5-2015

High Dimensional Model Selection and Validation: A Comparison Study

Zhengyi Li
St. Cloud State University

High Dimensional Model Selection and Validation: A Comparison Study

by

Zhengyi Li

A Thesis

Submitted to the Graduate Faculty of

St. Cloud State University

in Partial Fulfilment of the Requirements

for the Degree

Master of Science in

Applied Statistics

May, 2015

Thesis Committee:
Hui Xu, Chairperson
David Robinson
Richard Sundheim

Abstract

Model selection is a challenging issue in high dimensional statistical analysis, and many approaches have been proposed in recent years. In this thesis, we compare the performance of three penalized logistic regression approaches (Ridge, Lasso, and Elastic Net) and three information criteria (AIC, BIC, and EBIC) on binary response variable in high dimensional situation through extensive simulation study. The models are built and selected on the training datasets, and their performance are evaluated through AUC on the validation datasets. We also display the comparison results on two real datasets (Arcene Data and University Retention Data). The performance differences among those approaches are discussed at the end.

Acknowledgments

I am grateful to my Thesis advisor, Hui Xu, for introducing this interesting topic to me. I cannot give enough thanks to his extraordinary guidance and patience. I am so thankful for his continuous support, motivation and immense knowledge. I would like to thank the rest of my thesis committee: Dr. David Robinson and Dr. Richard Sundheim, for their constructive and insightful comments and hard questions. I also thankful for their inspiration on helping me see the meaning of statistical work and motive me to work as hard as they do. Last but not least, I would like to thank my professors who are not my thesis committee, thank you for guidance through my study in Saint Cloud State University.

Table of Contents

Page

LIST OF TABLES 5

LIST OF FIGURES 6

Chapter

1. Introduction 7

2. Penalized Least Squares Regression Approaches 8

3. Model Selection Criteria 13

4. Penalized Logistic Regression 17

5. Simulation 19

6. Applications 30

Arcene Data 30

Retention Data 32

Exploratory Data Analysis 33

Model Results 36

7. Discussion 39

References 42

Appendices

A. Variable Definitions for Retention Data 44

B. R Code 45

List of Tables

Table Page

1. Correlation Matrix for Case 2 Simulation Data 21
2. Simulation Results for Case 1 and Case 2 22
3. Simulation Results for Case 3 and Case 4 25
4. Simulation Results for Case 5 and Case 6 26
5. Simulation Results for Case 7 and Case 8 26
6. Variables Selection for Lasso and Elastic Net by EBIC 28
7. Variables Selection for Lasso and Elastic Net by BIC 28
8. Model Results for Arcene Data in Original Scale 31
9. Model Results for Normalized Arcene Data 31
10. Distributions of Normal Variables of Retention Data 33
11. Distribution of Continuous Variables of Retention Data 35
12. Correlation Matrix for ACT scores 36
13. Model Results for Retention Data 37
14. Variables Selection for Retention Data 38

List of Figures

Figure — Page

1. Bias-variance trade-off 9
2. Variable trace plots for Ridge, Lasso and Elastic net for case 1 23
3. Variable trace plots for Ridge, Lasso and Elastic Net for case 2 24

Chapter 1: Introduction

Various regression methods have been used to build models to predict future results for decades, the ordinary least squares method is one of them that has been widely applied. The analysis procedure of this approach is mathematically easy and the results it produces are easily interpretable. However in some areas, like gene expression data analysis and medical studies, data with small number of observations and large number of variables is a typical situation. Least squares regression algorithm will fail to implement. Having too many variables in a model may cause the overfitting problem, and it may affect the accuracy of the prediction. So model selection becomes crucial. The traditional model selection methods such as subset selection, Akaike's information criterion, cross-validation, generalized cross validation and ordinary Bayesian information criterion tend to choose too many variables (Chen & Chen, 2008). Several penalized regression methods were invented to fix the drawbacks of the least squares method. We compare several variable selection approaches like Ridge, Lasso, and Elastic Net on binary response data, and I am also going to apply these methods to medical data obtained from The National Cancer Institute and the education data from St. Cloud State University.

Chapter 2: Penalized Least Squares Regression Approaches

In the most recent years, technologies have greatly change the traditional ways of collecting data, it is very common to collect a data with numerous variables, however the number of observations may be small due to the cost. Data sets with more variables than observations are known as high-dimensional. Classical statistical methods for regression and classification are developed for the data with less variables than observations, when the number of features is greater than the number of observations, the traditional classical methods, like ordinary least square method, tend to over fit the model.

By the Gauss-Markov theorem, the estimators of the linear regression coefficients produced by ordinary least square procedure are the best linear unbiased estimators which mean the estimators have the smallest variances among all the unbiased estimators. However when the collinearity between the explanatory variables presents or the number of predictors is much greater than the number of observations, some of the estimates ordinary least squares produce have high variance. Trying to reduce the variance in this situation, Horel and Kennard (1970) proposed ridge regression which can obtain more accurate prediction in the sense of mean squares error by introducing a little bias. Sometimes biased estimators may yield better prediction accuracy: Assume $\hat{\beta}$ is an unbiased estimator of β, which has mean 1 and variance 1. $\tilde{\beta}$ is a biased estimator of β, and $\tilde{\beta} = \frac{\hat{\beta}}{a}$, where a is a shrinkage factor and $a > 1$. We assume β is 1: Mean squared error: $E(\tilde{\beta} - 1)^2 = Var(\tilde{\beta}) + (E(\tilde{\beta}) - 1)^2 = \frac{1}{a^2} + (\frac{1}{a} - 1)^2$;

Bias: $Bias(\tilde{\beta}) = E(\tilde{\beta}) - \beta = E(\tilde{\beta}) - E(\hat{\beta}) = E\left(\frac{\hat{\beta}}{a}\right) - 1 = \frac{1}{a} - 1$

Variance: $VAR(\tilde{\beta}) = VAR\left(\frac{\hat{\beta}}{a}\right) = \frac{1}{a^2}$

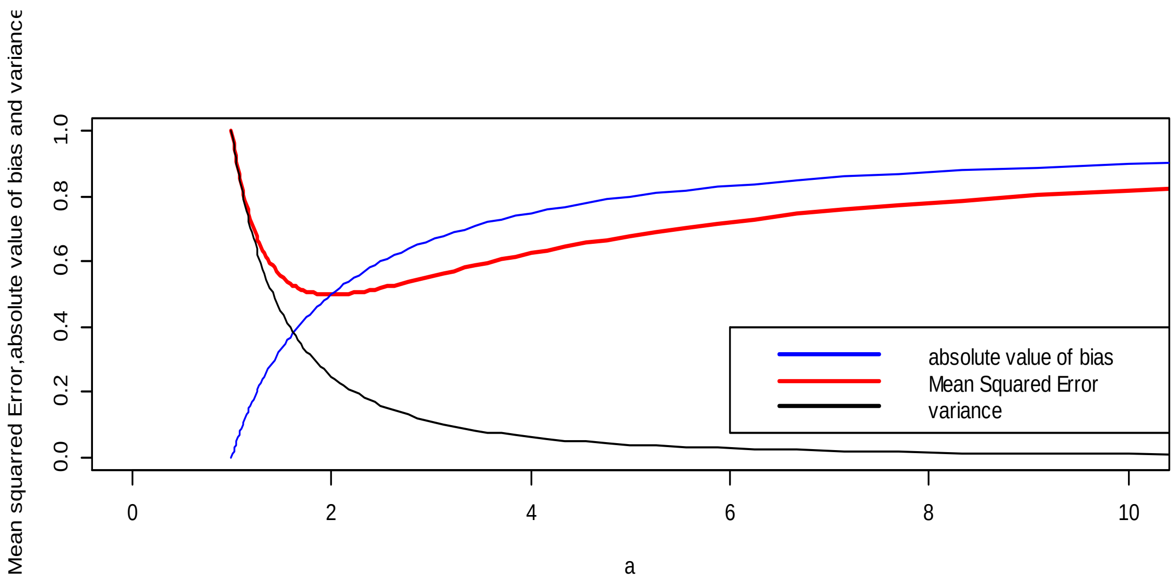

Figure 1. Bias-variance trade-off.

