BACKWARD ELIMINATION ALGORITHM FOR HIGH DIMENSIONAL VARIABLE SCREENING

FOLI SOPHIA KORKOR

Master's Program in Mathematical Sciences

APPROVED:

Sangjin Kim, Ph.D., Chair

Amy Wagler, Ph.D.

Michael Pokojovy, Ph.D.

Li Lin, Ph.D.

Charles Ambler, Ph.D.
Dean of the Graduate School

Dedication

to my

family and loved ones

with love and gratitude

BACKWARD ELIMINATION ALGORITHM FOR HIGH DIMENSIONAL VARIABLE SCREENING

by

FOLI SOPHIA KORKOR

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Master's Program in Mathematical Sciences

THE UNIVERSITY OF TEXAS AT EL PASO

August 2018

Acknowledgements

The Psalmist said: "Give thanks to the God of heaven. His faithful love endures forever". My heartfelt gratitude belongs to the Almighty God for His great provisions throughout my degree. I would like to express my sincere gratitude to my supervisor, Dr. Sangjin Kim of the Department of Mathematical Sciences at the University of Texas at El Paso, for his academic advice, supervision, encouragement, enduring patience and constant support. He believed in me even when i had doubts about myself. He is always readily available to provide clear explanations when i don't seem to make headway. He encouraged me to present a part of thesis at a workshop even when i wasn't prepared and that really built my presentation skills.

I also wish to extend my gratitude to my committee members, Dr. Michael Pokojovy, Dr. Amy Wagler both of the Department of Mathematical Sciences and Dr. Lin Li of the Physics Department,all of the University of Texas at El Paso. Their comments and guidance were valuable to the completion of this work.

I wish to thank Dr. Panagis Moschopoulos (my academic advisor in 1st year), Dr. Naijun Sha, Dr. Ori Rosen, Dr. Amy Wagler and Dr. Xiaogang Su, all of the Department of Mathematical Sciences at the University of Texas at El Paso who taught and mentored me through my degree. I thank Mr. Desmond Koomson of the Department of Mathematical Sciences for his support and advice.

Finally, I want to thank The University of Texas at El Paso Mathematical Science Department professors and staff who taught me and guided me to enable me complete my degree.

Abstract

In recent times, variable selection in high-dimensional data has become a challenging problem. We investigate here a popular but classical variable screening method, the Backward Elimination (BE) in a high dimensional setup (small-n-large P). The BE method as a variable screening method reduces the dimension of small-n-large P data into a lower dimensional data and then established shrinkage methods such as: LASSO, SCAD and MCP can be applied directly. To overcome the problems in high dimensional data, Chen and Chen (2008) recently developed a family of Extended Bayesian Information Criterion (EBIC) which is consistent with finite sample properties (Chen and Chen, 2008) which we used in this study to select the best candidate model from the models generated by the proposed BE method. We compare the BE with other screening methods such as: Sure Independence Screening(SIS), Iterative Sure Independence Screening and Forward Regression (FR) in simulation studies and real-data analysis to illustrate the selection consistency of our proposed BE method. Our numerical analysis reveals that the BE with EBIC can identify all important variables with high coverage probability, low false discovery rate and a very good model size with high signal-to-noise.

KEY WORDS: Variable Screening, Backward Elimination, EBIC, LASSO, SCAD, MCP, ISIS, SIS, FR, High Dimensional data

Table of Contents

List of Tables

List of Figures

Chapter 1

1. Introduction

1.1 Background

In recent years, modern research in genome-wide studies, health science, etc, encounters high dimensional data problems. In most cases, the predictor variables are much larger than the sample and this poses a lot of problem in analyzing such data set analysis such data set. In a high dimensional setup, among the predictor variables are relevant variables as well as irrelevant in reference to the response variable. The goal of a researcher in such cases is to discover all of the relevant predictors while discading the irrelevant ones in relation to the response variable. To this end, a lot of research has been devoted to this subject in the last decades and one of these research areas is variable selection. There are lots of variable selection methods like Forward regression, Backward regression, etc., however these methods are good for situations where the sample size is usually much larger than the predictor variables. Therefore, variable selection with a high dimensional predictor is a problem of fundamental importance according to Fan & Li (2006). To address this problem, variable screening has become of importance as a step in variable selection. This step generally removes irrelevant predictors, greatly simplifying the problem of high dimension to a low dimension one. To this end, various methods such as SIS, FR, etc.,have gained popularity in the last decades. The selection consistency of these methods have been established both theoretically and numerically. Motivated by the outstanding performance of the FR and the SIS, we rigorously analyze another popular yet classical variable screening method, the Backward Elimination (BE). Like the FR, we establish the screening consistency of the BE method under high-dimensional setup. Chen and Chen (2008) recently proposed a family

of extended Bayes information criteria (EBIC) especially for variable selection in high dimensional situations to address the problem of overestimation of the BIC. They established the consistency of EBIC under normal linear models.

The objective of our study is to contribute to existing literature by establishing modifying the BE with EBIC's screening consistency procedure in a high dimensional setup. The resulting model from the BE method can serve as a starting point from where other variable selection method (e.g LASSO, SCAD and MCP) can be applied. We confirm this through simulation studies and real-data analysis. We use the EBIC instead of the conventional BIC to access the models selected by the BE method.

1.2 Outline of Thesis

The remaining parts of the thesis are organized in this manner. Chapter 2 provides a literature review on variable selection, screening and model selection criteria in detail. In Chapter 3, our proposed method, Backward Elimination is presented and explained in detail. Through simulation studies we will analyze and compare the performance of the proposed method to existing methods in Chapter 4. We will apply our proposed method further on a real data example and report on the findings. Finally, we will discuss the limitations as well as strength of our proposed method and provide areas for future work in Chapter 5.

Chapter 2

Literature Review

The aim of this chapter is to present a literature review on the steps involved in variable selection with focus on the screening step. Limitations of variable selection methods in high-dimensional settings are also addressed. Literature on variable screening methods are further examined.

2.1 Variable Selection

In most practical problems, the analyst has a rather large pool of possible candidate variables, of which only a few are likely to be important. Finding an appropriate subset of variables for a model is often called the variable selection problem. This problem generally falls under the Exploratory Observational Studies (e.g., exploration of numerous gene sets that might not all be associated with the continuous response). This is because a model with numerous explanatory variables may be difficult to maintain. Also, the presence of many highly intercorrelated explanatory variables may worsen the model's predictive ability (Applied Linear Regression Model ,M.Kutner,2004)

In recent years, there has been much research efforts on dealing with the challenging problem of variable selection in high dimensional data (small-n-large P). This is largely due to modern applications in medical studies, genetic research, bioinformatics, and other fields. The consistency of various traditional methods of variable selection has been established over the years. These methods include but are not limited to the LASSO (Tibshirani, 1996, 1997), the SCAD (Fan and Li, 2001; Fan and Peng, 2004), MCP (Zhang, 2010), and related models. These methods have been applied for simultaneously selecting important

variables and estimating their effects in high-dimensional statistical inference, and have demonstrated excellent performance in simulation studies (J.S. Hwang& T.H Hu, 2014). All these methods have been shown to be useful and can be formulated as penalized optimization problems which could be selection consistent in a low dimensional data setup (Hwang, 2009; Fan and Peng, 2004; Huang et al., 2008; Zou and Zhang,2008). Nevertheless, in high dimensional setup, these methods may not work well due to the simultaneous challenges of computational expediency, statistical accuracy, and algorithmic stability (Fan et al., 2009).

When the predictor dimension is much larger than the sample size, efficient algorithms exist for methods like LASSO (Efron et al., 2004, LARS) where the objective functions are strictly convex. Similarly, for methods like SCAD, etc., computationally how to optimize these non-convex objective functions remains a non-trivial task (Hunter and Li, 2005;Chen and Chen, 2008). Though most of these variable selection methods above tend to have good theoretical properties; the difficulty in choosing a penalty function still remains a challenge.

To address the problem of variable selection in high dimensional setup, Ing and Lai (2011) introduced a fast stepwise regression algorithm called High-dimensional information criterion (HDIC) which has been shown to have the oracle property of being equivalent to least squares regressions on an asymptotically minimal set of relevant predictors under a strong sparsity assumption. This method is shown to have impressive performance. The question still remains on how we can do variable selection in a high-dimensional setup.

2.2 Variable Screening

One reasonable solution to variable selection in a high dimensional setup is variable screening. As an example, consider a gene expression data ($p >> n$) with 5000 genes as predictor variables, there would be 2^{5000} possible regression models to be considered. This would be an overwhelming task hence the need for screening. Variable screening is therefore use-

ful in reducing the dimensionality of the variable space to a moderate one and then we can apply variable selection techniques (He, Wang and Hong, 2013). Motivated by these concerns, there has been a dramatic growth in the development of statistical methodology in the analysis of high dimensional data (Shahriari, Fana and Goncalves, 2015). According to the paper by Fan,J. and J.Lv(2008), a common practice for variable screening is using independence learning which treats the features as independent and thus applies marginal regression techniques. Motivated by the aforementioned fundamental challenges of ultra-high dimensional data analysis, the sure independence screening (SIS) was formally introduced and justified by Fan,J. and J.Lv (2008) to address both issues of scalability and noise accumulation. SIS method as proposed by Fan and Lv (2008) for linear regression first filters out the variables that have weak correlation with the response, effectively reducing the dimensionality to a moderate scale below the sample size n, and then performing variable selection and parameter estimation through a lower dimensional penalized least squares method. According to S.Kim and S. Halabi (2016),Although this approach is popular, it does not perform well under some situations. First, unimportant variables that are heavily correlated with important predictors are more highly likely to be selected than relevant variables that are weakly associated with the response. Second, important variables that are not marginally significantly related to the response are screened out. Finally, there may be collinearity between variables that may impact the calculations of the individual predictors.

An important methodology extension, Iterative Sure Independence Screening (ISIS), was also proposed by Fan and Lv (2008) to handle cases where regularity conditions may fail, such as when some important variables are marginally uncorrelated with the response, or when an unimportant predictor has higher marginal correlation than some important features. This method iteratively performs variable selection to recruit small number of predictors, computing residuals based on the model fitted using these recruited predictors, and then using as the working response variable to continue recruiting new predictors. Motivated by the outstanding performance of SIS, Wang (2009) also investigated on the

screening consistency of the forward regression (FR) under an ultra-high dimensional setup. The forward regression (FR) with EBIC can consistently identify both theoretically and numerically all relevant predictors. The performance of the FR with variable selection methods was established through simulation studies and real data application. Extensive work has been done on FR as a variable screening method as well Stepwise regression as a selection technique; but to the best of my knowledge, this is the first work that utilizes the Backward Elimination (BE) as a variable screening method in a high dimensional setup. Nevertheless, the BE has been extensively discussed as a variable selection method in the low dimensional setup ($n >> p$)

2.3 Model Selection Criteria

To practically select the "best" candidate from the models generated by the methods discussed, model selection criteria(e.g. Akaike Information Criterion (AIC) (Akaike, 1974), Bayesian Information Criterion (BIC) (Schwarz, 1978)) have been developed in past years. The efficacy of these criteria has been established through various simulation studies and real data applications. However, these methods may not be good choices in a high dimensional setting. This is because these classical model selection criteria become overly liberal and fail to serve the purpose of variable selection. They tend to select a model with spurious predictors when used together with the traditional selection methods.Moreover, these criteria are not selection consistent. This has been observed by Broman & Speed (2002), Siegmund (2004), and Bogdan et al. (2004) in genetic studies.

To tackle this problem, Chen and Chen (2008) recently proposed the family of extended Bayesian information criterion(EBIC). They further established that under normal linear regression, EBIC is found to be consistent with interesting finite sample properties (Chen & Chen, 2008). As an extension to this, they further established the consistency of this method under generalized linear models in a high dimensional situation. The EBIC crite-

ria tightly controls false discovery rate (FDR) though it incurs a small loss in the positive selection rate. Due to the performance of this criteria, Wang (2009) used this method to select the best model generated by the FR.

Chapter 3

Methodolgy

In this chapter we describe variable screening methods which reduces high dimensionality (Backward Elimination,Sure Independence Screening(SIS) and Iterative Sure Independence Screening (ISIS)). We then describe methods of variable screening (LASSO, SCAD and MCP). Finally, we describe the methods needed to assess variable selection models.

3.1 Backward Elimination

The Backward Elimination method begins with the full least squares model containing all p predictors, and then iteratively removes the least useful predictor, one-at-a-time. Briefly, the procedure works as follows ;

1. Start with all variables, p, in the model.
2. Remove the variable with the largest BIC value; that is the variable that is the least statistically significant.
3. The new (p - 1) variable model is compared with the full model and then the variable with the largest BIC is removed.
4. Continue until all selected variables in the final model minimizes the BIC value.

This method however is computationally infeasible in a high dimensional setup. Moreover, the model criteria used along with this method (AIC,BIC,p-value,etc.) become overly liberal and fail to serve the purpose of variable selection.

