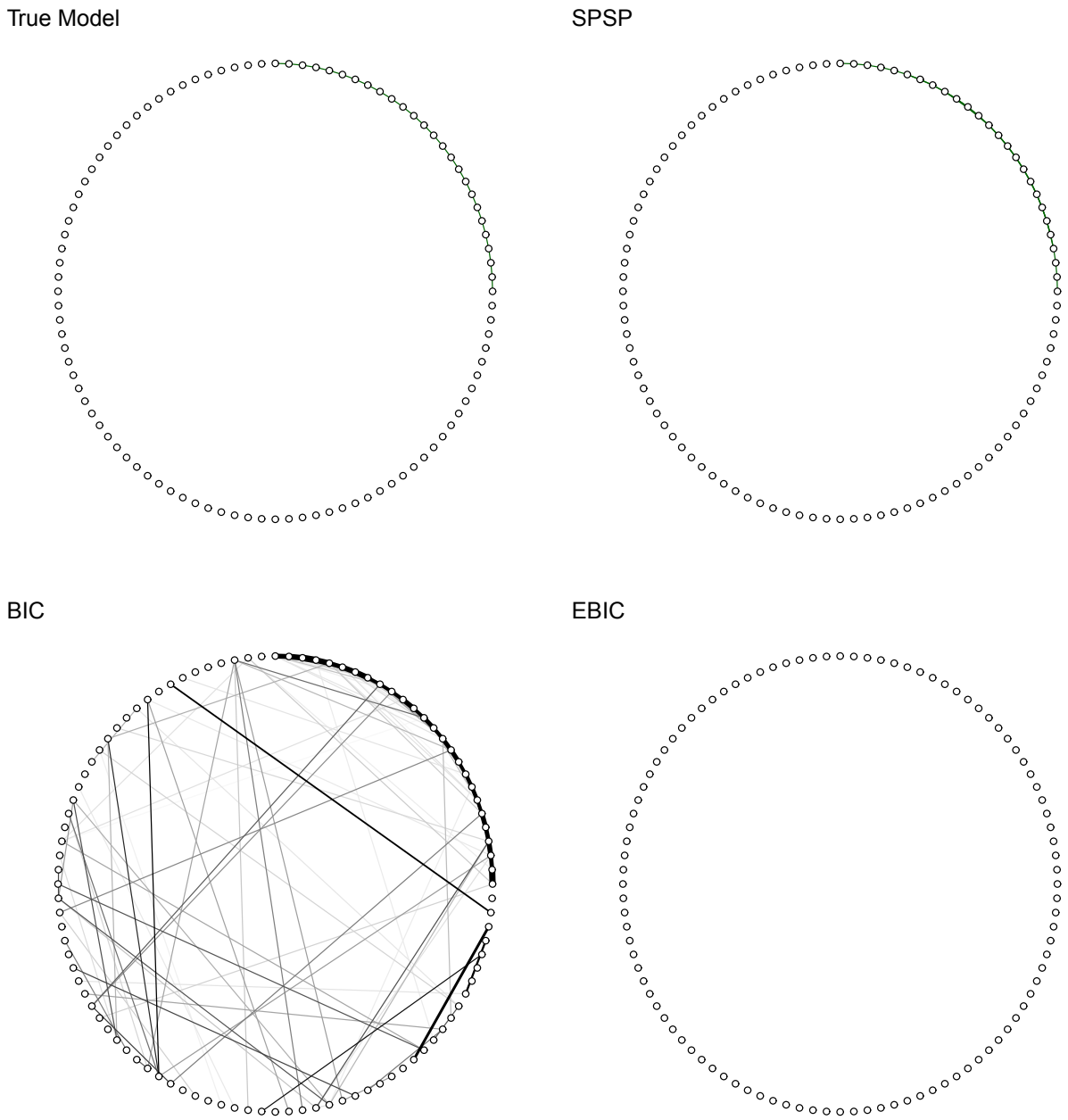

Figure 5.13: The graphical models of one replicate. Top left: the graph of the true model. Top right: the graph of the graphical model from the SPSP algorithm. Bottom left: the graph of the model selected by BIC. Bottom right: the graph of the model selected by EBIC.

CHAPTER 6 DATA ANALYSIS

In this chapter, we present two real data variable selection problems to further demonstrate the performance of the proposed algorithm.