From Figure 1 above, we can see both of the mean squares error and variance reduced by introducing a little bias to the estimator. Consider the model:

$$\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\varepsilon}$$

Where $\boldsymbol{y}$ is the response vector; $\boldsymbol{X}$ is the design matrix; the unknown parameters denotes as $\boldsymbol{\beta}$, which represent a vector; $\boldsymbol{\varepsilon}$ is the error term which is distributed as N $(0,\sigma_{\varepsilon}^2\boldsymbol{I})$. For the ordinary least square procedure, we define the loss function as:

$$||\boldsymbol{y} - \boldsymbol{X\beta}||^2$$

Where $||.||^2$ denotes the squared Euclidean norm.

The solution which minimizes the loss function:

$$\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{y}$$

Penalty function is an additional term for the ordinary least squares, which is used to control the complexity of the model. The most commonly used penalty functions are L_1 and L_2 penalty:

$$L_1 = \sum_{j=1}^{p} |\beta_j|$$

$$L_2 = \sum_{j=1}^{p} \beta_j^2$$

The loss function for ridge regression can be written as the ordinary least squares regression loss function with L_2 penalty:

$$||\boldsymbol{y} - \boldsymbol{X\beta}||^2, subject\ to\ \sum_{j=1}^{p} \beta_{\boldsymbol{j}}^{\boldsymbol{2}} \leq t$$

It can also be written as a penalized loss functions:

$$||\boldsymbol{y} - \boldsymbol{X\beta}||^{\boldsymbol{2}} + \boldsymbol{\lambda_2\beta^T\beta}$$

The solution to minimize the loss function is by taking derivate with respect to. We obtain:

$$\widehat{\boldsymbol{\beta}} = (\boldsymbol{X^TX} + \boldsymbol{\lambda_2 I})^{\boldsymbol{-1}}\boldsymbol{X^Ty}$$

The inclusion of λ_2 makes $(\boldsymbol{X^TX} + \boldsymbol{\lambda_2 I})$ non-singular even if $\boldsymbol{X^TX}$ is not invertible. This was the original motivation for ridge regression. Since the solution depends on $\boldsymbol{\lambda_2}$ we need to find the "best" $\boldsymbol{\lambda_2}$. In Wahba and Golub's paper (1979), they showed that generalized cross-validation could be used to find the optimal $\boldsymbol{\lambda_2}$. Which minimizes the estimated prediction error.

In ridge regression, the coefficients are shrunk towards zero, but will never be zero. When we have a very high dimensional sparse data, the model for large sparse data is not easy to interpret. To overcome this difficulty, the lasso method (least absolute shrinkage and selection operator) was proposed by Robert Tibshirani (1996). The loss function for lasso:

$$\|\boldsymbol{y} - \boldsymbol{X\beta}\|^2, \mathit{subject\ to} \sum_{j=1}^{p} |\beta_j| \leq t$$

The regression coefficients are estimated as:

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\boldsymbol{argmin}}\, \boldsymbol{\beta}^{\boldsymbol{T}}(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X})\boldsymbol{\beta}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{y} - \boldsymbol{2}\boldsymbol{y}^{\boldsymbol{T}}\boldsymbol{X\beta} + \boldsymbol{\lambda_1}|\boldsymbol{\beta}|_{\boldsymbol{1}}$$

In the original paper of Lasso, Tibshirani (1996) described three methods to find the estimation of lasso parameter $\boldsymbol{\lambda_1}$: cross-validation, generalized cross-validation and an analytical unbiased estimate of risk. He suggested that in the practical problems, we might simply choose the most convenient method. Compared to the ridge regression, Lasso method gives us an interpretable model by shrinking some coefficients to exact zero. With a large number of independent variables, the lasso method can select a simpler model with the strongest effects.

Although the Lasso has been used widely and successfully in many situations, it still has some drawbacks: (a) in the $p \gg n$ case, Lasso algorithms are limited because it can only select at most n variables. (b) When we have a group of highly correlated explanatory variables, the Lasso tends to choose just one of them. It cannot reveal the group information

(c) for usual n>p case, the ridge regression perform better than Lasso regression when we have high correlation between independent variables (Zou & Hastie, 2005). Zou and Hastie proposed a new regularization technique named Elastic Net. This method is similar to Lasso and whenever ridge regression improves the Ordinary least squares, the elastic net will improve the lasso, and it also can select groups of variable with high correlation. Firstly, they introduced the naïve elastic net method: The loss function for elastic net:

$$\|\boldsymbol{y} - \boldsymbol{X\beta}\|^2, subject\ to\ \sum_{j=1}^{p} |\beta_j| \leq t_1\ and\ \sum_{j=1}^{p} \beta_j^2 \leq t_2$$

The elastic net penalty function is the combination of the lasso and ridge penalty functions. So it maintains the characteristics of both lasso and ridge regression. However, empirical evidence shows that the naïve elastic net does not perform satisfactorily unless it is very close to either ridge regression or the lasso (Zou & Hastie, 2005). In order to improve the accuracy of prediction of naïve elastic net, they developed the elastic net method by rescaling naïve elastic net coefficients, with a scaling transformation preserves the variable selection property of the naïve elastic net and empirically the elastic net performs very well when compared with lasso and ridge regression (Zou & Hastie , 2005).

Chapter 3: Model Selection Criteria

The Akaike information criterion is generally considered as the first model selection criterion, which was introduced by Hirotugu Akaike in his seminal paper (1973). The traditional maximum likelihood paradigm could only estimate the unknown parameters of a model with a specified structure. Akaike proposed a new paradigm that could simultaneously process model estimation and selection.

Some Notion used in this section:

True model: $g(y)$

Candidate models: $f(y|\beta_j)$

Fitted model: $f(y|\hat{\beta}_j)$

Candidate model space: F

The dimension of β_j: j

Akaike information criterion is basically a method to measure the difference between the fitted model and true model. The best model is the one has smallest difference. The measurement is the Kullback-Leibler information.

For our model selection purpose, we consider Kullback-Leibler information between true model g(y) and fitted model $f(y|\hat{\beta}_j)$

$$I(\beta_j) = E\left\{Log\,\frac{g(y)}{f(y|\beta_j)}\right\}$$

Where E denotes the expectation under g(y).

Kullback discrepancy is defined as $d(\beta_j) = E\{-2Logf(y|\beta_j)\}$.

We can write

$2I(\beta_j) = E\{-2Logf(y|\beta_j)\} - (-E\{-2Logg(y)\}) = d(\beta_j) - E\{-2Logg(y)\}$

Since the true model g(y) does not depend on β_j , we can use $d(\beta_j)$ to substitute $I(\beta_j)$, $d(\hat{\theta}_j) = E\{-2Logf(y|\beta_j)\}|_{\beta_j=\hat{\beta}_j}$ can also approximately reflect the difference between the true model and fitted model. But we cannot evaluate $d(\hat{\beta}_j)$ directly due to only the relative magnitude of AIC is useful in model selection. In Akaike's paper, he suggested that $-2Logf(y|\beta_j)$ could be used as a biased estimator of $d(\hat{\beta}_j)$. He also proved that the bias can be asymptotically estimated by twice the dimension of θ_j . Then we get $AIC = -2Logf(y|\beta_j) + 2 * p_j$, which is asymptotically unbiased estimator of $d(\hat{\beta}_j)$ in the situation that the simple size n is comparatively larger that the number of variables.