3.1.1 Modified BE Algorithm

Considering the Backward Elimination method as a screening method; We propose a modification to the BE Algorithm with the the idea of the SIS method that ranks all the p variables based on the marginal correlations, $\hat{corr}(x_j, y)$, of x_{js} with the response ,y, and retains the top d covariates with the largest absolute correlations, where d is the reduced dimension. The covariates with the largest absolute correlations is collected in the set $\widehat{\mathcal{M}}$; that is,

$$\widehat{\mathcal{M}} = \{1 \leq j \leq p : |\widehat{corr}(x_j, y)| \ is \ among \ the \ top \ d \ largest \ ones\} \tag{3.1}$$

where $\widehat{corr}$ denotes the sample correlation. According to Fan & Lv (2008), the idea of SIS is identical to selecting predictors using their correlations with the response. To implement SIS, they proposed choosing $d = [\gamma n]$ to be conservative, for instance, $n - 1$ or $n/logn$ depending on the order of sample size n. We borrowed this idea on choosing variables based on correlations and chose $d = n/logn$. The Modified BE algorithm works below as:

Step 1: Rank the variables according to its correlation from largest to smallest.

Step 2: Select top n number of variables.

Step 3: Split the variables by d = [n/log n] thus $\{p_1, \ldots, p_d\}$ variables.

Step 4: Start with the $\{p_1, \ldots, p_d\}$ variables in the model.

Step 5: Find sub-best model based on extended BIC using backward elimination.

Step 6: Add the next variables from $\{p_{d-1}, \ldots, p_p\}$ to variables in the sub-best model to obtain d number of variables in the model such that $d = [n/logn]$ variables are screened in each iterative step

Step 7: Repeat steps 5 and 6 until all of the p variables are used or extended BICs between reduced model and full model are the same in Step 5.

3.2 Sure Independence Screening(SIS)

Consider the linear regression model

$$y = X\beta + \epsilon \tag{3.2}$$

where $y = (y_1, \ldots, y_n)^T$ is an n-dimensional response vector, $X = (x_1, \ldots, x_p.)^T$ is an $n \times p$ design matrix consisting of p covariates $x_{j's}, \beta = (\beta_1, \ldots, \beta_p)$ is a p-dimensional regression coefficient vector, and $\epsilon = (\epsilon_1, \ldots, \epsilon_n)^T$ is an n-dimensional error vector. The focus of Fan, J. and J.Lv (2008) is the ultra-high dimensional setting with $\log p = O(n^\alpha)$ for some $0 \leq \alpha \leq 1$. To ensure model identifiability, the true regression coefficient vector $\beta_0 = (\beta_0, \ldots, \beta_p)^T$ is assumed to be sparse. The covariates x_{js} with indices in the support $\mathcal{M}_* = supp(\beta_0) = 1 \leq j \leq p : \beta_{0,j} \neq 0$ are called important variables, while the remaining covariates are referred to as noise variables(Fan,J. and J.Lv(2008)). The SIS is a two-scale learning framework in which large-scale screening is first applied to reduce the dimensionality from p to a moderate one d, say, below sample size n, and moderate-scale learning and inference are then conducted on the much reduced variable space.

In particular, the SIS ranks all the p features using the marginal utilities based on the marginal correlations $\widehat{corr}(x_j, y)$ of x_{js} with the response y and retains the top d covariates with the largest absolute correlations collected in the set $\widehat{\mathcal{M}}$; that is,

$$\widehat{\mathcal{M}} = \{1 \leq j \leq p : |\widehat{corr}(x_j, y)| \ is \ among \ the \ top \ d \ largest \ ones\} \tag{3.3}$$

where $\widehat{corr}$ denotes the sample correlation. This achieves the goal of variable screening (Fan,J and J. Lv(2008)).

An important question is whether it contains all the important covariates in the set $\mathcal{M}_*$ with asymptotic probability one; that is,

$$P = \{\mathcal{M}_* \subset \widehat{\mathcal{M}}\} \to 1 \tag{3.4}$$

as $n \to \infty$. The property in (3.3) was termed as the sure screening property by Fan,J. and J. Lv(2008) which is crucial to the second step of refined variable selection. Surprisingly,

SIS was shown by Fan,J. and J.Lv (2008) to enjoy the sure screening property under fairly general conditions, with a relatively small size of $\widehat{\mathcal{M}}$. Specifically, the p covariates x_{js} are allowed to be correlated with covariance matrix $\sum$ and the p-dimensional random covariate vector multiplied by $\sum^{-1/2}$ is assumed to have a spherical distribution. The sure screening property of SIS depends upon the so-called concentration property for random design matrix X introduced by Fan,J and J.Lv (2008).

With such a property, the sure screening property (3.3) can hold for $d = o(n)$, leading to the suggestion of choosing $d = n-1$ or $[n/\log n]$ for SIS in the original paper by Fan,J. and J.Lv (2008). In practice, the parameter d can be chosen by some data-driven methods such as the cross-validation and generalized information criterion (Fan,Y and C.Tang,2013).

3.3 Iterative Sure Independence Screening(ISIS)

ISIS is an extension to the SIS method introduced by Fan,J. and J.Lv(2008) to address the limitations of the SIS. Mainly, the idea is to iteratively update the estimated set of important variables, using SIS conditional on the estimated set of variables from the previous step. Intuitively, such an iterative procedure can help recruit important covariates that have very weak or no marginal associations with the response in the presence of other important ones identified from earlier steps.

For logistic regression, the algorithm according to S.Kim and S.Halabi (2016) works in the following way:

1. The likelihood of marginal logistic regression (LMLR) is computed for every $j \in S = \{1, 2, \ldots, p\}$. Then which is $N/4log(N)$ of the top ranked variables of the descending order list of the LMLR is selected to obtain the index set $\widehat{I}_1$

2. Apply those variables in $\widehat{I}_1$ to the penalized logistic models to obtain a subset of indices $\widehat{M}_1$

3. For every variable $j \in \{S - \widehat{M}_1\}$,the likelihood of the marginal logistic regression

condition on the variables in $\widehat{M}_1$ is solved. Then the likelihood estimators are sorted in descending order and then the d top ranked variables are selected to get the index set $\widehat{I}_2$.

4. Apply those variables in $\widehat{I}_2 \cup \widehat{M}_1$ to the penalized logistic models to obtain a new index set $\widehat{M}_2$

5. Steps (3) and (4) are repeated until $\widehat{M}_l = d$ or $\widehat{M}_l = \widehat{M}_{l-1}$

3.4 Variable Selection Methods

3.4.1 Least Absolute Shrinkage and Selection Operator (LASSO)

Definition 1 *The Lasso is a shrinkage and selection method for linear regression. It minimizes the usual sum of squared errors, with a bound on the sum of the absolute values of the coefficients.It forces the coefficients of unimportant variables to be set to 0. The LASSO has sparsity property (S.Kim and S. Halabi,2016).*

According to R.Tibshirani (1996), Suppose that we have data (x^i, y_i), $i = 1, 2, \ldots, N$,where $x^i = (x_{i1}, \ldots, x_{ip})^T$ are the predictor variables and y_i are the responses. As in the usual regression set-up, we assume either that the observations are independent or that the $y_i's$ are conditionally independent given the x_{ij}^s. We assume that the x_{ij} are standardized so that $\sum_i x_{ij}/N = 0$, $\sum_i x_{ij}^2/N = 1$. Letting $\hat{\beta} = (\hat{\beta}_1, \ldots, \hat{\beta}_p)^T$, the lasso $(\hat{\alpha}, \hat{\beta})$ is defined by

$$(\hat{\alpha}, \hat{\beta}) = argmin\left\{ \sum_{i=1}^{N} \left(y_i - \alpha - \sum_j \beta_j x_{ij} \right)^2 \right\} \ subject \ to \ \sum_j \mid \beta_j \mid \leq t \qquad (3.5)$$

here $t \geq 0$ is a tuning parameter. Now, for all t, the solution for α is $\hat{\alpha} = \bar{y}$. We can assume without loss of generality that $\bar{y} = 0$ and hence omit α. The parameter $t \geq 0$ controls the amount of shrinkage that is applied to the estimates. Let $\hat{\beta}_j^o$ be the full least squares estimates and let $t_0 = \sum \mid \hat{\beta}^0 \mid$. Values of $t \leq t_0$ will cause shrinkage of the solutions 0, and some coefficients may be equal to 0. The design matrix need not be of full rank.

3.4.2 Smoothly Clipped Absolute Deviation (SCAD)

The SCAD with concave penalty function overcomes the limitation of LASSO. The LASSO thresholding penalty functions do not simultaneously satisfy the mathematical conditions for unbiasedness, sparsity, and continuity. The continuous differentiable penalty function defined by

$$p_\lambda^1(\theta) = \lambda\Big\{I(\theta \le \lambda) + \frac{(a\lambda - \theta)+}{(a-1)\lambda} I(\theta > \lambda)\Big\} \;\; for \;\; some \;\; a > 2 \;\; and \;\; \theta > 0, \tag{3.6}$$

improves the properties of the L1 penalty.We call this penalty function the smoothly clipped absolute deviation (SCAD) penalty.It corresponds to a quadratic spline function with knots at λ and $a\lambda$. This penalty function leaves large values of θ not excessively penalized and makes the solution continuous (Fan,J. and Li,R.,2001). The smoothly clipped absolute deviation method not only selects important variables consistently, but also produces parameter estimators as efficient as if the true model were known, i.e., the oracle estimator, a property not enjoyed by the Lasso. The above features of the smoothly clipped absolute deviation method rely on the proper choice of tuning or regularization parameter, which is usually selected by generalized cross validation(Wang,H., Li,R., Tsai,C.) (Craven & Wahba, 1979).

3.4.3 Minimum Concave Penalty (MCP)

The idea behind the MCP is very similar to the SCAD. The continuous differentiable penalty function is defined by

$$p_\lambda^1(\theta) = \Big(\lambda - \frac{\mid \theta \mid}{a}\Big) sign(\theta) \;\; for \;\; some \;\; a > 1 \tag{3.7}$$

As with SCAD, MCP starts out by applying the same rate of penalization as the lasso, then smoothly relaxes the rate down to zero as the absolute value of the coefficient increases in comparison to SCAD, however, the MCP relaxes the penalization rate immediately while with SCAD the rate remains flat for a while before decreasing(C.H.Zhang,2010)

3.5 The Extended Bayes Information Criterion (EBIC)

The extended Bayes information criterion is particularly suitable for model selection for large model spaces (J.Chen & Z.Chen,2008).

The EBIC is given by

$$BIC_\gamma(s) = -2\log L_n\{\hat{\theta}(s)\} + v(s)logn + 2\gamma\log\tau(S_j) \;\; 0 \leq \gamma \leq 1 \tag{3.8}$$

where $\hat{\theta}(s)$ is the maximum likelihood estimator of $\theta(s)$ given model s. Also, S_j is a partition of the model space and $\tau(s_j)$ is size of S_j. The first two terms in $BIC_\gamma(s)$ are the Laplace approximation to $-2\log\{m(Y|s)\}$ where $m(Y|s)$ is the likelihood of the model s, and the last term is $-2\log\{p(s)\}$ up to a constant. Chen & Chen(2008) showed that, under certain conditions, the EBIC is selection consistent when γ is larger than $1-1/(2\kappa)$. Theoretically, Chen &Chen(2008) proved the selection consistency of the EBIC if $P = p_n = O(n^\kappa)$ as n $\to \infty$ for some $\kappa > 0$.

Chapter 4

Simulation Studies

4.1 Preliminaries

In this chapter extensive simulation studies have been conducted to evaluate the performance of the Backward selection method. The performances of the SIS, Iterative SIS and FR were also examined for the purpose of comparism. Mainly, there are four variable screening methods (i.e., SIS, ISIS, FR and BE) were compared. For the variable selection methods, we considered the LASSO, SCAD and MCP with the EBIC criterion (3.7).
Following the examples used in the Forward regression paper by H.Wang(2009) and other papers;
For each parameter setup,a total of 200 simulation replications are conducted. The theoretical $R^2 = var(X_i^T\beta)/var(Y_i)$ are given by $30\%, 60\%$ or 90%. Let $\hat{\beta}_{(k)} = (\hat{\beta}_{1(k)}, \ldots, \hat{\beta}_{p(k)})^T \in R^d$ be the estimator realized in the kth simulation replication by one particular method (e.g., BE-SCAD). Then, the model selected by $\hat{\beta}_{(k)}$ is given by $\hat{S}_{(k)} = \{j :| \hat{\beta}_{j(k)} |> 0\}$ and the corresponding Model size $= \sum_k \mid \hat{S}_{(k)} \mid$. To characterize the method's capability in producing sparse solutions, we defined: the True Positive rate (TPR) which measures the proportion of correctly identified true variables; False Positive rate (FPR) which is the proportion of variables incorrectly identified as important or true; True Negative rate (TNR) which measures the proportion of variables that are correctly identified as unimportant; False Negative rate (FNR) measures the proportion of true variables that are incorrectly identified as unimportant; and the False Discovery rate (FDR) which is the expected proportion of Type 1 error (FP)

4.2 The Simulation Models

For numerical comparison, we considered the following five simulation models.