6.1 Boston Housing Data

In this section, we apply the SPSP algorithm with the other selection criteria on the Lasso, the adaptive Lasso, the ridge to the Boston housing data (available in **R** package *MASS*). In the dataset, we are interested in identifying the informative variables on the housing prices in Boston. Particularly, we use the median value of owner-occupied homes in Boston as the response. For the predictors, as suggested in Radchenko and James (2001, [45]), we include all the remaining 13 variables as well as all the interaction terms between the variables in the analysis. Hence we have $p = 91$ variables and $n = 506$ observations in the dataset. Apparently, high correlations among the variables are presented in the dataset due to the existence of the interaction terms. Note that Cho and Fryzlewicz (2012, [12]) also used the same setting to compare the performance of different feature selection methods in their paper.

We randomly divide the dataset into a training set with $n_1 = 91$ observations ($= p$) and a test set with $n_2 = 415$ observations 100 times to evaluate the average performance of the selection approaches. Each time we standardize the data firstly; then we apply the SPSP approach and the other popular selection criteria including GCV, AIC, BIC, the EBIC on the Lasso, the adaptive Lasso and the ridge to the training set. We record the number of the selected features (Nm) and the average prediction error ($PE = n_2^{-1}||\mathbf{y} - \hat{\mathbf{y}}||_2^2$) by applying the estimations in the test set.

Table 6.1 shows the results of all the approaches averaged over the 100 training and test data sets. It is seen that the SPSP algorithm on the Lasso has the minimum prediction error with selecting a small number of variables in the model. The SPSP algorithm on the adaptive Lasso and ridge also perform well in the prediction accuracy. The GCV, AIC, BIC, EBIC on both Lasso and adaptive Lasso all include many more variables than the SPSP algorithm, which also produce large prediction

errors in the test sets.

Table 6.1: Results of Boston Housing data analysis (Standard Error in the parentheses)

Lasso	SPSP	GCV	AIC	BIC	EBIC
Nm	14.26	84.97	85.60	72.15	45.44
	(6.95)	(5.08)	(4.90)	(10.33)	(11.78)
PE	30.43	88.41	93.43	65.69	32.25
	(15.42)	(62.39)	(89.45)	(42.83)	(16.06)
ALasso	SPSP	GCV	AIC	BIC	EBIC
Nm	14.47	71.33	72.32	59.20	45.41
	(8.76)	(7.62)	(7.44)	(7.02)	(9.45)
PE	33.18	82.29	84.86	63.60	46.20
	(11.02)	(57.78)	(61.23)	(40.85)	(31.78)
Ridge	SPSP				
Nm	10.24				
	(7.19)				
PE	30.85				
	(13.00)				

6.2 Glioblastoma Gene Expression Data

In this section, we apply the SPSP approach on the glioblastoma gene expression data ([38]), which aims at identifying the highly informative genes to explain the glioblastoma tumor behavior. In the analysis, we use all the censor subjects and take the logarithm of the survival time as the response variable. Finally we obtain a data with $n = 185$ subjects and $p = 930$ genes.

Similarly, we randomly divide the dataset into a training set (120 observations) and a test set (65 observations) 100 times to evaluate the average performance of the selection methods. Each time we standardize the data firstly; then we apply the SPSP approach and the other popular selection criteria on the Lasso, the adaptive Lasso and the ridge on the training set. We also record the number of the selected features (Nm) and the average prediction error (PE) by applying the estimations in the test set.

Table 6.2 shows the results of all the approaches. We can see that the GCV, the AIC, and the BIC selected many features in the model which makes the model excessively complicated to interpret.

For the extended BIC, it cannot identify any features and proposes a simple average model. The SPSP algorithm on the Lasso, the adaptive Lasso and the ridge select a few informative features with small PE values. Here we notice that although the average simple model yields the smallest prediction error, the SPSP algorithm can provide some information with regard to identifying the informative genes in the model.

Table 6.2: Results of glioblastoma gene expression analysis (Standard Error in the parentheses)

Lasso	SPSP	GCV	AIC	BIC	EBIC
Nm	12.81	139.65	140.92	140.65	0.00
	(0.989)	(0.433)	(0.467)	(0.438)	(0.000)
PE	1.25	1.62	1.62	1.62	1.00
	(0.025)	(0.024)	(0.024)	(0.024)	(0.02)
ALasso	SPSP	GCV	AIC	BIC	EBIC
Nm	8.58	70.88	71.53	44.96	0.04
	(0.561)	(1.554)	(1.538)	(3.520)	(0.020)
PE	1.22	1.31	1.31	1.22	1.00
	(0.023)	(0.023)	(0.024)	(0.025)	(0.019)
Ridge	SPSP				
Nm	3.29				
	(0.330)				
PE	1.08				
	(0.021)				