Bayesian information criterion is another widely used approach to determine the dimensionality of model. Superficially, the only difference between BIC and AIC is the second term, but the BIC can be derived as an estimate of the Bayes factor for two models (Ghosh, Delampady, & Samanta, 2006).

Suppose we have two models m_1 with density function $f(y|\beta_1)$ and m_0 with density function $|f(y|\beta_0)$. Let $g(\beta_i)$ be the prior density of β conditional on M_i, i=0, 1. The Bayes factor

$$\mathrm{B}_{01}(\mathrm{y}) = \frac{m_0(y)}{m_1(y)}$$

Where $m_i = \int f(y|\beta_i)g_i(\beta_i)d\beta_i, i = 0,1$

Using a second order Taylor series approximation, we expand m_i around the maximum likelihood estimate $\hat{\beta}_i$, here $H_{\beta_i}$is the observed fisher information matrix, if the observations are distributed identically and independently, we have that $H_{\widehat{\beta}_i}$=n$H_{1,\widehat{\beta}_i}$.

$$log\big(f(y|\beta_i)g_i(\beta_i)\big) \approx \log\Big(f\big(y\big|\hat{\beta}_i\ \big)g_i\big(\hat{\beta}_i\big)\Big) - \frac{1}{2}(\beta_i - \hat{\beta}_i)' H_{\widehat{\beta}_i}(\beta_i - \hat{\beta}_i)$$

Applying this to the Bayes factor:

$$m_i(y) \approx f\big(y\big|\hat{\beta}_i\big)g_i\big(\hat{\beta}_i\big)\int \exp(-\frac{1}{2}\big(\beta_i - \hat{\beta}_i\big)' H_{\theta_i}(\beta_i - \hat{\beta}_i))d\beta_i$$

$$= f\big(y\big|\hat{\beta}_i\big)g_i\big(\hat{\beta}_i\big)(2\pi)^{\frac{p_i}{2}}(n)^{-\frac{p_i}{2}}\left|H_{\widehat{\beta}_i}^{-1}\right|^{\frac{1}{2}}$$

Where p_i is the dimension of the parameter vector.

$$2\ln\big(\mathrm{B}_{01}(\mathrm{y})\big) = 2\log\frac{m_0(y)}{m_1(y)} = 2\ln\left(\frac{f(y|\widehat{\beta}_0)g_0(\widehat{\beta}_0)(2\pi)^{\frac{p_0}{2}}(n)^{-\frac{p_0}{2}}\left|H_{\widehat{\beta}_0}^{-1}\right|^{\frac{1}{2}}}{f(y|\widehat{\beta}_1)g_1(\widehat{\beta}_1)(2\pi)^{\frac{p_1}{2}}(n)^{-\frac{p_0}{2}}\left|H_{\widehat{\beta}_1}^{-1}\right|^{\frac{1}{2}}}\right) \approx 2\ln\left(\frac{f(y|\widehat{\beta}_0)}{f(y|\widehat{\beta}_1)}\right) +$$

$$ln\frac{g_0(\widehat{\beta}_0)}{g_1(\widehat{\beta}_1)} - (p_0 - p_1)\ln\left(\frac{n}{2\pi}\right) + ln\frac{\left|H_{\widehat{\beta}_0}^{-1}\right|}{\left|H_{\widehat{\beta}_1}^{-1}\right|}$$

Approximately

$$2\log\big(\mathrm{B}_{01}(\mathrm{y})\big) = 2\log\left(\frac{f(y|\widehat{\beta}_0)}{f(y|\widehat{\beta}_1)}\right) - (p_0 - p_1)\log\left(\frac{n}{2\pi}\right)$$

We usually compare fitted model with the null model then we get:

$$BIC = 2\log f\big(y\big|\hat{\beta}\big) + p_j * \log(n)$$

Bayesian information criteria with uniform prior distribution, which means we assume that all the candidate models have equal probability to be true model, tends to select too many variable in small-N-large-p situation which has been observed by Broman and Speed

(2002), Siegmund (2004) and so on. This inspired Chen and Chen (2008) to propose the extended Bayesian information criteria, which considered both the complexity of the candidate model and the complexity of the model space. In their paper, the new prior distribution used in the EBIC paradigm is given by this procedure: partition the model space S into $\mathrm{U}_{\mathrm{j}=1}^{\mathrm{p}}\mathrm{S}_{\mathrm{j}}$, and each subspace S_{j} includes all the models with j variables. Suppose $\tau(\mathrm{S}_{\mathrm{j}})$ is the size of S_{j}, and $\tau(\mathrm{S}_{\mathrm{j}}) = \binom{p}{j}$ where p is the number of variables in the whole model space. If we assign an equal probability to each variable in the subspace: $\mathrm{p}(\mathrm{s}|\mathrm{S}_{\mathrm{j}}) = 1/\tau(\mathrm{S}_{\mathrm{j}})$ which means every model in the model space has same probability to be chosen. Unlike in the ordinary BIC, we assign ($pr(\mathrm{S}_{\mathrm{j}})$ proportional to $\tau^{\xi}(\mathrm{S}_{\mathrm{j}})$ instead of $\tau(\mathrm{S}_{\mathrm{j}})$ for ξ between 0 and 1. We get the p(s) for variable in each subspace being the proportional to $\tau^{1-\gamma}(\mathrm{S}_{\mathrm{j}})$, where$\gamma = 1 - \xi$. Then this results the extended Bayesian information criteria:

$$BIC_{\gamma}(s) = -2\log L_n\{\hat{\beta}(s)\} + v(s)log(n) + 2\gamma log\tau(S_j), 0 \leq \gamma \leq 1$$

Where v(s) denotes the number of parameters in model s, $L_n\{\hat{\beta}(s)\}$ is the likelihood of model s. The choice of γ is important issue, one way proposed by Chen and Chen in the normal regression for choosing γ is to solve k from $p = n^k$, and $\gamma = 1 - \frac{1}{2k}$. In 2012, Chen and Chen proved that EBIC is consistent under generalized linear models. The simulation results in the paper support their conclusion.

Chapter 4: Penalized Logistic Regression

Ridge, Lasso and Elastic Net could also applied for the data with binary response. The regular logistic regression model has the form:

$$log\frac{\rho}{1-\rho} = \beta_0 + \boldsymbol{X^T\beta}$$

Where X is a vector of predictors. The coefficients are typically derived by maximizing the likelihood. Similar to the penalized linear regression, the coefficients are estimated by maximizing the log-likelihood subject to penalization on L1 or L2 (or the combination of L1 and L2) norm of the coefficients for penalized logistic regression. We can write the function in the following form:

$$L\left(\beta_{0,}\boldsymbol{\beta}, \lambda_1, \lambda_2\right) = -l(\beta_0, \boldsymbol{\beta}) + \lambda_1|\boldsymbol{\beta}|_1 + \lambda_2\boldsymbol{\beta^T\beta}$$

Where l indicates the binomial log-likelihood, λ_1 and λ_2 are the tuning parameter which control the amount of shrinkage, when $\lambda_1 = 0$, the penalized term is in the same manner as in ridge regression; when $\lambda_2 = 0$.

The coefficients are shrunk like these in lasso regression. When both of the tuning parameters are not zero, the combination of Lasso and Ridge penalties, which gives the Elastic net regression.