Example 1. (Independent Predictors) This is an example by Fan and Lv (2008) with $(n, p, p_0)= (200, 5000, 8)$. X_i is generated independently according to a standard multivariate normal distribution. Thus, different predictors are mutually independent. The jth $(1 \leq j \leq p_0)$ nonzero coefficient of β is given by $\beta_j = (-1)^{U_j}(4 \log n/\sqrt{n} + \mid Z_j \mid)$, where U_j is a binary random variable with $P(U_j = 1) = 0.4$ and Z_j is a standard normal random variable.

Example 2 (Autoregressive Correlation) This is an example from Tibshirani (1996) and modified by Wang (2009).We considered an autoregressive correlation structure type. This correlation structure might be useful if a natural order exists among the predictors. As a consequence, the predictors with large distances in order are expected to be mutually independent approximately.An example from Tibshirani (1996) with $(n, p, p_0)= (200, 8000, 3)$. X_i is generated from a multivariate normal distribution with mean 0 and $cov(X_{ij1}, X_{ij2})$=$0.5^{|j_1 - j_2|}$. The 1st, 4th, and 7th components of β are given by 3, 1.5, and 2 respectively. Other components of β are to be fixed at 0.

Example 3 (Compound Symmetry). This is an example from Fan and Lv (2008) and Wang (2009). By this structure, all predictors are equally correlated with each other. From the example used by Fan and Lv (2008) with $(n, p, p_0)= (75, 5000, 3)$. X_i is generated such that $var(X_ij) = 1$ and $var(X_{ij1}, X_{ij2}) = 0.5$ for any $j_1 \neq j_2$. The nonzero coefficients of β are fixed to be 5.

Example 4 (A Challenging Case or Extreme Correlation) To further test the performance of BE, we used the challenging example by H.Wang(2009) with $(n, p, p_0)= (300, 5000, 5)$ and $\beta_j = 2j$ for every $1 \leq j \leq p_0$. They simulated independently $Z_i = (Z_{ij}) \in R^d$ and $W_i = (W_{ij}) \in R^d$ from a standard multivariate normal distribution. Next they generated X_j according to $X_{ij} = (Z_{ij} + W_{ij})/\sqrt{2}$ for every $1 \leq j \leq p_0$ and $X_{ij} = (Z_{ij} + \sum_{j\prime=1}^{p_0} Z_{ij\prime})/2$ for every $p_0 \leq j \leq p$. According to the paper, a simple Monte Carlo computation reveals

that the correlation coefficient of X_{i1} and Y_i is much smaller than that of X_{ij} and Y_i for every $j > p_0$; where X_{ij} is an irrelevant predictor for every $j > p_0$.

Example 5 (Normality Assumption) For linear regression, we mostly assume normality for the error term, ϵ_i. To this end, the performance of the BE is tested against non-normally distributed ϵ_i. Wang (2009) replicated Example 1 but with both X_{ij} and ϵ_i generated independently from a standardized exponential distribution, i.e., exp(1)-1.

4.3 Simulation Results

The simulation results are summarized in Tables 1-10. Based on this, we may draw the following conclusions.

Considering Example 1 & 2:

From Tables 4.1 and 4.3, in the low signal-to-noise (i.e. Theoretical R^2=30%) no method performs well in terms of its ability to correctly select important variables aside the SIS and the ISIS. The BE and FR have almost the same performance but the BE has a better estimated model size and TPR as compared to the FR. Though the SIS and ISIS performs well in correctly identifying true variables, this however is obtained by sacrificing a much larger model size. We want to maintain a low FPR and the FDR but the SIS and the ISIS have quite high FDR and FPR as compared to the BE. It is worth noting that for Table 4.3, the FDR for the ISIS and SIS is 91.9% which doesn't serve our aim of maintaining a low FDR.

From Tables 4.2, as the signal-to-noise increases (i.e.Theoretical R^2=90%), all the methods perform well in terms of it's ability to correctly identify relevant variables and irrelevant ones. The BE method performs competitively with the other methods. The BE method discovers 95% of important variables and 100% of unimportant variables under Example 1. Also, the estimated Model size is 8.4 which is very similar to the predetermined model size, 8. The BE method comparably maintains a low FDR of 8.6% and a FPR of 0,

which is our aim. This is because the EBIC criteria which was used with the BE method tightly controls FDR and incurs a small loss in the TPR (Chen & Chen, 2008).
From Table 4.4, increasing the signal-to-noise doesn't change the perform of the SIS and ISIS but significantly improves the TPR, the Model size and minimizes the FDR of the BE. Considering Example 3 and 4:

From Table 4.5 and 4.7, the BE method as well as the FR doesn't perform well especially when there is extreme correlation between predictors (i.e. Table 4.7). However, the SIS and ISIS performs better though the Model size is large and the FDR is large. From Table 4.6, the performance of the BE is pretty good in terms of the Model size and the FDR and the performance of the SIS and ISIS remains unchanged under the Theoretical R^2=90%. It is noteworthy that, from Table 4.8, increasing the Theoretical R^2 doesn't significantly improve the performance of the BE method. We can conclude that in the case of extreme correlation, our BE method finds it difficult to identify true variables. The simulation results for Example 5 is quite similar to the results of Example 1. This is because we replicated Example 1 without the normality assumption of X_{ij} and ϵ_i.

We are not claiming the BE as the only good method for variable screening though it can very promising as compared with other variable screening methods reported in the simulation studies.

Table 4.1: Example 1: (n, d, d_0)=(200,5000,8) and R^2=30%

Screening Method	Selection Method	Coverage Probability	True Positive Rate (%)	False Positive Rate (%)	True Negative Rate (%)	False Negative Rate (%)	Model Size	False Discovery Rate (%)
			Theoretical $R^2 = 30\%$					
BE	NONE	0.262	0.262	0.001	0.999	0.738	6.9	0.605
	LASSO	0.262	0.262	0.001	0.999	0.738	6.9	0.605
	SCAD	0.262	0.262	0.001	0.999	0.738	6.9	0.605
	MCP	0.262	0.262	0.001	0.999	0.738	6.9	0.605
FR	NONE	0.25	0.25	0.001	0.999	0.75	5.2	0.596
	LASSO	0.25	0.25	0.001	0.999	0.75	5.2	0.596
	SCAD	0.25	0.25	0.001	0.999	0.75	5.2	0.596
	MCP	0.25	0.25	0.001	0.999	0.75	5.2	0.596
SIS	LASSO	1	1	0.006	0.994	0	37	0.784
	SCAD	1	1	0.006	0.994	0	37	0.784
	MCP	1	1	0.006	0.994	0	37	0.784
ISIS	LASSO	1	1	0.006	0.994	0	37	0.784
	SCAD	1	1	0.006	0.994	0	37	0.784
	MCP	1	1	0.006	0.994	0	37	0.784

Table 4.2: Example 1: (n, d, d_0)=(200,5000,8)and R^2=90%

Screening Method	Selection Method	Coverage Probability	True Positive Rate (%)	False Positive Rate (%)	True Negative Rate (%)	False Negative Rate (%)	Model Size	False Discovery Rate (%)
			Theoretical $R^2 = 90\%$					
BE	NONE	0.95	0.95	0	1	0.05	8.4	0.086
	LASSO	0.95	0.95	0	1	0.05	8.4	0.086
	SCAD	0.95	0.95	0	1	0.05	8.4	0.086
	MCP	0.95	0.95	0	1	0.05	8.4	0.086
FR	NONE	0.95	0.95	0	1	0.05	8.4	0.086
	LASSO	0.95	0.95	0	1	0.05	8.4	0.086
	SCAD	0.95	0.95	0	1	0.05	8.4	0.086
	MCP	0.95	0.95	0	1	0.05	8.4	0.086
SIS	LASSO	0.975	0.975	0.006	0.994	0.025	37	0.789
	SCAD	0.938	0.938	0.006	0.994	0.062	37	0.797
	MCP	0.925	0.925	0.006	0.994	0.075	37	0.8
ISIS	LASSO	0.975	0.975	0.006	0.994	0.025	37	0.789
	SCAD	0.938	0.938	0.006	0.994	0.062	37	0.797
	MCP	0.925	0.925	0.006	0.994	0.075	37	0.8

Table 4.1 and 4.2, represents the simulation result for low signal-to-noise (i.e.,Theoretical R^2=30%) and high signal-to-noise (i.e.,Theoretical R^2=90%). The BE method as well as FR doesn't perform well however increasing the signal-to-noise improves the performance of these methods

Table 4.3: Example 2: (n, d, d_0)=(200,8000,3) and R^2=30%

Screening Method	Selection Method	Coverage Probability	True Positive Rate (%)	False Positive Rate (%)	True Negative Rate (%)	False Negative Rate (%)	Model Size	False Discovery Rate (%)
			Theoretical $R^2 = 30\%$					
BE	NONE	0.2	0.2	0	1	0.8	2.2	0.717
	LASSO	0.2	0.2	0	1	0.8	2.2	0.717
	SCAD	0.2	0.2	0	1	0.8	2.2	0.717
	MCP	0.2	0.2	0	1	0.8	2.2	0.717
FR	NONE	0.233	0.233	0	1	0.767	2.2	0.65
	LASSO	0.233	0.233	0	1	0.767	2.2	0.65
	SCAD	0.233	0.233	0	1	0.767	2.2	0.65
	MCP	0.233	0.233	0	1	0.767	2.2	0.65
SIS	LASSO	1	1	0.004	0.996	0	37	0.919
	SCAD	1	1	0.004	0.996	0	37	0.919
	MCP	1	1	0.004	0.996	0	37	0.919
ISIS	LASSO	1	1	0.004	0.996	0	37	0.919
	SCAD	1	1	0.004	0.996	0	37	0.919
	MCP	1	1	0.004	0.996	0	37	0.919

Table 4.4: Example 2: (n, d, d_0)=(200,8000,3) and R^2=90%

Screening Method	Selection Method	Coverage Probability	True Positive Rate (%)	False Positive Rate (%)	True Negative Rate (%)	False Negative Rate (%)	Model Size	False Discovery Rate (%)
			Theoretical $R^2 = 90\%$					
BE	NONE	1	1	0	1	0	3.1	0.025
	LASSO	1	1	0	1	0	3.1	0.025
	SCAD	1	1	0	1	0	3.1	0.025
	MCP	1	1	0	1	0	3.1	0.025
FR	NONE	1	1	0	1	0	3.1	0.025
	LASSO	1	1	0	1	0	3.1	0.025
	SCAD	1	1	0	1	0	3.1	0.025
	MCP	1	1	0	1	0	3.1	0.025
SIS	LASSO	1	1	0.004	0.996	0	37	0.919
	SCAD	1	1	0.004	0.996	0	37	0.919
	MCP	1	1	0.004	0.996	0	37	0.919
ISIS	LASSO	1	1	0.004	0.996	0	37	0.919
	SCAD	1	1	0.004	0.996	0	37	0.919
	MCP	1	1	0.004	0.996	0	37	0.919

Table 4.3 and 4.4, represents the simulation result for low signal-to-noise (i.e.,Theoretical R^2=30%) and high signal-to-noise (i.e.,Theoretical R^2=90%). Similar to Example 1, increasing the signal-to-noise improves the BE method's performance.

Table 4.5: Example 3: (n, d, d_0)=(75,5000,3) and R^2=30%

Screening Method	Selection Method	Coverage Probability	True Positive Rate (%)	False Positive Rate (%)	True Negative Rate (%)	False Negative Rate (%)	Model Size	False Discovery Rate (%)
			Theoretical $R^2 = 30\%$					
BE	NONE	0.267	0.267	0.001	0.999	0.733	6.2	0.855
	LASSO	0.267	0.267	0.001	0.999	0.733	6.2	0.855
	SCAD	0.267	0.267	0.001	0.999	0.733	6.2	0.855
	MCP	0.267	0.267	0.001	0.999	0.733	6.2	0.855
FR	NONE	0.333	0.333	0.001	0.999	0.667	5.2	0.796
	LASSO	0.333	0.333	0.001	0.999	0.667	5.2	0.796
	SCAD	0.333	0.333	0.001	0.999	0.667	5.2	0.796
	MCP	0.333	0.333	0.001	0.999	0.667	5.2	0.796
SIS	LASSO	1	1	0.003	0.997	0	17	0.824
	SCAD	1	1	0.003	0.997	0	17	0.824
	MCP	1	1	0.003	0.997	0	17	0.824
ISIS	LASSO	1	1	0.003	0.997	0	17	0.824
	SCAD	1	1	0.003	0.997	0	17	0.824
	MCP	1	1	0.003	0.997	0	17	0.824

Table 4.6: Example 3 :(n, d, d_0)=(75,5000,3) and R^2=90%

Screening Method	Selection Method	Coverage Probability	True Positive Rate (%)	False Positive Rate (%)	True Negative Rate (%)	False Negative Rate (%)	Model Size	False Discovery Rate (%)
			Theoretical $R^2 = 90\%$					
BE	NONE	1	1	0	1	0	3.5	0.115
	LASSO	1	1	0	1	0	3.5	0.115
	SCAD	1	1	0	1	0	3.5	0.115
	MCP	1	1	0	1	0	3.5	0.115
FR	NONE	1	1	0	1	0	3.7	0.155
	LASSO	1	1	0	1	0	3.7	0.155
	SCAD	1	1	0	1	0	3.7	0.155
	MCP	1	1	0	1	0	3.7	0.155
SIS	LASSO	1	1	0.003	0.997	0	17	0.824
	SCAD	1	1	0.003	0.997	0	17	0.824
	MCP	1	1	0.003	0.997	0	17	0.824
ISIS	LASSO	1	1	0.003	0.997	0	17	0.824
	SCAD	1	1	0.003	0.997	0	17	0.824
	MCP	1	1	0.003	0.997	0	17	0.824

Tables 4.5 and 4.6 are the simulation results based on a low and high signal-to-noise. Increasing the signal to noise improves the TPR, the Model size and minimizes the FDR of the BE method.