Specifically, RRAS2, PAK1 and FRAT1 are identified by the SPSP procedure on the Lasso and the adaptive Lasso for almost all the replicates. Some previous studies have demonstrated the strong relations between these genes and the glioblastoma tumor behavior. Throughout the experiments, Aoki et al. (2007, [3]) found out that the presence of phosphorylated PAK1 in the cytoplasm of glioblastoma cells is associated with shorter survival, which suggests that the PAK1 plays a role in the invasiveness of glioblastoma and it might be a potential target for the management of glioblastoma. Demuth et al. (2008, [14]) showed the RRAS2 is one of the candidate genes whose functions are linked to glioblastoma via technical validation. Guo et al. (2010, [27]) detected that the expression of the FRAT1 in human gliomas by immunohistochemistry, Western blot and RT-PCR and concluded that FRAT1 may be an important factor in the tumorigenesis and progression of

gliomas. These studies all confirmed the selection accuracy of the proposed SPSP algorithm.

CHAPTER 7 DISCUSSION

In the thesis, we mainly propose to carry out feature selection based on the whole solution paths instead of choosing a single tuning parameter, in the framework of the penalized least squares estimation. Based on the adjacent distances between the estimated coefficients at each tuning parameter, we partition the variables into two sets: the relevant and the irrelevant. Then we identify the union of all the relevant index sets as the final relevant index set. We name the algorithm *selection by partitioning the solution paths (SPSP)*. It turns out that this algorithm can efficiently balance the trade-off between the model fitting and the model complexity. The simulation studies illustrate that the algorithm can significantly improve the selection accuracy especially when complicated correlation structures exist among the variables in the data. Especially, the SPSP algorithm has a remarkably good performance in handling the high dimensional problems.

In addition, we present a new type of scores, noted as the Area-out-of-zero-region Importance Scores (AIS), to rank the importance of the variables. The scores, defined as the areas between the coefficient curves and the boundary curve, also utilizes the information over the whole solution paths. The simulation studies show that the AIS algorithm performs well in ranking the variables for the stepwise selection problems.

Regarding the theoretical properties of the SPSP estimators, we establish several large sample theories in the dissertation. The main result illustrates that the SPSP estimators are selection consistent over the the solution paths while the original estimators are either estimation consistent or selection consistent. Specially, the SPSP estimator on the Lasso can possess the selection consistency over the whole solution paths under the irrepresentable conditions, proposed by Zhao and Yu (2006, [64]). The theoretical results of the SPSP estimators on the general penalized least squares estimation are also provided under the restricted eigenvalue conditions.

For the future work, we will continue working on the theoretical properties of the proposed SPSP algorithm for the general penalized least squares estimations over the whole solution paths.

For the selection consistency of the SPSP algorithm, we only need to guarantee the existence of a sufficiently large gap between the relevant and the irrelevant estimators. Therefore, compared with the conditions we applied in the dissertation, we will develop some weaker conditions on the design matrix to obtain the selection consistency of the proposed SPSP algorithm.

In practice, we look forward to exploring more applications on the proposed algorithms particularly when the data carries a more complicated structure. For instance, the parameters in high dimensional problems can have a group structure, which means the variables are partitioned into disjoint groups, such as the factor variables. Yuan and Lin (2006, [59]) has proposed the group Lasso to solve the problem and we are interested in modifying the SPSP algorithm for the group Lasso to analyse the data with such a group sparsity. In addition, we also want to extend the proposed algorithms into a wider family of models such as the non-parametric additive models, the graphical models. We believe that the fact a strictly convex penalty could be applied based on the proposed SPSP approach could have a huge potential in areas where computations or optimizations are a major challenge.

BIBLIOGRAPHY

[1] Tobias Abenius and Maintainer Tobias Abenius. *lassoshooting: L1 regularized regression (Lasso) solver using the Cyclic Coordinate Descent algorithm aka Lasso Shooting*, 2012.

[2] Hirotugu Akaike. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723, 1974.

[3] Hiroshi Aoki, Tomohisa Yokoyama, Keishi Fujiwara, Ana M Tari, Raymond Sawaya, Dima Suki, Kenneth R Hess, Kenneth D Aldape, Seiji Kondo, Rakesh Kumar, et al. Phosphorylated pak1 level in the cytoplasm correlates with shorter survival time in patients with glioblastoma. *Clinical Cancer Research*, 13(22):6603–6609, 2007.

[4] Maria Maddalena Barbieri and James O Berger. Optimal predictive model selection. *The Annals of Statistics*, 32(3):870–897, 2004.