Similar to the way of constructing AIC, BIC and EBIC for ordinary least squares regression, the formulas for regularized logistic regression are shown below:

$$AIC_j = -2 * l(\beta_0, \boldsymbol{\beta}) + 2 * p_j$$

$$BIC_j = -2 * l(\beta_0, \boldsymbol{\beta}) + p_j * \log(n)$$

$$EBIC_j = -2 * l(\beta_0, \boldsymbol{\beta}) + p_j * \log(n) + 2 * \gamma * \log(\tau)$$

Where p_j the number of variable in model j, and n is the number of observations in the model; $\gamma = 0.25$, which is suggested by Chen and Chen; $\tau = \binom{p}{p_j}$ and p is the number of variables in the full model.

Chapter 5: Simulation

In this paper, simulation study is conducted to examine the performance of the regression methods and variable selection criteria described above in 8 different data settings:

1. 500 observations and 10 explanatory variables;
2. 500 observations and 10 explanatory variables with collinearity;
3. 500 observations and 100 explanatory variables;
4. 500 observations and 100 explanatory variables with collinearity;
5. 500 observations and 500 explanatory variables;
6. 500 observations and 500 explanatory variables with collinearity;
7. 500 observations and 1500 explanatory variables;
8. 500 observations and 1500 explanatory variables with collinearity;

The following logistic model is employed in all of these cases to generate the simulation data:

$$\log\left(\frac{\rho}{1-\rho}\right) = \beta_i + \boldsymbol{X^T\beta}$$

Where all the explanatory variables are generated from N (0, 1). Then we have $\rho = \frac{e^{\beta_0+\boldsymbol{X^T\beta}}}{1+e^{\beta_0+\boldsymbol{X^T\beta}}}$, if $\rho < 0.5$ then y=0, else y=1; All the coefficients are generated from random uniform (0, 1), however in order to examine the variable selection criteria, some variables are assigned bigger coefficients, which will be dominant: for case 1 and case 2, the coefficient of x1 is multiplied by 10 to make x1 significant important; for the rest of the simulation data, the dominant variables that pre-selected for the simulation data are: x1, x2, x3, x22, x23, and x24 with coefficients 7.18, 1.62, 5.56, 9.46, 9.06, and 5.43, which are derived by multiplying

10 to simulated coefficients. Our expectation is the best variable selection criteria are able to identify these variables. In the cases with collinearity, one group of variables are correlated with correlation coefficient 0.75; the other group of variables are correlated with correlation coefficient 0.95. When we run the regressions on the simulated data, 2/3 randomly selected data will be used to train the models, the rest of the data is the validation data which is used to test the models performance, Area under curve for the ROC of the validation data is the major evaluation of the model performance. For all the cases, Ridge, Lasso and Elastic net models will be fitted, and cross-validation, AIC, BIC and EBIC are used to select the best models, since regular logistic is applicable in the cases when P<<n, we will also fit logistic models for case 1 and case 2.

We use the R package “glmnet” to fit ridge, lasso and Elastic Net regression, this package contains many functions which can fit various kinds of model, the functions we will use here are cv.glmnet and glmnet, which have a factor alpha, when alpha = 0, the model is a ridge regression; When alpha = 0.5, Elastic Net model is built; when alpha = 1, a lasso model will be fitted. Lambda is the tuning parameter which determents the amount of shrinkage, we test 1001 different lambdas from 0 to 1, every time we add 0.001 to previous tuning parameter.

For Case 1 and Case 2, x1 is the variable which has a significantly bigger coefficient. All other coefficients are created from random uniform (0, 1) distribution; In case 2, the correlation matrix is shown below:

Table 1

Correlation Matrix for Case 2 Simulation Data

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
x1	1.0000	***0.9638***	***0.9355***	0.0023	***0.7506***	***0.7623***	-0.0939	-0.0270	0.0362	0.1073
x2	0.9638	1.0000	0.9701	0.0089	0.7167	0.7524	-0.0987	-0.0378	0.0402	0.1037
x3	0.9355	0.9701	1.0000	0.0174	0.7058	0.7331	-0.1178	-0.0241	0.0323	0.0929
x4	0.0023	0.0089	0.0174	1.0000	0.0161	0.0358	-0.0779	-0.0222	-0.0540	-0.0480
x5	0.7506	0.7167	0.7058	0.0161	1.0000	0.5622	-0.1161	-0.0190	0.0300	0.1176
x6	0.7623	0.7524	0.7331	0.0358	0.5622	1.0000	-0.0264	-0.0530	-0.0007	0.1069
x7	-0.0939	-0.0987	-0.1178	-0.0779	-0.1161	-0.0264	1.0000	0.0300	0.0431	0.0586
x8	-0.0270	-0.0378	-0.0241	-0.0222	-0.0190	-0.0530	0.0300	1.0000	-0.0020	0.0520
x9	0.0362	0.0402	0.0323	-0.0540	0.0300	-0.0007	0.0431	-0.0020	1.0000	-0.0119
x10	0.1073	0.1037	0.0929	-0.0480	0.1176	0.1069	0.0586	0.0520	-0.0119	1.0000

Table 2 below shows the simulation results for case 1 and case 2, it is not surprising that AIC, BIC and EBIC selected the model with all the variables for ridge regression. Ridge regression keeps all the variables in the model and only shrinks the coefficients towards to zero for the variables that are less important. When we apply AIC, BIC and EBIC to ridge regression, all the information criteria tend to select the model with least log likelihood, in other word, these criteria always select the full model, which also means these selection criteria are not applicable for Ridge regression.

Table 2

Simulation Results for Case 1 and Case 2

Case 1: p=10 and n=500 witout collinearity

	EBIC		
Model	Lambda	Number of variables	AUC
Logistic	N/A	3	0.8948
Ridge	0	10	0.9083
Lasso	0.076	2	0.8820
Elastic Net	0.032	6	0.9080

	BIC		
Model	Lambda	Number of variables	AUC
Logistic	N/A	4	0.9041
Ridge	0	10	0.9083
Lasso	0.018	6	0.9076
Elastic Net	0.032	6	0.9080

	AIC		
Model	Lambda	Number of variables	AUC
Logistic	N/A	6	0.9072
Ridge	0	10	0.9083
Lasso	0	10	0.9083
Elastic Net	0	10	0.9083

Case 2: p=10 and n=500 with collinearity

	EBIC		
Model	Lambda	Number of variables	AUC
Logistic	N/A	3	0.9329
Ridge	0	10	0.9275
Lasso	0.01	8	0.9249
Elastic Net	0.014	9	0.9235

	BIC		
Model	Lambda	Number of variables	AUC
Logistic	N/A	3	0.9329
Ridge	0	10	0.9275
Lasso	0.01	8	0.9249
Elastic Net	0.014	9	0.9235

	AIC		
Model	Lambda	Number of variables	AUC
Logistic	N/A	4	0.9246
Ridge	0	10	0.9275
Lasso	0	10	0.9275
Elastic Net	0	10	0.9275

For logistic regression, information criterion based on stepwise method was applied here. In both cases, x1 which is the dominant variable is in all the stepwise models chosen by the variable selection criteria. AIC tends to select the candidate models with more variables, compared to the models favored by BIC and EBIC. In case 2, the variables (x2 and x3) that are highly correlated with x1 are dropped by all the criteria, the models fitted by stepwise methods perform similarly in terms of AUC. AUC is an abbreviation of Area Under Curve commonly used to determine which of the models predicts the binary response accurately, and the curve is called Receiver Operating Characteristic curve which is a plot of the true positive rate against the false positive rate for different possible cutoff points. The value of AUC is usually less or equal to one, the model with AUC close to 1 is considered as a good model. To compare the variable selection feature for Ridge, Lasso and Elastic Net, we look the variables trace plots for these three regression approaches on the case 1 data first.