Table 4.7: Example 4: (n, d, d_0)=(300,5000,5) and R^2=30%

Screening Method	Selection Method	Coverage Probability	True Positive Rate (%)	False Positive Rate (%)	True Negative Rate (%)	False Negative Rate (%)	Model Size	False Discovery Rate (%)
			Theoretical $R^2 = 30\%$					
BE	NONE	0.14	0.14	0	1	0.86	1.4	0.65
	LASSO	0.14	0.14	0	1	0.86	1.7	0.5
	SCAD	0.14	0.14	0	1	0.86	1.4	0.65
	MCP	0.14	0.14	0	1	0.86	1.4	0.65
FR	NONE	0.14	0.14	0	1	0.86	1.5	0.683
	LASSO	0.14	0.14	0	1	0.86	1.8	0.533
	SCAD	0.14	0.14	0	1	0.86	1.5	0.683
	MCP	0.14	0.14	0	1	0.86	1.5	0.683
SIS	LASSO	1	1	0.009	0.991	0	52	0.904
	SCAD	1	1	0.009	0.991	0	52	0.904
	MCP	1	1	0.009	0.991	0	52	0.904
ISIS	LASSO	1	1	0.009	0.991	0	52	0.904
	SCAD	1	1	0.009	0.991	0	52	0.904
	MCP	1	1	0.009	0.991	0	52	0.904

Table 4.8: Example 4: (n, d, d_0)=(300,5000,5) and R^2=90%

Screening Method	Selection Method	Coverage Probability	True Positive Rate (%)	False Positive Rate (%)	True Negative Rate (%)	False Negative Rate (%)	Model Size	False Discovery Rate (%)
			Theoretical $R^2 = 90\%$					
BE	NONE	0.22	0.22	0	1	0.78	2.5	0.55
	LASSO	0.22	0.22	0	1	0.78	2.5	0.55
	SCAD	0.22	0.22	0	1	0.78	2.5	0.55
	MCP	0.22	0.22	0	1	0.78	2.5	0.55
FR	NONE	0.22	0.22	0	1	0.78	2.5	0.55
	LASSO	0.22	0.22	0	1	0.78	2.5	0.55
	SCAD	0.22	0.22	0	1	0.78	2.5	0.55
	MCP	0.22	0.22	0	1	0.78	2.5	0.55
SIS	LASSO	0.98	0.98	0.009	0.991	0.02	51.9	0.906
	SCAD	0.9	0.9	0.009	0.991	0.1	51.9	0.913
	MCP	0.92	0.92	0.009	0.991	0.08	51.9	0.911
ISIS	LASSO	0.98	0.98	0.009	0.991	0.02	51.9	0.906
	SCAD	0.9	0.9	0.009	0.991	0.1	51.9	0.913
	MCP	0.92	0.92	0.009	0.991	0.08	51.9	0.911

From Table 4.1 and 4.2, In the case of extreme correlation, increasing the signal-to-noise doesn't improve on the performance of our method as well as the FR significantly. Our method doesn't perform well under this scenario.

Table 4.9: Example 5: (n, d, d_0)=(200,5000,8) and R^2=30%

Screening Method	Selection Method	Coverage Probability	True Positive Rate (%)	False Positive Rate (%)	True Negative Rate (%)	False Negative Rate (%)	Model Size	False Discovery Rate (%)
			Theoretical $R^2 = 30\%$					
BE	NONE	0.225	0.225	0.001	0.999	0.775	7.2	0.682
	LASSO	0.225	0.225	0.001	0.999	0.775	7.2	0.682
	SCAD	0.225	0.225	0.001	0.999	0.775	7.2	0.682
	MCP	0.225	0.225	0.001	0.999	0.775	7.2	0.682
FR	NONE	0.225	0.225	0.001	0.999	0.775	4.3	0.553
	LASSO	0.225	0.225	0.001	0.999	0.775	4.3	0.553
	SCAD	0.225	0.225	0.001	0.999	0.775	4.3	0.553
	MCP	0.225	0.225	0.001	0.999	0.775	4.3	0.553
SIS	LASSO	1	1	0.006	0.994	0	37	0.784
	SCAD	1	1	0.006	0.994	0	37	0.784
	MCP	1	1	0.006	0.994	0	37	0.784
ISIS	LASSO	1	1	0.006	0.994	0	37	0.784
	SCAD	1	1	0.006	0.994	0	37	0.784
	MCP	1	1	0.006	0.994	0	37	0.784

Table 4.10: Example 5: (n, d, d_0)=(200,5000,8) and R^2=90%

Screening Method	Selection Method	Coverage Probability	True Positive Rate (%)	False Positive Rate (%)	True Negative Rate (%)	False Negative Rate (%)	Model Size	False Discovery Rate (%)
			Theoretical $R^2 = 90\%$					
BE	NONE	0.975	0.975	0	1	0.025	8.4	0.065
	LASSO	0.975	0.975	0	1	0.025	8.4	0.065
	SCAD	0.975	0.975	0	1	0.025	8.4	0.065
	MCP	0.975	0.975	0	1	0.025	8.4	0.065
FR	NONE	0.975	0.975	0	1	0.025	8.4	0.065
	LASSO	0.975	0.975	0	1	0.025	8.4	0.065
	SCAD	0.975	0.975	0	1	0.025	8.4	0.065
	MCP	0.975	0.975	0	1	0.025	8.4	0.065
SIS	LASSO	1	1	0.006	0.994	0	37	0.784
	SCAD	0.975	0.975	0.006	0.994	0.025	37	0.789
	MCP	0.962	0.962	0.006	0.994	0.038	37	0.792
ISIS	LASSO	1	1	0.006	0.994	0	37	0.784
	SCAD	0.975	0.975	0.006	0.994	0.025	37	0.789
	MCP	0.962	0.962	0.006	0.994	0.038	37	0.792

Tables 4.9 and 4.10 is quite similar to Tables 4.1 and 4.2. This is because Example 1 was replicated however under different conditions.

4.4 Real Data Application

The modified BE method was used to classify cancer samples as well as for gene selection for 3 types of cancer. Gene expression data used in this work were retrieved from Gene Expression Omnibus (GEO). Three types of common cancers including colon cancer, prostate cancer, and leukemia were selected and the raw data were downloaded. These gene expression data sets are described below:

Colon cancer (GSE44861) : This data set included 111 samples and 22277 genes that data were divided into 2 groups of 55 control and 56 cancer samples.

Prostate cancer (GSE71783) : This data set consisted of 30 samples and 17881 genes that data were divided into 2 groups of 15 control and 15 cancer samples.

Leukemia (GSE9476): This dataset contained 64 samples and 22283 genes that data were divided into 2 groups of 38 control and 26 cancer samples.

Table 4.11: The Number of Genes and Samples Used for the Considered Cancers

Data Set	Gene	Sample(+/-)	Class
Colon Cancer	22277	111(56/55)	No tumor/tumor
Prostate Cancer	17881	30(15/15)	Normal/tumor
Leukemia	22283	64(26/38)	Normal/leukemia

These data sets are examples of high dimensional data where the number of predictors, p, is much greater than the number of observations, n. The data set is an already prepared data set thus the gene expressions are normalized with no missing values.

In order to access the prediction accuracy, we employed all the methods used by the BE method (i.e.BE-NONE,BE-LASSO,BE-SCAD,BE-MCP). We apply the entire data set to the Modified BE algorithm and the screened variables in the sub-best model with the minimum EBIC value becomes our final selected variables. We further apply these variables to the shrinkage methods (LASSO, SCAD and MCP) used in the simulation study.

The selected models are refitted via ordinary least squares for prediction purposes. We evaluate the prediction accuracy by the Area Under the Receiver Operating Characteristic curve (AUROC) which is simply a plot of the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.
It is noteworthy that all methods (BE-NONE, LASSO, SCAD, MCP) selected the same variables hence we report only these selected variables for the three data set.

Figure 1 shows the AUROC plots for the three data set. For the Colon data set:based on BE, the LASSO, SCAD, and MCP identified 2 genes (Probe ID: 205697_at (SCGN), 206871_at (COL5A1)) as relevant. For the Prostate data set: based on BE, the LASSO, SCAD, and MCP identified 3 genes (Probe ID: 2777714 (SNCA), 3145980 (HRSP12), 2363484(PPOX)) as important. For the Leukemia data set: based on BE, the LASSO, SCAD, and MCP identified importance 1 gene (Probe ID: 210976_s_at(PFKM)). From the figure 4.1, the AUROC score is 0.932, 0.964 and 0.867 which shows a good prediction accuracy of the proposed BE method. The AUC score, 0.867, reported for the Leukemia data set is not surprising because a regression model with more than one variable improves its prediction accuracy.

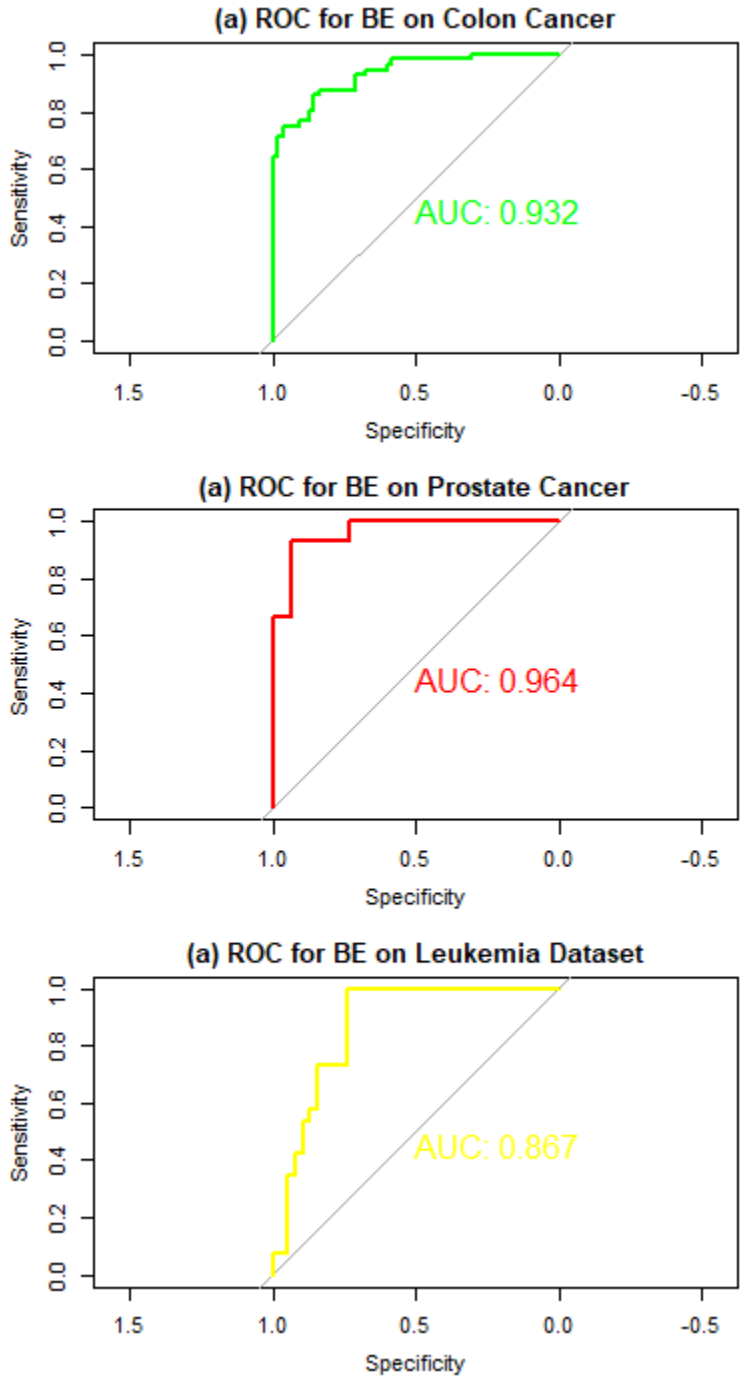


Figure 4.1: AUROC plots of using the BE on Colon, Prostate and Leukemia data sets, respectively.

Tables 4.12, 4.13, and 4.14 show the gene set selected by the BE method for the different data sets and its functions.