[5] Peter J Bickel, Ya'acov Ritov, and Alexandre B Tsybakov. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, 37(4):1705–1732, 2009.

[6] Małgorzata Bogdan, Jayanta K Ghosh, and RW Doerge. Modifying the schwarz bayesian information criterion to locate multiple interacting quantitative trait loci. *Genetics*, 167(2):989–999, 2004.

[7] Karl W Broman and Terence P Speed. A model selection approach for the identification of quantitative trait loci in experimental crosses. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(4):641–656, 2002.

[8] Peter Bühlmann and Sara Van De Geer. *Statistics for high-dimensional data: methods, theory and applications*. Springer, 2011.

[9] Florentina Bunea, Alexandre Tsybakov, Marten Wegkamp, et al. Sparsity oracle inequalities for the lasso. *Electronic Journal of Statistics*, 1:169–194, 2007.

[10] Emmanuel J Candes and Terence Tao. Decoding by linear programming. *Information Theory, IEEE Transactions on*, 51(12):4203–4215, 2005.

[11] Jiahua Chen and Zehua Chen. Extended bayesian information criteria for model selection with large model spaces. *Biometrika*, 95(3):759–771, 2008.

[12] Haeran Cho and Piotr Fryzlewicz. High dimensional variable selection via tilting. *Journal of the Royal Statistical Society: series B (statistical methodology)*, 74(3):593–622, 2012.

[13] Peter Craven and Grace Wahba. Smoothing noisy data with spline functions. *Numerische Mathematik*, 31(4):377–403, 1978.

[14] Tim Demuth, Jessica Rennert, Dominique Hoelzinger, Linsey Reavie, Mitsutoshi Nakada, Christian Beaudry, Satoko Nakada, Eric Anderson, Amanda Henrichs, Wendy McDonough, et al. Glioma cells on the run–the migratory transcriptome of 10 human glioma cell lines. *BMC genomics*, 9(1):54–68, 2008.

[15] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004.

[16] Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001.

[17] Jianqing Fan and Runze Li. Variable selection for cox's proportional hazards model and frailty model. *The Annals of Statistics*, 30(1):74–99, 2002.

[18] Jianqing Fan and Runze Li. Statistical challenges with high dimensionality: Feature selection in knowledge discovery. *arXiv preprint math/0602133*, 2006.

[19] Jianqing Fan and Jinchi Lv. A selective overview of variable selection in high dimensional feature space. *Statistica Sinica*, 20(1):101–148, 2010.

[20] Jianqing Fan and Heng Peng. Nonconcave penalized likelihood with a diverging number of parameters. *The Annals of Statistics*, 32(3):928–961, 2004.

[21] Rina Foygel and Mathias Drton. Extended bayesian information criteria for gaussian graphical models. In *Advances in Neural Information Processing Systems*, 2010.

[22] LLdiko E Frank and Jerome H Friedman. A statistical view of some chemometrics regression tools. *Technometrics*, 35(2):109–135, 1993.

[23] Jerome Friedman, Trevor Hastie, Holger Höfling, Robert Tibshirani, et al. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

[24] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.

[25] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[26] Wenjiang J Fu. Penalized regressions: the bridge versus the lasso. *Journal of computational and graphical statistics*, 7(3):397–416, 1998.

[27] Geng Guo, Xinggang Mao, Peng Wang, Bolin Liu, Xiang Zhang, Xiaofan Jiang, Chengliang Zhong, Junli Huo, Ji Jin, and Yuzhen Zhuo. The expression profile of frat1 in human gliomas. *Brain research*, 1320:152–158, 2010.

[28] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

[29] Trevor Hastie and Brad Efron. *lars: Least Angle Regression, Lasso and Forward Stagewise*, 2013.

[30] Jennifer A Hoeting, David Madigan, Adrian E Raftery, and Chris T Volinsky. Bayesian model averaging: a tutorial. *Statistical science*, 14(4):382–401, 1999.

[31] S Horvath, B Zhang, M Carlson, KV Lu, S Zhu, RM Felciano, MF Laurance, W Zhao, S Qi, Z Chen, et al. Analysis of oncogenic signaling networks in glioblastoma identifies aspm as a molecular target. *Proceedings of the National Academy of Sciences*, 103(46):17402–17407, 2006.

[32] Jian Huang, Joel L Horowitz, and Fengrong Wei. Variable selection in nonparametric additive models. *The Annals of statistics*, 38(4):2282–2313, 2010.