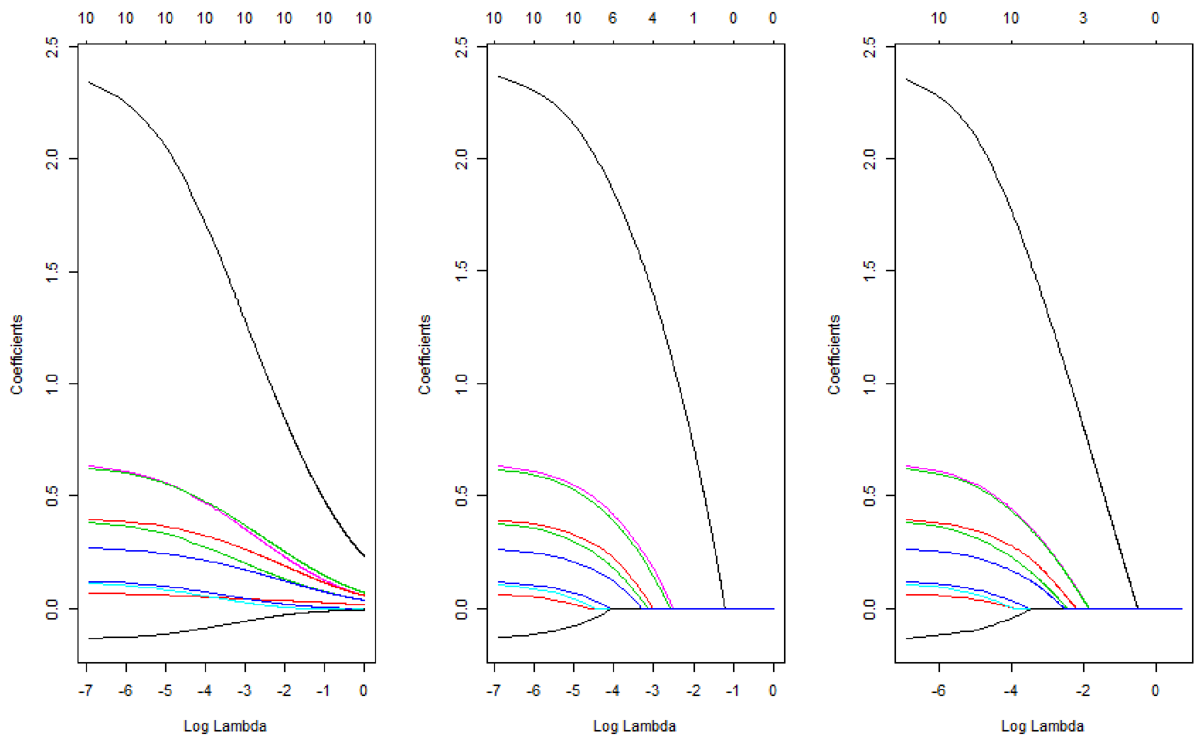

Figure 2. Variable trace plots for Ridge, Lasso and Elastic net for case 1. The left one is for ridge; middle one is the trace plot for lasso; the one on right is Elastic net.

In the plot above, each colored line represents the values of different coefficient, lambda is the tuning parameter tested in the selection procedure. As lambda increases, the coefficients are pulled towards to zero, with less important parameters being pulled to zero earlier. The coefficients of ridge regression could never be zero, when the lambda is big enough, the lasso and elastic net will assign zeros to variables which contribute to model very little. The Elastic net trace plot looks almost identical with the lasso plot, when there is no collinearity present in the data, Lasso and Elastic net behave similarly. Now, let us look the trace plots for case 2, where the collinearity problem exists.

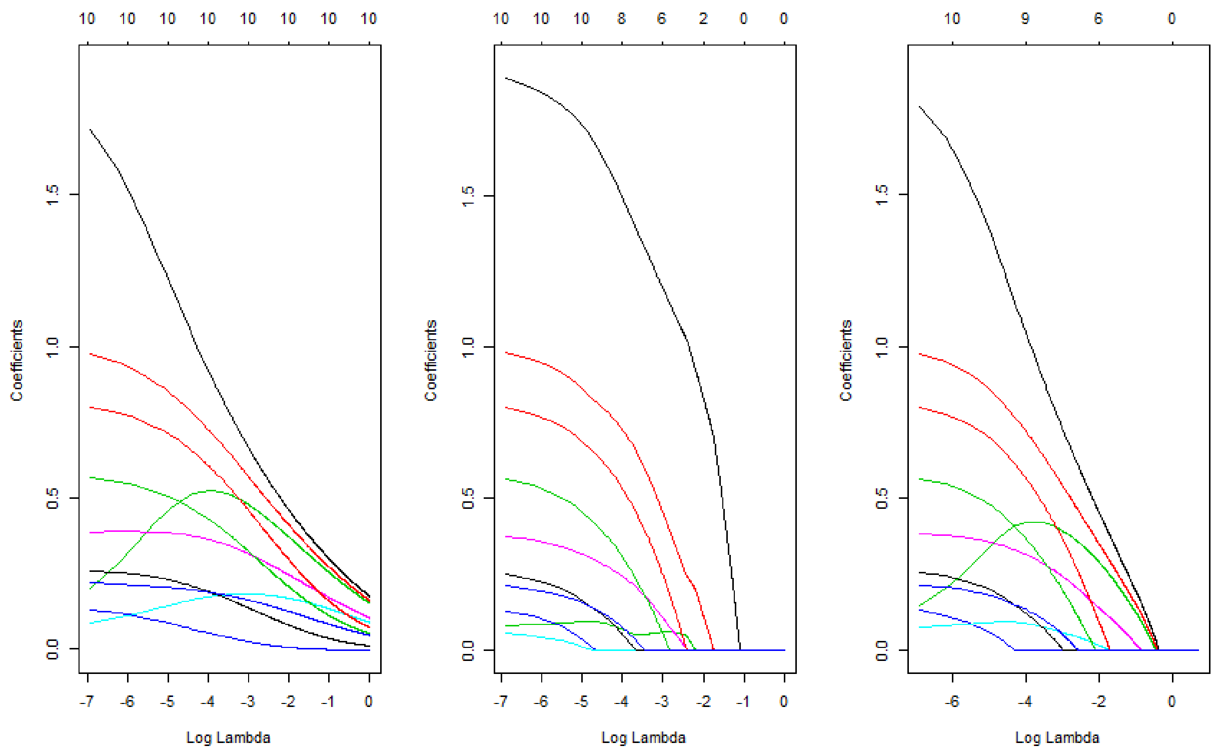

Figure 3. Variable trace plots for Ridge, Lasso and Elastic net for case 2. The left one is for ridge; middle one is the trace plot for lasso; the one on right is Elastic net.

Ridge regression solves the collinearity problem by shrinking the coefficients towards to each other. For Lasso, it randomly chooses one from a set of strong but correlated variables, however Elastic net has a compromise solution by keeping all the correlated variables in the model and assigns similar coefficients to these variables. If we compare the model performance on the prediction accuracy by looking the AUC for the validation data, no regression method outperform others substantially for the p<<n cases regardless of the existence of collinearity. AIC, BIC and EBIC are the model fit assessing method with the

penalty terms on the number of parameters that are selected for the models, similar to the models fitted by regular logistic regression. AIC has a favor to model with more variables, which adequately describe the unknown of the data. BIC has stricter penalty terms on the number of parameters, which tend to select models that are simpler and easier to interpret. EBIC adds one more penalty term to BIC, which considers the complexity of entire model space, however with this new penalty term, EBIC has a favor to models with even fewer variables by scarifying some prediction accuracy. From the simulation results above, we see that the models chosen by EBIC contains the fewest variables for each regression method for case 1. In case 2, BIC and EBIC agree on the models selections, however these models are the simplest models. The pairwise correlated variables groups for the simulation data with collinearity: Correlation=0.95: x1, x4, x5, x6, x7, x8; Correlation=0.75: x3, x9, x10, x11, x12, x13.