Table 4.12: Selected gene set from Leukemia Data by BE methods

Probe ID	Gene Symbol	Function
210976_*s_at*	PFKM	Three phosphofructokinase isozymes exist in humans: muscle, liver and platelet. These isozymes function as subunits of the mammalian tetramer phosphofructokinase, which catalyzes the phosphorylation of fructose-6-phosphate to fructose-1,6-bisphosphate. Tetramer composition varies depending on tissue type. This gene encodes the muscle-type isozyme. Mutations in this gene have been associated with glycogen storage disease type VII, also known as Tarui disease.

Table 4.13: Selected gene set from Colon Data by BE methods

Probe ID	Gene Symbol	Function
205697_*at*	SCGN	SCGN, also known as secretagogin, is a cytoplasmic protein that contains six EF-hand domains and is related to the calicium-binding proteins Calretinin and Calbindin D28K. This protein is thought to be involved in cell proliferation and KCl (potassium chloride)-mediated calcium flux events. Through its interaction with KCl and its subsequent ability to modulate calcium storage pools within the cell, SCGN may function to negatively control growth and differentiation rates and, thus, indirectly inhibit cell replication. Recombinant SCGN protein, fused to His-tag at N-terminus, was expressed in E.coli and purified by using conventional chromatography techniques.
212489_*at*	COL5A1	This gene encodes an alpha chain for one of the low abundance fibrillar collagens. Fibrillar collagen molecules are trimers that can be composed of one or more types of alpha chains. Type V collagen is found in tissues containing type I collagen and appears to regulate the assembly of heterotypic fibers composed of both type I and type V collagen. This gene product is closely related to type XI collagen and it is possible that the collagen chains of types V and XI constitute a single collagen type with tissue-specific chain combinations. The encoded procollagen protein occurs commonly as the heterotrimer pro-alpha1(V)-pro-alpha1(V)-pro-alpha2(V). Mutations in this gene are associated with Ehlers-Danlos syndrome, types I and II. Alternative splicing of this gene results in multiple transcript variants.

Table 4.14: Selected gene set from Prostate Data by BE methods

Probe ID	Gene Symbol	Function
2777714	SNCA	Alpha-synuclein is a member of the synuclein family, which also includes beta and gamma synuclein. Among its related pathways are transport to the Golgi and subsequent modification and EGFR1 signaling pathway.
3145980	HRSP12	HRSP12, also called ribonuclease uK114, is an endoribonuclease found predominantly in human adult kidney and liver, and which is responsible for inhibiting translation by cleaving mRNA. This protein cleaves phosphodiester bonds only in single-stranded RNA. It may be an important biomarker for heptatic carcinoma. Recombinant human HRSP12 protein, fused to His-tag at N-terminus, was expressed in E.coli and purified by using conventional chromatography.
2363484	PPOX	This gene encodes the penultimate enzyme of heme biosynthesis, which catalyzes the 6-electron oxidation of protoporphyrinogen IX to form protoporphyrin IX. Mutations in this gene cause variegate porphyria, an autosomal dominant disorder of heme metabolism resulting from a deficiency in protoporphyrinogen oxidase, an enzyme located on the inner mitochondrial membrane. Alternatively spliced transcript variants encoding the same protein have been identified.

Finally, we report on the boxplot of differentially expressed gene for normal and tumor samples for all three data sets.

From figure 4.2a, the selected gene (SNCA) is under expressed in Cancer tumor samples than normal samples. This means this gene is deactivated in cancer samples. The same applies to the other selected genes. However, the expression levels are significantly differentially expressed in figure 4.2a between normal and tumor cancer samples. Similarly, from figure 4.3, gene ID 205697_at (SCGN) is lowly differentially expressed in tumor cancer samples than for no tumor samples. However, the gene ID 212489_at (COL5A1) is over expressed in colon cancer samples. This means it is highly activated in tumor samples. Finally, for the Leukemia data set, the gene ID 210976_s_at is overly-expressed in normal samples than in leukemia samples.

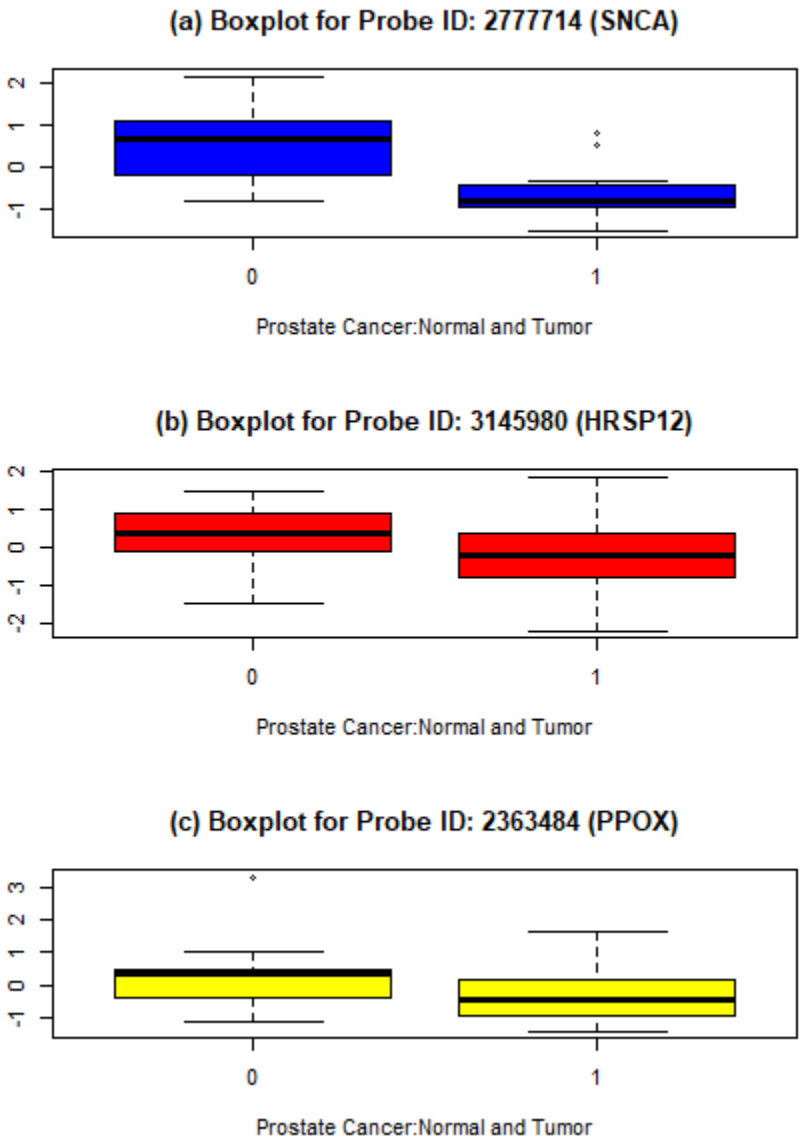
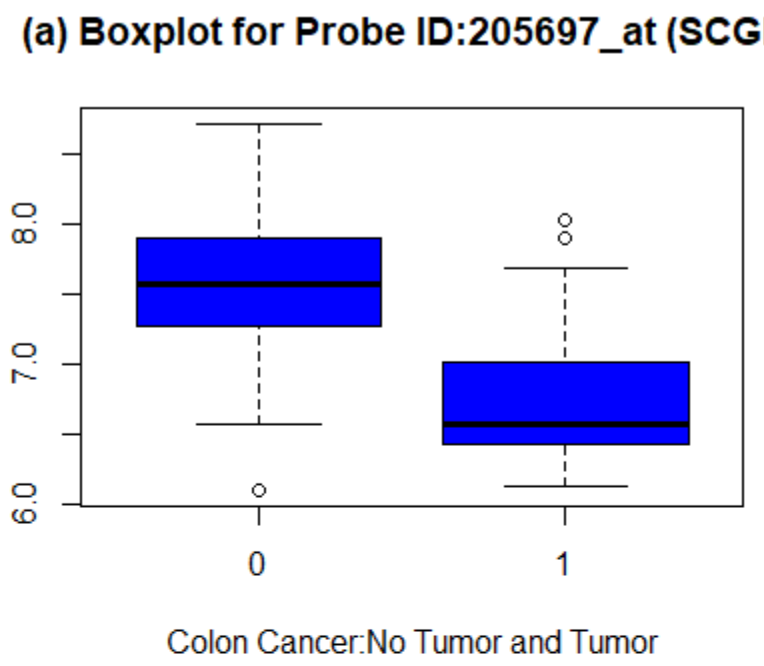


Figure 4.2: Box plot for comparing differentially expressed gene for Normal and Tumor Prostate cancer samples



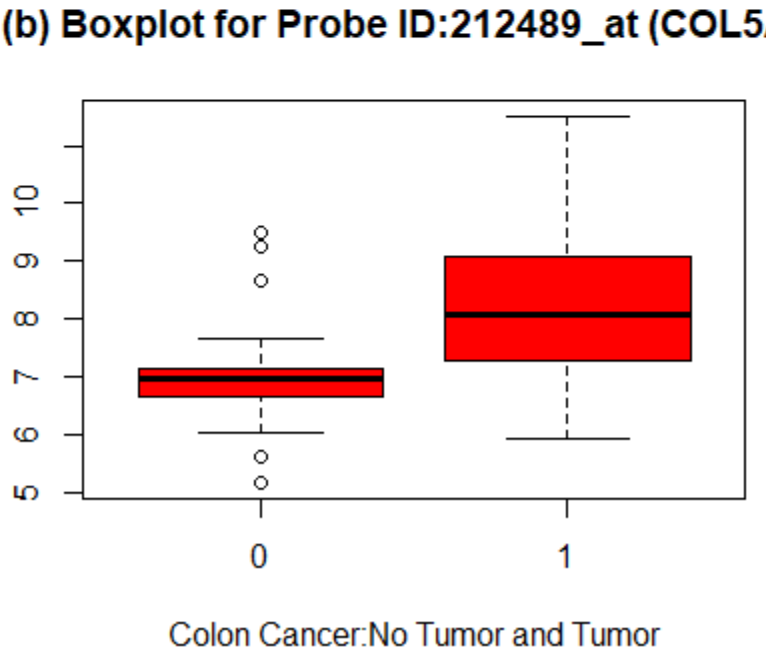


Figure 4.3: Box plot for comparing differentially expressed gene for No Tumor and Tumor colon cancer samples

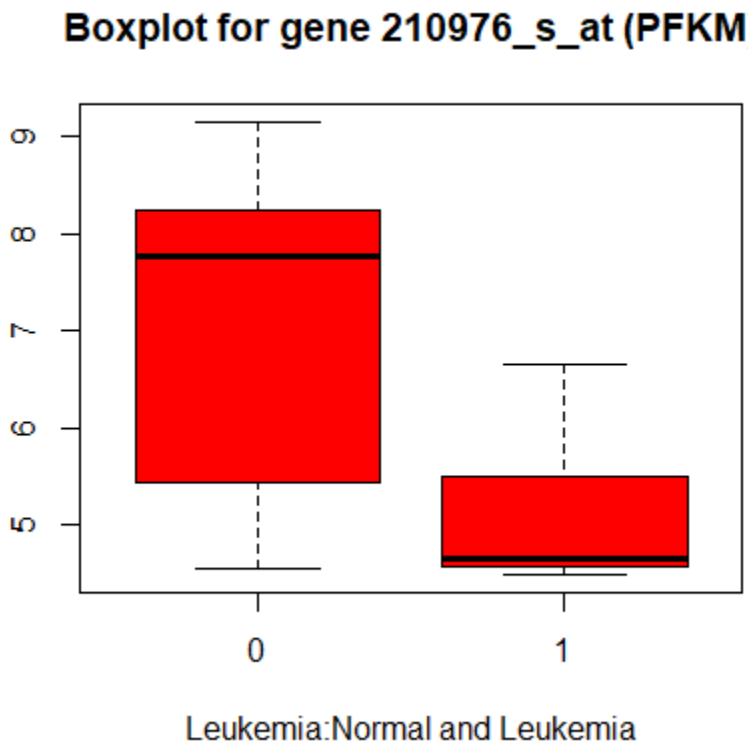


Figure 4.4: Box plot for comparing differentially expressed gene for Normal and Tumor Leukemia samples

Chapter 5

Discussion and Conclusion

5.1 Summary

In genomic studies, we encounter a number of genes which is usually much larger than the sample size. Biologically, only a small subset of these genes are related to a disease under study. Most of the genes are unrelated and have noise which can greatly influence the performance of classification. Therefore, choosing at least a minimal gene list is one of the most important applications in the analysis of omic data, which are effective in complex diseases. The essence of variable screening is to screen out these genes which might be unrelated to the disease under study greatly reducing the dimensionality of the omic data. These screened variables are further applied to various shrinkage methods to further select those genes that are important or related the disease under study. The aim is to select a subset of useful and appropriate genes to diagnose the disease among the whole genes. This improves the accuracy of classification.

It has been shown extensively from numerical studies that the Modified BE algorithm can be a very useful variable screening method to discover all relevant predictors, even if the number of predictors is larger than the sample size. However, we do not claim the Modified BE as the only good variable screening method. From the simulation studies, we observed that the SIS and ISIS performed well under all conditions with a much larger model size and FDR.