[33] Clifford M Hurvich and Chih-Ling Tsai. Regression and time series model selection in small samples. *Biometrika*, 76(2):297–307, 1989.

[34] Yongdai Kim, Hosik Choi, and Hee-Seok Oh. Smoothly clipped absolute deviation on high dimensions. *Journal of the American Statistical Association*, 103(484):1665–1673, 2008.

[35] Kenneth W Kinzler, Sandra H Bigner, Darell D Bigner, Jeffrey M Trent, Martha L Law, Stephen J O'Brien, Albert J Wong, and Bert Vogelstein. Identification of an amplified, highly expressed gene in a human glioma. *Science*, 236(4797):70–73, 1987.

[36] Steffen L Lauritzen. *Graphical models*. Oxford University Press, 1996.

[37] Jinchi Lv and Jun S Liu. Model selection principles in misspecified models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(1):141–167, 2014.

[38] Roger McLendon, Allan Friedman, Darrell Bigner, Erwin G Van Meir, Daniel J Brat, Gena M Mastrogianakis, Jeffrey J Olson, Tom Mikkelsen, Norman Lehman, Ken Aldape, et al. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature*, 455(7216):1061–1068, 2008.

[39] Nicolai Meinshausen. Lasso with relaxation. In *Seminar für Statistik, Eidgenössische Technische Hochschule (ETH)*, 2005.

[40] Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3):1436–1462, 2006.

[41] Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

[42] Nicolai Meinshausen and Bin Yu. Lasso-type recovery of sparse representations for high-dimensional data. *The Annals of Statistics*, 37(1):246–270, 2009.

[43] Mee Young Park and Trevor Hastie. L1-regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):659–677, 2007.

[44] David Posada and Thomas R Buckley. Model selection and model averaging in phylogenetics: advantages of akaike information criterion and bayesian approaches over likelihood ratio tests. *Systematic biology*, 53(5):793–808, 2004.

[45] Peter Radchenko, Gareth M James, et al. Improved variable selection with forward-lasso adaptive shrinkage. *The Annals of Applied Statistics*, 5(1):427–448, 2011.

[46] Adrian E Raftery, David Madigan, and Jennifer A Hoeting. Bayesian model averaging for linear regression models. *Journal of the American Statistical Association*, 92(437):179–191, 1997.

[47] Gideon Schwarz et al. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

[48] Xiaotong Shen, Wei Pan, and Yunzhang Zhu. Likelihood-based selection and sharp parameter estimation. *Journal of the American Statistical Association*, 107(497):223–232, 2012.

[49] David Siegmund. Model selection in irregular problems: Applications to mapping quantitative trait loci. *Biometrika*, 91(4):785–800, 2004.

[50] Mervyn Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):111–147, 1974.

[51] Wei Sun, Junhui Wang, and Yixin Fang. Consistent selection of tuning parameters via variable selection stability. *The Journal of Machine Learning Research*, 14(1):3419–3440, 2013.

[52] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[53] Robert Tibshirani et al. The lasso method for variable selection in the cox model. *Statistics in medicine*, 16(4):385–395, 1997.

[54] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

[55] Sara van de Geer. The deterministic lasso. In *Seminar für Statistik, Eidgenössische Technische Hochschule (ETH) Zürich*, 2007.

[56] Hansheng Wang, Guodong Li, and Chih-Ling Tsai. Regression coefficient and autoregressive order shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(1):63–78, 2007.

[57] Sijian Wang, Bin Nan, Saharon Rosset, and Ji Zhu. Random lasso. *The Annals of Applied Statistics*, 5(1):468–485, 2011.

[58] Eleanor Wong, Suyash Awate, and P Thomas Fletcher. Adaptive sparsity in gaussian graphical models. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013.

[59] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[60] Ming Yuan and Yi Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

[61] Cun-Hui Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2):894–942, 2010.

[62] Cun-Hui Zhang, with contributions from Yi Yu Ofer Melnik, and Stephanie Zhang. *plus: Penalized Linear Unbiased Selection*, 2012.

[63] Yiyun Zhang, Runze Li, and Chih-Ling Tsai. Regularization parameter selections via generalized information criterion. *Journal of the American Statistical Association*, 105(489):312–323, 2010.

[64] Peng Zhao and Bin Yu. On model selection consistency of lasso. *The Journal of Machine Learning Research*, 7:2541–2563, 2006.

[65] Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429, 2006.

[66] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

[67] Hui Zou and Runze Li. One-step sparse estimates in nonconcave penalized likelihood models. *The Annals of Statistics*, 36(4):1509–1533, 2008.