Table 3

Simulation Results for Case 3 and Case 4

p=100 and n=500 witout collinearity

	EBIC		
Model	Lambda	Number of variables	AUC
Ridge	0.000	100	0.9690
Lasso	0.047	6	0.9690
Elastic Net	0.111	6	0.9651

	BIC		
Model	Lambda	Number of variables	AUC
Ridge	0.000	100	0.9406
Lasso	0.047	6	0.9690
Elastic Net	0.111	6	0.9651

	AIC		
Model	Lambda	Number of variables	AUC
Ridge	0.000	100	0.9406
Lasso	0.001	72	0.9614
Elastic Net	0.001	84	0.9466

Case 3

p=100 and n=500 with collinearity

	EBIC		
Model	Lambda	Number of variables	AUC
Ridge	0.000	100	0.9608
Lasso	0.056	8	0.9752
Elastic Net	0.075	16	0.9904

	BIC		
Model	Lambda	Number of variables	AUC
Ridge	0.000	100	0.9608
Lasso	0.026	19	0.9926
Elastic Net	0.075	16	0.9904

	AIC		
Model	Lambda	Number of variables	AUC
Ridge	0.000	100	0.9608
Lasso	0.002	60	0.9887
Elastic Net	0.002	71	0.9901

Case 4

Table 4

Simulation Results for Case 5 and Case 6

Case 5

	EBIC		
Model	Lambda	Number of variables	AUC
Ridge	0.000	500	0.8131
Lasso	0.076	4	0.8344
Elastic Net	0.151	4	0.8601

	BIC		
Model	Lambda	Number of variables	AUC
Ridge	0.000	500	0.8131
Lasso	0.076	4	0.8344
Elastic Net	0.151	4	0.8601

	AIC		
Model	Lambda	Number of variables	AUC
Ridge	0.000	500	0.8131
Lasso	0.011	130	0.8345
Elastic Net	0.078	44	0.8630

Case 6

	EBIC		
Model	Lambda	Number of variables	AUC
Ridge	0.000	500	0.8812
Lasso	0.060	7	0.8756
Elastic Net	0.121	9	0.8691

	BIC		
Model	Lambda	Number of variables	AUC
Ridge	0.000	500	0.8812
Lasso	0.600	7	0.8756
Elastic Net	0.107	11	0.8710

	AIC		
Model	Lambda	Number of variables	AUC
Ridge	0.0000	500	0.8812
Lasso	0.0380	22	0.8851
Elastic Net	0.0810	24	0.8768

Table 5

Simulation Results for Case 7 and Case 8

Case 7

	EBIC		
Model	Lambda	Number of variables	AUC
Ridge	0.000	1500	0.7623
Lasso	0.112	2	0.6779
Elastic Net	0.224	2	0.6797

	BIC		
Model	Lambda	Number of variables	AUC
Ridge	0.000	1500	0.7623
Lasso	0.090	3	0.7309
Elastic Net	0.180	3	0.7302

	AIC		
Model	Lambda	Number of variables	AUC
Ridge	0.0000	1500	0.7623
Lasso	0.0630	16	0.7415
Elastic Net	0.1300	15	0.7408

Case 8

	EBIC		
Model	Lambda	Number of variables	AUC
Ridge	0.000	1500	0.7697
Lasso	0.095	3	0.8401
Elastic Net	0.195	6	0.8361

	BIC		
Model	Lambda	Number of variables	AUC
Ridge	0.000	1500	0.7697
Lasso	0.070	8	0.8429
Elastic Net	0.195	6	0.8361

	AIC		
Model	Lambda	Number of variables	AUC
Ridge	0.000	1500	0.7697
Lasso	0.054	21	0.8504
Elastic Net	0.111	24	0.8474

The ridge regression always keeps all the variables in the model, which results the penalty terms in AIC, BIC and EBIC has no effect on it. So we mainly compare Lasso and

Elastic Net, which have automated variable selection feature. Similar to what we see for case 1 and case 2, these two regression methods would not outperform to each other substantially for cases 3, 4, 5 and 6 data by comparing the Area under Curve for validation data; in case 3, case 5 and case 7, it turns out that EBIC and BIC agree on the models selection for Lasso and Elastic net regression, Lasso and Elastic net also contains same variables for each information criteria; in case 3 the models contains 5 out of 6 the pre-selected powerful variables: x1, x3, x22, x23, x24; in case 5, the models EBIC chooses the models with 4 variables: x1, x22, x23 and x24; in case 7, the models chosen by EBIC correctly select 2 variables: x22 and x23, BIC selects the model with 3 variables: x1, x22 and x23; For the data without presence of collinearity, EBIC and BIC could effectively identify the dominant variables and both information criteria tend to choose the same model in most cases, the models selected by AIC have more variables as we expected, however with more variables in the model, it does not improve the performance dramatically.

When we look case 4, case 6 and case 8, the simulation data with the existence of collinearity, the number of variables selected by lasso is quite different with Elastic net. For table 6 and 7, regardless of the information criteria, only one of the pairwise correlated variables for each correlation group is chosen for lasso model, however Lasso does not always choose the strongest variable among all the correlated one, it tends to choose one of them randomly and ignore others. Unlike lasso regression, Elastic net has the feature to select group effects, the simulation results in table 6 confirm this: in case 4, the models selected by EBIC and BIC not only contain all the pre-selected important variables but also the variables correlated; in case 6 and case 8, model fitted by Elastic net also contains some of the correlated variables. Now let us compare if EBIC has any advantages over BIC on identifying important variables, similar to what we see for the simulation data without collinearity, EBIC and BIC favor to the same models for most of the cases. For the cases EBIC and BIC do not agree to each other, EBIC tends to simpler model.

Table 6

Variables Selection for Lasso and Elastic Net by EBIC (Underline represents correlation 0.75 group; Bold represents correlation 0.95 group.)

Variables of Lasso models selected by EBIC	
Case 4	**x4**,x10,x22,x23,x24,x33,x35,x60
Case 6	x3,**x4**,x22,x23,x24,x68,x486
Case 8	**x3,x4**,x22

Variables of Elastic net models selected by EBIC	
Case 4	**x1**, x3,**x4,x5,x6,x7,x8,**x9,x10,x11,x12,x13,x22,x23,x24,x96
Case 6	**x1**,x2,x3,**x4,x5,**x22,x23,x24,x68
Case 8	**x4,x5,**x22,x23,x98,x242

Table 7

Variables Selection for Lasso and Elastic Net by BIC. (Underline represents correlation 0.75 group; Bold represents correlation 0.95 group.)

Variables of Lasso models selected by BIC	
Case 4	**x1**,x3,x20,x22,x23,x24,x33,x35,x60,x68,x79,x82,x96,x98,x242,x486
Case 6	x3,**x4**,x22,x23,x24,x68,x486
Case 8	x3,**x5**,x22,x23,x98,x242,x1377,x1421

Variables of Elastic net models selected by BIC	
Case 4	**x1**,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12,x13,x22,x23,x24,x96
Case 6	**x1**,x2,x3,**x4,x5,**x22,x23,x24,x46,x68,x486
Case 8	**x4,x5,**x22,x23,x98,x242

From the simulation results above, Ridge, Lasso and Elastic net do not outperform each other on prediction accuracy substantially. Ridge regression does not have the automated variable selection feature, it always keeps all the variables in the model with shrinking the coefficients of the less important variables towards to zero. Lasso and Elastic net behave similarly for the data without highly correlated variables, however, when collinearity problem presents in the data, Lasso tends to randomly choose one of the correlated variables, conversely, Elastic net keeps correlated variables in the model. EBIC is a stricter variable selection criteria, which has favor to parsimony model, compare to BIC and AIC. However, EBIC does not perform significantly better than BIC in terms of variable selection, in most cases EBIC and BIC pick the same models.