The aim of this study was to investigate the BE method as a variable screening in a high dimensional set up with motivation from H.Wang's (2009) FR paper and the performance has been very encouraging especially under high signal-to-noise.

5.2 Recommendations for Future Work

For recommendations for future work, we first of all address the limitations and strengths of our proposed method and then we propose a possible solution as future work.

1. In the case of small n sample size, the BE selects a small subset of variables which are consistent or the same across all methods used under BE thus BE-NONE, BE-LASSO, BE-SCAD and BE-MCP.

We propose this method in the case of large n sample size (e.g. n=300,400, etc.). This is because our proposed BE method can be a good method for filtering. In this case, more variables will be selected in the screening stage and then further applied to various shrinkage methods (LASSO, SCAD and MCP) for final selection.

2. The variables selected by the BE method cannot guarantee error control. Therefore as future work, we wish to develop a methodology for our BE method with error control.

References

[1] Bogdan,M., Doerge, R.,& Ghosh, J.K.(2004). Modifying the Schwarz Bayesian information criterion to locate multiple interacting quantitative trait loci. *Genetics* 167, 989-99

[2] Broman, K. W. & Speed, T.P. (2002). A model selection approach for the identi- cation of quantitative trait loci in experimental crosses. J. R. Statist. Soc. B 64, 641-56.

[3] Chen, J. & Chen, Z. (2008). Extended Bayesian information criterion for model selection with large model space. *Biometrika* ,94, 759-771

[4] Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics*, 32, 407-489.

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

[6] Fan, J., Feng, Y. and Song, R. (2011). Nonparametric independence screening in sparse ultra-high dimensional additive models. *Journal of American Statistical Association*, 116,544-557.

[7] Fan, J. & Li, R. (2001). Variable selection via non-concave penalized likelihood and its oracle properties. *Journal of the American Statistical Association* 96, 1348-1360.

[8] Fan, J. and Lv, J. (2008). Sure independence screening for ultra-high dimensional feature space (with discussion). *Journal of the Royal Statistical Society, Series* B, 70, 849-911.

[9] He, X., Wang, L. and Hong, H. G. (2013). Quantile-adaptive model-free variable screening for high-dimensional heterogeneous data. DOI:10.1214/13-AOS1087SUPP.

[10] Hunter, D. R. and Li, R. (2005). Variable selection using MM algorithms. *The Annals of Statistics*, 33, 1617-1642.

[11] Ing. C. and T. L. Lai (2011). A stepwise regression model and consistent model selection for high-dimensional sparse linear model. *Statistica Sinica* 21, 1473-1513

[12] Jia, J. and Yu, B. (2008). On model selection consistency of the elastic net when p ¿¿ n,*Unpublished Manuscript, Department of Statistics, UC-Berkeley.*

[13] Kong, E. and Xia, Y. (2007). Variable selection for single-index model, *Biometrika, 94, 217-229.*

[14] Leng, C., Lin, Y., and Wahba, G. (2006). A note on lasso and related procedures in model selection,*Statistica Sinica, 16, 1273-1284.*

[15] M.H.Kutner(2004) Applied Linear Statistical Models *McGraw-Hill/lrwin series operations and decision sciences.*

[16] Sangjin, K. and Halabi, S. (2016). High Dimensional Variable Selection with Error Control. *BioMed Research International Volume 2016, Article ID 8209453, 11 pages*

[17] Shirin.S, Faria, S., & A. Manuela Gonalves (2015) Variable Selection Methods in High-dimensional RegressionA Simulation Study, Communications in Statistics - Simulation and Computation, 44:10, 2548-2561, DOI: 10.1080/03610918.2013.833231

[18] Siegmund, D. (2004). Model selection in irregular problems: Application to mapping quantitative trait loci. *Biometrika* 91, 785-800.

[19] Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society, Series* B 58, 267-88.

[20] Wang, H. (2009). Forward regression for ultra-high dimensional variable screening. *Journal of the American Statistical Association*, 104, 1512-1524.

[21] Yuan, M. and Lin, Y. (2007). On the nonnegative garrote estimator, *Journal of the Royal Statistical Society, Series* B, 69, 143-161.

[22] Zhang, C. (2010). Nearly unbiased variable selection under minimax concave penalty. *Annals of Statistics Volume* 38, 894-942.

[23] Zhang, C. H. and Huang, J. (2008). The sparsity and bias of the lasso selection in high-dimensional linear regression, *The Annals of Statistics*, 36, 1567-1594.

[24] Zhang, H. H. and Lu, W. (2007). Adaptive lasso for Cox's proportional hazard model, *Biometrika*, 94, 691-703.

[25] Zhao, P. and Yu, B. (2006). On model selection consistency of lasso, *Journal of Machine Learning Research*, 7, 2541-2567.

[26] Zou, H. and Hastie, T. (2005). Regression shrinkage and selection via the elastic net with application to microarrays, *Journal of the Royal Statistical Society, Series* B, 67, 301-320.

[27] Zou, H. and Li, R. (2008). One-step sparse estimates in nonconcave penalized likelihood models (with discussion), *The Annals of Statistics*, 36, 1509-1533.

[28] Zou, H. and Zhang, H. H. (2008). On the adaptive elastic-net with a diverging number of parameters,*The Annals of Statistics, To appear*.

Appendix

Simulation Results for Theoretical $R^2 = 60\%$ The Tables for the simulation results based on the Theoretical $R^2 = 60\%$ are reported in the appendix.

Table 5.1: Example 1: (n, d, d_0)=(200,5000,8) and R^2=60%

Screening Method	Selection Method	Coverage Probability	True Positive Rate (%)	False Positive Rate (%)	True Negative Rate (%)	False Negative Rate (%)	Model Size	False Discovery Rate (%)
			Theoretical $R^2 = 60\%$					
BE	NONE	0.788	0.788	0	1	0.212	8.4	0.221
	LASSO	0.788	0.788	0	1	0.212	8.4	0.221
	SCAD	0.788	0.788	0	1	0.212	8.4	0.221
	MCP	0.788	0.788	0	1	0.212	8.4	0.221
FR	NONE	0.762	0.762	0	1	0.238	8.1	0.24
	LASSO	0.762	0.762	0	1	0.238	8.1	0.24
	SCAD	0.762	0.762	0	1	0.238	8.1	0.24
	MCP	0.762	0.762	0	1	0.238	8.1	0.24
SIS	LASSO	0.988	0.988	0.006	0.994	0.012	37	0.786
	SCAD	1	1	0.006	0.994	0	37	0.784
	MCP	1	1	0.006	0.994	0	37	0.784
ISIS	LASSO	0.988	0.988	0.006	0.994	0.012	37	0.786
	SCAD	1	1	0.006	0.994	0	37	0.784
	MCP	1	1	0.006	0.994	0	37	0.784

Table 5.2: Example 2: (n, d, d_0)=(200,8000,3) and R^2=60%

Screening Method	Selection Method	Coverage Probability	True Positive Rate (%)	False Positive Rate (%)	True Negative Rate (%)	False Negative Rate (%)	Model Size	False Discovery Rate (%)
			Theoretical $R^2 = 60\%$					
BE	NONE	0.867	0.867	0	1	0.133	3.2	0.183
	LASSO	0.867	0.867	0	1	0.133	3.2	0.183
	SCAD	0.867	0.867	0	1	0.133	3.2	0.183
	MCP	0.867	0.867	0	1	0.133	3.2	0.183
FR	NONE	0.8	0.8	0	1	0.2	3.2	0.242
	LASSO	0.8	0.8	0	1	0.2	3.2	0.242
	SCAD	0.8	0.8	0	1	0.2	3.2	0.242
	MCP	0.8	0.8	0	1	0.2	3.2	0.242
SIS	LASSO	1	1	0.004	0.996	0	37	0.919
	SCAD	0.967	0.967	0.004	0.996	0.033	37	0.922
	MCP	0.967	0.967	0.004	0.996	0.033	37	0.922
ISIS	LASSO	1	1	0.004	0.996	0	37	0.919
	SCAD	0.967	0.967	0.004	0.996	0.033	37	0.922
	MCP	0.967	0.967	0.004	0.996	0.033	37	0.922

Table 5.3: Example 3: (n, d, d_0)=(75,5000,3) and R^2=60%

Screening Method	Selection Method	Coverage Probability	True Positive Rate (%)	False Positive Rate (%)	True Negative Rate (%)	False Negative Rate (%)	Model Size	False Discovery Rate (%)
			Theoretical $R^2 = 60\%$					
BE	NONE	0.667	0.667	0.001	0.999	0.333	5.7	0.633
	LASSO	0.667	0.667	0.001	0.999	0.333	5.7	0.633
	SCAD	0.667	0.667	0.001	0.999	0.333	5.7	0.633
	MCP	0.667	0.667	0.001	0.999	0.333	5.7	0.633
FR	NONE	0.7	0.7	0.001	0.999	0.3	5.1	0.568
	LASSO	0.7	0.7	0.001	0.999	0.3	5.1	0.568
	SCAD	0.7	0.7	0.001	0.999	0.3	5.1	0.568
	MCP	0.7	0.7	0.001	0.999	0.3	5.1	0.568
SIS	LASSO	1	1	0.003	0.997	0	17	0.824
	SCAD	1	1	0.003	0.997	0	17	0.824
	MCP	1	1	0.003	0.997	0	17	0.824
ISIS	LASSO	1	1	0.003	0.997	0	17	0.824
	SCAD	1	1	0.003	0.997	0	17	0.824
	MCP	1	1	0.003	0.997	0	17	0.824

Table 5.4: Example 4: (n, d, d_0)=(300,5000,5) and R^2=60%

Screening Method	Selection Method	Coverage Probability	True Positive Rate (%)	False Positive Rate (%)	True Negative Rate (%)	False Negative Rate (%)	Model Size	False Discovery Rate (%)
			Theoretical $R^2 = 60\%$					
BE	NONE	0.18	0.18	0	1	0.82	2	0.583
	LASSO	0.18	0.18	0	1	0.82	2.1	0.533
	SCAD	0.18	0.18	0	1	0.82	2	0.583
	MCP	0.18	0.18	0	1	0.82	2	0.583
FR	NONE	0.18	0.18	0	1	0.82	2	0.583
	LASSO	0.18	0.18	0	1	0.82	2.1	0.533
	SCAD	0.18	0.18	0	1	0.82	2	0.583
	MCP	0.18	0.18	0	1	0.82	2	0.583
SIS	LASSO	0.98	0.98	0.009	0.991	0.02	52	0.906
	SCAD	0.98	0.98	0.009	0.991	0.02	52	0.906
	MCP	0.98	0.98	0.009	0.991	0.02	52	0.906
ISIS	LASSO	0.98	0.98	0.009	0.991	0.02	52	0.906
	SCAD	0.98	0.98	0.009	0.991	0.02	52	0.906
	MCP	0.98	0.98	0.009	0.991	0.02	52	0.906

Table 5.5: Example 5 : (n, d, d_0)=(200,5000,8) and R^2=60%

Screening Method	Selection Method	Coverage Probability	True Positive Rate (%)	False Positive Rate (%)	True Negative Rate (%)	False Negative Rate (%)	Model Size	False Discovery Rate (%)
			Theoretical $R^2 = 60\%$					
BE	NONE	0.838	0.838	0	1	0.162	8.9	0.246
	LASSO	0.838	0.838	0	1	0.162	8.9	0.246
	SCAD	0.838	0.838	0	1	0.162	8.9	0.246
	MCP	0.838	0.838	0	1	0.162	8.9	0.246
FR	NONE	0.85	0.85	0	1	0.15	8.6	0.206
	LASSO	0.85	0.85	0	1	0.15	8.6	0.206
	SCAD	0.85	0.85	0	1	0.15	8.6	0.206
	MCP	0.85	0.85	0	1	0.15	8.6	0.206
SIS	LASSO	1	1	0.006	0.994	0	37	0.784
	SCAD	1	1	0.006	0.994	0	37	0.784
	MCP	1	1	0.006	0.994	0	37	0.784
ISIS	LASSO	1	1	0.006	0.994	0	37	0.784
	SCAD	1	1	0.006	0.994	0	37	0.784
	MCP	1	1	0.006	0.994	0	37	0.784