Chapter 6: Applications

Arcene Data

The data were obtained from The National Cancer Institute, which consist of mass-spectra obtained with the SELDI technique containing the biological information for each patient. The sample has 200 observations including patients with cancer and healthy patients. The purpose of this data is to distinguish cancer versus normal patterns from these massive spectrometric data, to be precise, 10000 features which are all integers ranged from 0 to 700. The task here is to see which approach among Ridge, Lasso and Elastic Net chosen by AIC, BIC and EBIC could successfully identify important variables. Before we fit models on the data, some of the variables which have a lot missing values are removed, we have 9939 variables left. The whole sample is split into train (67% of the data) and validation (33% of the data). In order to test the effect of the normalization of the data, we will build two sets of models: the first set are based on the data with original scale; the other set of models are built on the normalized data.

Table 8

Model Results for Arcene Data in Original Scale

	EBIC			Variables
Model	Lambda	Number of variables	AUC	
Ridge	0.000	9939	0.6929	x1-x9939
Lasso	0.248	2	0.6076	**x3783,x6594**
Elastic Net	0.514	2	0.6151	**x3783,x6594**
	BIC			
Model	Lambda	Number of variables	AUC	
Ridge	0.000	9939	0.6929	x1-x9939
Lasso	0.141	5	0.6449	x1936,**x3783**,x5982,x6594,x9818
Elastic Net	0.514	2	0.6151	**x3783,x6594**
	AIC			
Model	Lambda	Number of variables	AUC	
Ridge	0.000	9939	0.6929	x1-x9939
Lasso	0.102	12	0.7130	x815,x306,x754,x766,x1748,x1936,**x3783,**x4684,**x6594,**x7544,x7891,x9818
Elastic Net	0.514	2	0.6151	**x3783,x6594**

Table 9

Model Results for Normalized Arcene Data

	EBIC			Variables
Model	Lambda	Number of variables	AUC	
Ridge	0.000	9939	0.6929	x1-x9939
Lasso	0.127	5	0.6392	x1936,**x3783,**x5982,x6594,x9818
Elastic Net	0.477	2	0.6285	**x3783,x6594**
	BIC			
Model	Lambda	Number of variables	AUC	
Ridge	0.000	9939	0.6929	x1-x9939
Lasso	0.127	5	0.6392	x1936,**x3783**,x5982,x6594,x9818
Elastic Net	0.477	2	0.6285	**x3783,x6594**
	AIC			
Model	Lambda	Number of variables	AUC	
Ridge	0.000	9939	0.6929	x1-x9939
Lasso	0.087	12	0.7115	x815,x306,x754,x766,x1748,x1936,**x378**+E2x4684,**x6594,**x7544,x7891,x9818
Elastic Net	0.477	2	0.6285	**x3783,x6594**

Normalization does not change the model results dramatically, dominant variables could always be picked out. The model results of the standardized data is very close to the results on the original data. Like what we observed for the simulation data, Ridge contains all variables in the model, it is not surprising the model fitted by Ridge perform fairly well for both data sets. However keeping all the variables in the model makes the model nearly impossible to interpret. For Elastic net, all three information selection criteria pick the same model with two variables (x3783 and x6594) in both cases, the AUC for the validation data for these three models are not excellent but acceptable. If we compare this model with the model fitted by Lasso under EBIC with same variables, Elastic net model perform slightly better. The best model in terms of AUC for validation data is the one fitted by Lasso under AIC, this model also contains the most variables. Lasso model picked by BIC has 5 variables, however the AUC is not as good as the model chosen by AIC.

Retention Data

Retention project is led by Dr.Robinson in St. Cloud State University, which intends to research students' academic information and understand the important variables that have strong relationship with the success of students. By analyzing students' historical patterns, the school wants to improve the ability of predicting which student is at high risk of dropping school and make necessary intervention to ensure students' academic success based on the needs of students. For this project, Identifying important variables and building model with high prediction accuracy are equally important. We will use the academic data of fall 2010 cohort to build Ridge, Lasso and Elastic Net model to predict if the student will return to school at their third term, and try to select the optimal predicting model that easy to interpret.

Exploratory Data Analysis

The data contains 4047 observations, $\frac{2}{3}$ of the data is used to build the model, and the rest of the data serves as validation purpose. Forty-seven variables will be used to build the models including 25 nominal variables and 22 continuous variables:

Table 10

Distributions of Nominal Variables of Retention Data

Variable	3rd term retention				P-value of chi square test
	1		0		
	1	0	1	0	
ACE	22%	78%	28%	72%	***0.0002***
Honors	7%	93%	2%	98%	***0.0000***
Dist1	96%	4%	95%	5%	0.0203
Female	54%	46%	52%	48%	0.3585
International	97%	3%	98%	2%	0.0107
StudentOfColor1	11%	89%	12%	88%	0.8810
FirstGeneration1	13%	87%	14%	86%	0.5627
HS_GPA1	97%	3%	97%	3%	0.7985
HS_Rank1	81%	19%	65%	35%	***0.0000***
HS_Pct1	89%	11%	89%	11%	0.7084
HS_MnSCU_Region7	31%	69%	28%	72%	0.0329
HS_MnSCU_Region11	33%	67%	33%	67%	0.8257
HS_MnSCU_Region_OutofState	10%	90%	14%	86%	***0.0011***
HS_MnSCU_Region_Unknown	5%	95%	4%	96%	0.1070
ACT1	97%	3%	96%	4%	***0.0054***
ACT_Math1	97%	3%	94%	6%	***0.0001***
ACT_English1	97%	3%	94%	6%	***0.0001***
ACT_Reading1	97%	3%	94%	6%	***0.0001***
ACT_Science1	97%	3%	94%	6%	***0.0000***
GrantFlag1	45%	55%	43%	57%	0.1442
ScholarshipFlag1	31%	69%	23%	77%	***0.0000***
LoanFlag1	66%	34%	73%	27%	***0.0001***
WorkStudyFlag1	12%	88%	12%	88%	0.9796
EFC_Total1	74%	26%	75%	25%	0.4829
1st_Term_On_Campus1	77%	23%	75%	25%	0.2111

Table 10 above shows the distribution of nominal variables that we will use in the model. Dist1 is a missing value indicates for students who do not have an address in file. ACT1, ACT_Math1, ACT_English1, ACT_Reading1, ACT_science1, GrantFlag1, LoanFlag1, WorkStudyFlag1 and EFC_Total1 serve the same role as missing value indicators here. The variables in table 10 above shows the variables with small P-value of Chi-square test between the nominal variable and the dependent variables, the small P-values indicate these variables have potential to play an important role for the predicting modeling.

Table 11 below shows the descriptive statistics of all the continuous variables, we also conduct the T test to compare if each explanatory variable in each level is significantly different. The variables with small P-values could be important in the model we will build.