R Codes

```
#################################################################
# SIMULATING FROM THE MULTIVARIATE NORMAL DISTRIBUTION WITH    #
#MEAN 0 AND THE COVARIANCE MATRIX SPECIFIED USING SIX MODELS   #
#################################################################
#Variable selection method
####################################
# MODEL 1: INDEPENDENT PREDICTORS  #
####################################
#Independent Predictors with standard multivariate normal distribution
library(MASS)
library(mvtnorm)
pred1=function(n,p)
{
set.seed(sample(1:1000000000,1))
s=diag(p)
predictors=rmvnorm(n,mean=rep(0,p),sigma=s,method="chol")
return (predictors)
}
#################################################################
# MODEL 2: COMPOUND SYMMETRY: PREDICTORS ARE EQUALLY CORRELATED #
# WITH CORRELATION SPECIFIED AS P=0.3,0.6 OR 0.9                #
#################################################################
#Compound Symmetry model
pred2=function(n,p,rho)
```

```
{
set.seed(sample(1:1000000000,1))
s=diag(p)
s[lower.tri(s)]=s[upper.tri(s)]=rho
predictors=rmvnorm(n,mean=rep(0,p),sigma=s,method="chol")
return(predictors)
}
#####################################################################
# MODEL 3: AUTOREGRESSIVE CORRELATION: THISCORRELATION STRUCTURE#
# ARISES WHEN THE PREDICTORS ARE NATURALLY ORDERED.              #
# THE CORRELATION IS SPECIFIED AS P=0.3,0.6,0.9                  #
#####################################################################
#Autoregressive Correlation model
pred3=function(n,p)
{
set.seed(sample(1:1000000000,1))
s=diag(p)
#enter values in upper triangular
for(i in 1:ncol(s))
{
for(j in 1:nrow(s))
{
if (i==j)next
s[i,j]=0.5^(abs(i-j))
}
}
predictors=rmvnorm(n,mean=rep(0,p),sigma=s,method="chol")
return(predictors)
```

```
}
#####################################################################
#MODEL 4:A CHALLENGING CASE                                         #
#####################################################################
#Challenging case
pred4=function(n,p)
{
set.seed(sample(1:1000000000,1))
w=matrix(rnorm(n*5),nrow=n,ncol=5)
z=matrix(rnorm(n*5),nrow=n,ncol=5)
x=(z+w)/sqrt(2)
x1=matrix(NA,nrow=n,ncol=(p-5))
sw=apply(w,1,sum)
for(i in 1:(p-5))
{
tmp.z=(rnorm(n)+sw)/2
x1[,i]=tmp.z
}
predictors=cbind(x,x1)
return(predictors)
}
#############################################################
#MODEL 5: NORMALITY ASSUMPTION                             ##
#############################################################
#Predictors with standard exponential distribution
#(p,n,p0)=(10000,200,8)
library(stats)
pred5=function(n,p)
```

```
{
set.seed(sample(1:1000000000,1))
sds=1
predictors=matrix(rexp(n*p,sds),nrow=n,ncol=p)
return (predictors)
}
#############################################################
#MODEL 6: DIVERGING MODEL SIZE                              ##
#############################################################
#Predictors with standard exponential distribution
#(p,n,p0)=(10000,sqrt(n))where n=200,400 and 800
# pred6=function(n,p)
# {
#   set.seed(sample(1:1000000000,1))
#   s=diag(p)
#   predictors=rmvnorm(n,mean=rep(0,p),sigma=s,method="chol")
#   return (predictors)
# }

#Generate beta for those variables
#beta_i=(-1)^ui*(|N(0,1)|+4log (n)/sqrt(n)),where ui~Ber(0.4)
library(LaplacesDemon)
pred1_beta=function(n,nsp)
{
ui=rbinom(nsp,size=c(0,1),prob=0.4) #nsp is the number of true variables
rn=rnorm(nsp)
betas=((-1)^(ui))*(abs(rn)+((4*log(n))/sqrt(n)))
return(betas)
```

```
}

pred5_beta=function(n,nsp)
{
beta=NULL
ui=rbinom(nsp,size=c(0,1),prob=0.4)
rn=rnorm(nsp)
betas=((-1)^(ui))*(abs(rn)+((4*log(n))/sqrt(n)))
return(betas)
}

# library(stats)
# pred6_beta=function(n,nsp) #The number of significant predictors(nsp)
# {
#   beta=NULL
#   ui=rbinom(nsp,size=c(0,1),prob=0.4)
#   rn=rnorm(nsp)
#   betas=((-1)^(ui))*(abs(rn)+((4*log(n))/sqrt(n)))
#   return(betas)
# }

###ERROR
#Calculating the error based on the variance of X%*%(betas)
#and R-squared(signal-to-noise)
error1=function(n,x,betas,R2)
{
set.seed(sample(1:1000000000,1))
sd.pred=sqrt((var(x%*%(betas))/R2)-var(x%*%(betas)))
```

```
error=rnorm(n,mean=0,sd=sd.pred)
return(error)
}


#The function below is not generating the data and because of this, i can't check
#the other functions are working since it depends on the data.
#the error message below is what i get when i try to work it out:

#Error in data_generation(options = "pred1", R2 = 0.3, rho = 0) :
#dims [product 200] do not match the length of object [1000]
#In addition: Warning messages:
#  1: In var(x %*% (betas))/R2 :
#  Recycling array of length 1 in array-vector arithmetic is deprecated.
#Use c() or as.vector() instead.

#2: In (var(x %*% (betas))/R2) - var(x %*% (betas)) :
#  Recycling array of length 1 in vector-array arithmetic is deprecated.
#Use c() or as.vector() instead.
##Combining all the examples
data_generation = function(options,R2)
{
if(options=="pred1")
{
n=200;p=5000;ntv=8
gamm = 1 # for EBIC arguments   gamm>=0
betas=pred1_beta(n,nsp=ntv)
#true variable name
```

```
indx=1:ntv
true.v=paste("X",indx,sep="")
x=pred1(n,p)
error=error1(n,x[,indx],betas,R2)
y=as.matrix(x[,indx])%*%as.matrix(betas)+error
}
if(options=="pred2")
{
#2.Case of pred2: Compound Symmetry
# 3 important variables to be selected
n=200;p=8000;rho=0.5;ntv=3
gamm = 1
betas=rep(5,ntv)
rho=0.5
#generate indexes of true vairable in vector
indx=1:ntv
true.v=paste("X",indx,sep="")
x=pred2(n,p,rho)
#sigma=1
error=error1(n,x[,indx],betas,R2)
y=as.matrix(x[,indx])%*%as.matrix(betas)+error
}
if(options=="pred3")
{
#3. Case of pred3: Autoregressive Correlation
#3 important variables to be selected
n=75;p=5000;ntv=3
gamm = 1
```

```
betas=c(3,1.5,2)
#generate indexes of true vairable in vector
indx=1:ntv
true.v=paste("X",indx,sep="")
x=pred3(n,p)

error=error1(n,x[,indx],betas,R2)
y=as.matrix(x[,indx])%*%as.matrix(betas)+error
}
if(options=="pred4")
{
n=300;p=5000;ntv=5
gamm = 1
betas = 2*(1:5)
#generate indexes of true vairable in vector
indx=1:ntv
true.v=paste("X",indx,sep="")
x=pred4(n,p)
error=error1(n,x[,indx],betas,R2)
y=as.matrix(x[,indx])%*%as.matrix(betas)+error
}
if(options=="pred5")
{
n=200;p=5000;ntv=8
gamm = 1
betas = pred5_beta(n,ntv);
#generate indexes of true vairable in vector
indx=1:ntv
```

```
true.v=paste("X",indx,sep="")
x=pred5(n,p)
error=error1(n,x[,indx],betas,R2)
y=as.matrix(x[,indx])%*%as.matrix(betas)+error
}
# if(options=="pred6")
# {
#   n=200;p=5000; ntv=as.integer(sqrt(n))
#   gamm = 1;R2=0.75
#   betas = pred6_beta(n,ntv);
#   #generate indexes of true vairable in vector
#   indx=1:ntv
#   true.v=paste("X",indx,sep="")
#   x=pred6(n,p)
#   #sigma=1
#   error=error1(n,x[,indx],betas,R2)
#   y=as.matrix(x[,indx])%*%as.matrix(betas)+error
# }
#name of column
colnames(x)=paste("X",1:ncol(x),sep="")
return(list(x,y,true.v))
}
###################################
########################################## SignifReg()
#install.packages("SignifReg")
library(SignifReg)

backward_elimination = function(x,y,gamm)
```

```
{
n = nrow(x)
newX=data.frame(y,x)
fullmodel <- lm(y ~ ., data =newX)
# add log(n)+2*gamm*log(ncol(X)) -> EBIC
model=step(fullmodel, direction ="backward",
k=log(n)+2*gamm*log(ncol(x)),trace=FALSE )
#model=names(model$coefficients)[-c(1)]
return(model)
}

forward_selection = function(x,y,gamm)
{
n = nrow(x)
newX=data.frame(y,x)
fit.null <- lm(y ~ 1, data=newX)
fullmodel <- lm(y ~ ., data=newX)
model=step(fit.null,scope=list(lower=fit.null, upper=fullmodel),
direction = "forward", k=log(n)+2*gamm*log(ncol(x)),trace=FALSE )
#model=names(model$coefficients)[-c(1)]
return(model)
}


model_selection = function(x,y,option,gamm)
{
n = nrow(x)
idx2 = seq(as.integer(n/log(n)),n,by = (as.integer(n/log(n))))
```

```
if(idx2[length(idx2)]!=n)idx2 = c(idx2,n)
idx1 = c(1,(idx2+1))
idx1=idx1[-length(idx1)]
xmat = list()
idx=NULL
scope1 = y~.
for(i in 1:length(idx2))
{
if(length(idx)>0)
{
if(option=="FWER")
{
tmp.model = SignifReg(scope1,data=data.frame(y,x[,c(idx,idx1[i]:idx2[i])]),
alpha=0.05,direction="forward",
criterion="p-value",correction="Bonf")
}
if(option=="backward")
{
tmp.model = backward_elimination(x[,c(idx,idx1[i]:idx2[i])],y,gamm)
}
if(option=="forward")
{
tmp.model = forward_selection(x[,c(idx,idx1[i]:idx2[i])],y,gamm)
}
}
if(length(idx)==0)
{
if(option=="FWER")
```

```
{
tmp.model = SignifReg(scope=scope1,data=data.frame(y,x[,c(idx1[i]:idx2[i])]),
alpha=0.05,direction="backward",
criterion="p-value",correction="Bonf")
}
if(option=="backward")
{
tmp.model = backward_elimination(x[,(idx1[i]:idx2[i])],y,gamm)
}
if(option=="forward")
{
tmp.model = forward_selection(x[,c(idx1[i]:idx2[i])],y,gamm)
}
}
if(dim(coef(summary(tmp.model)))[1]>1) # if there is any siginficant variable
{
xmat = as.character(names(tmp.model$coefficients)[-c(1)])
idx = match(xmat,colnames(x))
}
if(length(idx)==(n-1)) break;
}
# return final slected variables names
return(colnames(x)[idx])  ################################################  Changed
}


fwd.bwd.bonf.SIS.method=function(bnf,bd,fd,x,y) #x is matrix
{
```

```
#set of backward
idx = match(bd,colnames(x))
bwd.newX = as.matrix(x[,idx])
#set of forward
idx = match(fd,colnames(x))
fwd.newX = as.matrix(x[,idx])

#set of bonferroni
idx = match(bnf,colnames(x))
bnf.newX = as.matrix(x[,idx])
n = nrow(x)
# SIS with forward
fwd.lasso.fit =  SIS(fwd.newX,y,family="gaussian",nsis = as.integer(n/log(n)),iter=FALSE
fwd.scad.fit  =  SIS(fwd.newX,y,family="gaussian",nsis = as.integer(n/log(n)),iter=FALSE
fwd.mcp.fit   =  SIS(fwd.newX,y,family="gaussian",nsis = as.integer(n/log(n)),iter=FALSE
#The vector of indices selected by (I)SIS.
fwd.lasso.names = fd[fwd.lasso.fit$ix]
fwd.scad.names = fd[fwd.scad.fit$ix]
fwd.mcp.names =  fd[fwd.mcp.fit$ix]

# SIS with backward
bwd.lasso.fit =  SIS(bwd.newX,y,family="gaussian",nsis = as.integer(n/log(n)),iter=FALSE
bwd.scad.fit  =  SIS(bwd.newX,y,family="gaussian",nsis = as.integer(n/log(n)),iter=FALSE
bwd.mcp.fit   =  SIS(bwd.newX,y,family="gaussian",nsis = as.integer(n/log(n)),iter=FALSE
#The vector of indices selected by (I)SIS.
bwd.lasso.names = bd[bwd.lasso.fit$ix]
bwd.scad.names = bd[bwd.scad.fit$ix]
bwd.mcp.names =  bd[bwd.mcp.fit$ix]
```

```
# SIS with p.reg
bnf.lasso.fit =  SIS(bnf.newX,y,family="gaussian",nsis = as.integer(n/log(n)),iter=FALSE
bnf.scad.fit  =  SIS(bnf.newX,y,family="gaussian",nsis = as.integer(n/log(n)),iter=FALSE
bnf.mcp.fit   =  SIS(bnf.newX,y,family="gaussian",nsis = as.integer(n/log(n)),iter=FALSE
#The vector of indices selected by (I)SIS.
bnf.lasso.names = bnf[bnf.lasso.fit$ix]
bnf.scad.names = bnf[bnf.scad.fit$ix]
bnf.mcp.names =  bnf[bnf.mcp.fit$ix]

# return variable names for fwd lasso, scad,mcp
#                           bwd
#                           bonf
result = list(fwd.lasso = fwd.lasso.names,fwd.scad = fwd.scad.names,fwd.mcp=fwd.mcp.name
bwd.lasso = bwd.lasso.names,bwd.scad = bwd.scad.names,bwd.mcp=bwd.mcp.names,
bnf.lasso = bnf.lasso.names,bnf.scad = bnf.scad.names,bnf.mcp=fwd.mcp.names)
return(result)
}


performance.sim = function(true.v,select.v,total.p)
{
# Coverage Probability: (1/num.simulation)*sum(I(S(k) includes all true variables))
# True Negative, False Negative
# True Positive, False Positive
# FDR and its confidence error)
# Correctly Fitted: (1/num.simulation)*sum(I(S(k) == True Model))
# Average Model Size
```

```
# visualize EBIC : boxplots
#idx =  match(true.v,select.v) # return 1 if true, 0 otherwise
CP =  length(intersect(true.v,select.v))/length(true.v) #as.numeric(sum(idx>0,na.rm=T) =
TP = length(intersect(true.v,select.v))
FP = length(setdiff(select.v,true.v))#False positive
TN =(total.p-length(true.v)-length(setdiff(select.v,true.v)))
FN = length(setdiff(true.v,select.v))
TPR = TP/(TP+FN)
FPR=FP/(FP+TN)
TNR=TN/(FP+TN)
FNR=1-TPR
RP = length(select.v)  # model size
FDR = FP/length(select.v) #False discovery rate
result = list(Coverage.Prob=CP,
TPR = TPR,
FPR=FPR,
TNR=TNR,
FNR=FNR,
Model.Size = RP,
FDR = FDR)
return(result)
}

add.result = function(a1,tmp,itr)
{
# make list in the first simulation
if(itr==1){a1=list()}
# store each value into each category
```

```
a1$Coverage.Prob = c(a1$Coverage.Prob,tmp$Coverage.Prob)
a1$TPR = c(a1$TPR,tmp$TPR)
a1$FPR = c(a1$FPR,tmp$FPR)
a1$TNR = c(a1$TNR,tmp$TNR)
a1$FNR = c(a1$FNR,tmp$FNR)
a1$Model.Size = c(a1$Model.Size,tmp$Model.Size)
a1$FDR = c(a1$FDR,tmp$FDR)
return(a1)
}

############################
#Examples
library(ncvreg)
library(SIS)
library(glmnet)
library(mvtnorm)

options=c("pred1","pred2","pred3","pred4","pred5","pred6")
gamm = 1
Beg=Sys.time()
result = NULL
for (k in options){
if (k=="pred1"){true.v=paste("X",1:8,sep="");ntv=8;rho1=0;}
if (k=="pred2"){true.v = paste("X",c(1,4,7),sep="");ntv=3;rho1=R2;}
if (k=="pred3"){true.v = paste("X",1:3,sep="");ntv=3;rho1=R2;}
if (k=="pred4"){true.v = paste("X",1:5,sep="");ntv=5;rho1=R2;}
if (k=="pred5"){true.v = paste("X",1:8,sep="");ntv=8;rho1=R2;}
#if (k=="pred6"){true.v = paste("X",1:as.integer(sqrt(n)),sep="");ntv=as.integer(sqrt(n)
```

```
## Initialize all variables for results with empty set
bd1=fd1=bnf1=NULL
SIS.fwd.lasso=SIS.fwd.scad=SIS.fwd.mcp=NULL
SIS.bwd.lasso =SIS.bwd.scad =SIS.bwd.mcp =NULL
SIS.bnf.lasso = SIS.bnf.scad = SIS.bnf.mcp = NULL
SIS.lasso1=SIS.scad1=SIS.mcp1 = NULL
ISIS.lasso1=ISIS.scad1=ISIS.mcp1 = NULL

for(j in 1:rept)
{
set.seed(j)
data_final=data_generation(options=k,R2)

x = data_final[[1]]

y = data_final[[2]]
true.v = data_final[[3]]

#ranking variables with correlation from largest to smallest
p.value = apply(x,2,function(x)cor.test(x,y)$p.value)
order.idx = order(p.value,decreasing = F)
# rank with respect to p values
x = x[,order.idx]

#backward only with (n-2) variables
bd = model_selection(x,y,option="backward",gamm)
```

```
#forward only with (n-2) variables
fd = model_selection(x,y,option="forward",gamm)

#p value based (Bonf) model selection
bnf = model_selection(x,y,option="FWER",gamm)

############################################################
# LASSO and SCAD based on SIS (choose n/log(n))
fit.SIS.shrinkage = fwd.bwd.bonf.SIS.method(bnf,bd,fd,x,y)

# shrinkage methods with forward
fwd.lasso.names = fit.SIS.shrinkage$fwd.lasso
fwd.scad.names =  fit.SIS.shrinkage$fwd.scad
fwd.mcp.names =  fit.SIS.shrinkage$fwd.mcp
# shrinkage methods with backward
bwd.lasso.names = fit.SIS.shrinkage$bwd.lasso
bwd.scad.names =  fit.SIS.shrinkage$bwd.scad
bwd.mcp.names =  fit.SIS.shrinkage$bwd.mcp
# shrinkage methods with backward
bnf.lasso.names = fit.SIS.shrinkage$bnf.lasso
bnf.scad.names =  fit.SIS.shrinkage$bnf.scad
bnf.mcp.names =  fit.SIS.shrinkage$bnf.mcp

############################################################
#ISIS - lasso and scad: names(lasso$coef.est)[-1] (selected variables)
n = nrow(x)

fit.ISIS.scad = SIS(as.matrix(x),y,family="gaussian",penalty="SCAD",tune="ebic",
```

```
nfolds=10,type.measure="deviance",gamma.ebic=1,nsis = as.integer(n/log(n)),
iter.max=as.integer(log(n)-1),standardize=TRUE)
fit.ISIS.mcp = SIS(as.matrix(x),y,family="gaussian",penalty="MCP",tune="ebic",
nfolds=10,type.measure="deviance",gamma.ebic=1,nsis = as.integer(n/log(n)),
iter.max=as.integer(log(n)-1),standardize=TRUE)
fit.ISIS.lasso = SIS(as.matrix(x),y,family="gaussian",penalty="lasso",tune="ebic",
nfolds=10,type.measure="deviance",gamma.ebic=1 ,nsis = as.integer(n/log(n)),
iter.max=as.integer(log(n)-1),standardize=TRUE)

# dat = read.csv("result_1_.csv"dat = read.csv("result_1_.csv"dat =
#read.csv("result_1_.csv"dat = read.csv("result_1_.csv"find selected variables
ISIS.lasso.names =  names(fit.ISIS.lasso$coef.est)[-1]
ISIS.scad.names = names(fit.ISIS.scad$coef.est)[-1]
ISIS.mcp.names = names(fit.ISIS.mcp$coef.est)[-1]


# SIS - lasso and scad :
#iter=FALSE -> one time of SIS with n/log(n)

fit.SIS.scad = SIS(as.matrix(x),y,family="gaussian",penalty="SCAD",tune="ebic",
nfolds=10,type.measure="deviance",gamma.ebic=1,nsis=as.integer(n/log(n)),
iter=FALSE,standardize=TRUE)
fit.SIS.mcp = SIS(as.matrix(x),y,family="gaussian",penalty="MCP",tune="ebic",
nfolds=10,type.measure="deviance",gamma.ebic=1,nsis=as.integer(n/log(n)),
iter=FALSE,standardize=TRUE)
fit.SIS.lasso = SIS(as.matrix(x),y,family="gaussian",penalty="lasso",tune="ebic",
nfolds=10,type.measure="deviance",gamma.ebic=1 ,nsis=as.integer(n/log(n)),
iter=FALSE,standardize=TRUE)
```

```
SIS.lasso.names = names(fit.ISIS.lasso$coef.est)[-1]
SIS.scad.names = names(fit.ISIS.scad$coef.est)[-1]
SIS.mcp.names = names(fit.ISIS.mcp$coef.est)[-1]
################### result ###################

# Coverage Probability: (1/num.simulation)*sum(I(S(k) includes all true variables))
# True Negative, False Negative
# True Positive, False Positive
# FDR and its confidence error)
# Correctly Fitted: (1/num.simulation)*sum(I(S(k) == True Model))
# Average Model Size
# visualize EBIC : boxplots
p = ncol(x)
# variable selection methods only
tmp.bd1 = performance.sim(true.v,bd,p)
tmp.fd1 = performance.sim(true.v,fd,p)
tmp.bnf1 = performance.sim(true.v,bnf,p)

# SIS with variable slection methods
tmp.SIS.fwd.lasso = performance.sim(true.v,fwd.lasso.names,p)
tmp.SIS.fwd.scad = performance.sim(true.v,fwd.scad.names,p)
tmp.SIS.fwd.mcp = performance.sim(true.v,fwd.mcp.names,p)

tmp.SIS.bwd.lasso = performance.sim(true.v,bwd.lasso.names,p)
tmp.SIS.bwd.scad = performance.sim(true.v,bwd.scad.names,p)
tmp.SIS.bwd.mcp = performance.sim(true.v,bwd.mcp.names,p)
```

```
tmp.SIS.bnf.lasso = performance.sim(true.v,bnf.lasso.names,p)
tmp.SIS.bnf.scad = performance.sim(true.v,bnf.scad.names,p)
tmp.SIS.bnf.mcp = performance.sim(true.v,bnf.mcp.names,p)
##################
tmp.SIS.lasso1 = performance.sim(true.v,SIS.lasso.names,p)
tmp.SIS.scad1  = performance.sim(true.v,SIS.scad.names,p)
tmp.SIS.mcp1 = performance.sim(true.v,SIS.mcp.names,p)

tmp.ISIS.lasso1 = performance.sim(true.v,ISIS.lasso.names,p)
tmp.ISIS.scad1  = performance.sim(true.v,ISIS.scad.names,p)
tmp.ISIS.mcp1 = performance.sim(true.v,ISIS.mcp.names,p)

# store each value into lists
bd1 = add.result(bd1,tmp.bd1,j)
fd1 = add.result(fd1,tmp.fd1,j)
bnf1 = add.result(bnf1,tmp.bnf1,j)
#############################################################
SIS.fwd.lasso = add.result(SIS.fwd.lasso,tmp.SIS.fwd.lasso,j)
SIS.fwd.scad = add.result(SIS.fwd.scad,tmp.SIS.fwd.scad,j)
SIS.fwd.mcp = add.result(SIS.fwd.mcp,tmp.SIS.fwd.mcp,j)
#############################################################
SIS.bwd.lasso = add.result(SIS.bwd.lasso,tmp.SIS.bwd.lasso,j)
SIS.bwd.scad = add.result(SIS.bwd.scad,tmp.SIS.bwd.scad,j)
SIS.bwd.mcp = add.result(SIS.bwd.mcp,tmp.SIS.bwd.mcp,j)
#############################################################
SIS.bnf.lasso = add.result(SIS.bnf.lasso,tmp.SIS.bnf.lasso,j)
SIS.bnf.scad = add.result(SIS.bnf.scad,tmp.SIS.bnf.scad,j)
SIS.bnf.mcp = add.result(SIS.bnf.mcp,tmp.SIS.bnf.mcp,j)
```

```
#############################################################
SIS.lasso1 = add.result(SIS.lasso1,tmp.SIS.lasso1,j)
SIS.scad1 = add.result(SIS.scad1,tmp.SIS.scad1,j)
SIS.mcp1 = add.result(SIS.mcp1,tmp.SIS.mcp1,j)
######################################## ######################
ISIS.lasso1 = add.result(ISIS.lasso1,tmp.ISIS.lasso1,j)
ISIS.scad1 = add.result(ISIS.scad1,tmp.ISIS.scad1,j)
ISIS.mcp1 = add.result(ISIS.mcp1,tmp.ISIS.mcp1,j)
}
result = c(result,list(bd1,fd1,bnf1,
SIS.fwd.lasso,
SIS.fwd.scad,
SIS.fwd.mcp,
SIS.bwd.lasso,
SIS.bwd.scad,
SIS.bwd.mcp,
SIS.bnf.lasso,
SIS.bnf.scad,
SIS.bnf.mcp,
SIS.lasso1,
SIS.scad1,
SIS.mcp1,
ISIS.lasso1,
ISIS.scad1,
ISIS.mcp1))
}

Sys.time() - Beg
```

```
# result dimension is 60 by ???
file.name = paste("result",indx,".csv",sep="_")
write.csv(result,file.name,row.names=F)
```

Curriculum Vitae

Foli Sophia Korkor was born in a suburb of Ghana and the third child among five children. She graduated from OLA Girls Senior High School in 2011 and entered Kwame Nkrumah University of Science and Technology (KNUST) in the same year where she pursued a Bachelors Degree in Mathematics, graduating with a Second Class Honors in 2015. Upon graduation, she worked for a year at KNUST as a Teaching Assistant for the Department of Mathematics.

In Fall 2016, she joined the University of Texas at El Paso (UTEP) to pursue a Masters degree in Statistics. While in UTEP, she worked as a Teaching Assistant till July 2017 and then as a Research Assistant till May, 2018 with Dr. Sangjin Kim, conducting research on Backward Elimination Method for High Dimensional Variable Screening. She intends to work as a Lecturer as a career and pursue a PhD in Public Health or Biostatistics in any renowned university.

Email address: sfoli@miners.utep.edu
korkorfoli17@gmail.com