Table 11

Distributions of Continuous Variables of Retention Data

3rd Term Retention	Variable	N	Min	rst quanti	Median	Mean	Std Dev	ird Quanti	Max	P-value of T test
1	1st_Term_TermAttemptedCreditsUgrad	3055	3	14	15	14.53	1.42	16	18	***0.00000***
0	1st_Term_TermAttemptedCreditsUgrad	1232	1	13	14	14.28	1.6	15	20	
1	1st_Term_TermCompletedCreditsUgrad	3055	0	13	14	13.77	2.33	15	18	***0.00000***
0	1st_Term_TermCompletedCreditsUgrad	1232	0	7	12	10.31	5.07	14	20	
1	1st_Term_TermGPAUgrad	3055	0	2.37	2.91	2.82	0.73	3.36	4	***0.00000***
0	1st_Term_TermGPAUgrad	1232	0	1	2.07	1.96	1.18	2.97	4	
1	ACT_English2	3055	0	17	20	19.99	5.66	23	36	***0.00019***
0	ACT_English2	1232	0	16	20	19.2	6.49	23	35	
1	ACT_Math2	3055	0	18	22	21.14	5.5	24	35	***0.00000***
0	ACT_Math2	1232	0	17	21	20.02	6.34	24	34	
1	ACT_Reading2	3055	0	18	22	21.14	5.5	24	35	***0.00000***
0	ACT_Reading2	1232	0	17	21	20.02	6.34	24	34	
1	ACT_Science2	3055	0	20	22	21.25	5.17	24	35	***0.00003***
0	ACT_Science2	1232	0	19	21	20.41	6.29	24	35	
1	ACT2	3055	0	19	21	21	4.93	24	35	***0.00036***
0	ACT2	1232	0	18	21	20.34	5.69	24	33	
1	Age	3055	16	18	18	18.13	0.44	18	20	0.06208
0	Age	1231	16	18	18	18.16	0.49	18	20	
1	AppDaysBeforeTerm	3055	-4	181	244	227.21	76.1	285	417	***0.00000***
0	AppDaysBeforeTerm	1232	-2	153	223	207.05	81.63	265	417	
1	Dist2	3055	0	21.8	48.2	56.86	52.3	72.09	251	0.07079
0	Dist2	1232	0	24.5	50.7	60.07	52.76	74.59	251	
1	EFC_Total2	3055	0	0	6119	11667.96	15845.72	17178	215272	0.63160
0	EFC_Total2	1232	0	10	6232	11997.03	21885.55	16316.25	322989	
1	GrantFlag2	3055	0	0	0	1222.84	1606.49	2735.6	7184.22	0.03927
0	GrantFlag2	1232	0	0	0	1113.83	1549.96	2525	7413	
1	HS_GPA2	3055	0	2.87	3.23	3.12	0.74	3.57	4.76	***0.00000***
0	HS_GPA2	1232	0	2.71	3	2.94	0.69	3.34	4.17	
1	HS_Pct2	3055	0	37.5	56.58	53.33	27.32	74.64	99.81	***0.00000***
0	HS_Pct2	1232	0	31.28	48.19	46.2	25.05	63.53	98.71	
1	HS_Rank2	3055	0	11	64	101.94	114.28	154	702	***0.00828***
0	HS_Rank2	1232	0	0	45	91.46	118.84	137	739	
1	LoanFlag2	3055	0	0	2750	2713.1	2587.39	4750	11518	***0.00040***
0	LoanFlag2	1232	0	0	2750	3021.98	2578.51	4750	11811	
1	ScholarshipFlag2	3055	0	0	0	347.03	748.44	500	6376	***0.00000***
0	ScholarshipFlag2	1232	0	0	0	230.04	624.06	0	5738	
1	TransferCredits2	3055	0	0	0	5.4	10.17	7	71	***0.00000***
0	TransferCredits2	1232	0	0	0	3.67	8.62	3	71	
1	TransferGPA2	3055	0	0	0	0.87	1.41	2.15	4	***0.00000***
0	TransferGPA2	1232	0	0	0	0.58	1.19	0	4	
1	WorkStudyFlag2	3055	0	0	0	161.15	454.95	0	2250	0.96214
0	WorkStudyFlag2	1232	0	0	0	161.88	455.94	0	1590	

We keep ACT composite score and all the ACT subjects' scores in the model, which are highly correlated. We expect Elastic net could select the group effects, and we also want to test how Lasso and Ridge handle the collinearity.

Table 12

Correlation Matrix for ACT Scores

	ACT2	ACT_Math2	Act_English2	ACT_Reading2	ACT_Science2
ACT2	1.0000	0.8512	0.8808	0.8512	0.8640
ACT_Math2	0.8512	1.0000	0.7640	1.0000	0.8365
Act_English2	0.8808	0.7640	1.0000	0.7640	0.7860
ACT_Reading2	0.8512	1.0000	0.7640	1.0000	0.8365
ACT_Science2	0.8648	0.8365	0.7860	0.8365	1.0000

Model Results

Model results shown below, the logistic model was selected by stepwise procedure. If we look at the AUC for the validation data, all the models selected by different information criteria perform similarly to each other. EBIC and BIC choose the same model for each regression method. Elastic net models chosen by EBIC and BIC has the least number of variables, which contains all the variables that also present in all other models. We also run the models on the standardized retention data, we have the same results, which means normalization does not have great impact on the model fitting.

Table 13

Model Results for Retention Data

	EBIC		
Model	Lambda	Number of variables	AUC
Logistic	N/A	8	0.7642
Ridge	0.001	47	0.7583
Lasso	0.021	6	0.7562
Elastic Net	0.045	5	0.7533
	BIC		
Model	Lambda	Number of variables	AUC
Logistic	N/A	8	0.7642
Ridge	0.001	47	0.7583
Lasso	0.021	6	0.7562
Elastic Net	0.011	5	0.7533
	AIC		
Model	Lambda	Number of variables	AUC
Logistc	N/A	20	0.7653
Ridge	0.001	47	0.7583
Lasso	0.002	36	0.7653
Elastic Net	0.004	37	0.7662

Table 14

Variables Selection for Retention Data

	EBIC				BIC				AIC			
	Logistic	Ridge	Lasso	Elastic Net	Logistic	Ridge	Lasso	Elastic Net	Logistic	Ridge	Lasso	Elastic Net
ACE1		X				X				X	X	X
Honors1		X				X			X	X	X	X
AppDaysBeforeTerm		X				X				X		
Dist1		X				X				X		
Dist2		X				X				X	X	X
Female		X				X			X	X	X	X
Age		X				X			X	X	X	X
US Citizen		X				X				X	X	X
StudentOfColor1	X	X	X		X	X	X			X	X	X
FirstGeneration1		X				X				X	X	X
Veteran1		X				X				X		
HS_GPA1		X				X			X	X	X	X
HS_GPA2		X				X				X		X
HS_Rank1	X	X	X	X	X	X	X	X	X	X	X	X
HS_Rank2		X				X			X	X	X	X
HS_Pct1	X	X	X	X	X	X	X	X	X	X	X	X
HS_Pct2		X				X			X	X	X	X
HS_MnSCU_Region7		X				X			X	X	X	X
HS_MnSCU_Region11		X				X			X	X	X	X
HS_MnSCU_Region_OutofState	X	X	X	X	X	X	X	X	X	X	X	X
HS_MnSCU_Region_Unknown		X				X				X	X	X
ACT1		X				X			X	X		
ACT2		X				X				X	X	X
ACT_Math1		X				X				X		
ACT_Math2		X				X				X		
ACT_English1		X				X				X		
ACT_English2		X				X				X	X	X
ACT_Reading1		X				X				X		
ACT_Reading2		X				X				X		
ACT_Science1	X	X			X	X			X	X	X	X
ACT_Science2		X				X			X	X	X	X
TransferCredits2		X				X				X	X	X
TransferGPA2		X				X				X	X	X
GrantFlag1		X				X				X	X	X
GrantFlag2		X				X				X	X	X
ScholarshipFlag1		X				X				X	X	X
ScholarshipFlag2		X				X				X		
LoanFlag1		X				X				X	X	X
LoanFlag2		X				X				X		
WorkStudyFlag1		X				X				X	X	X
WorkStudyFlag2		X				X				X	X	X
EFC_Total1		X				X				X	X	X
EFC_Total2	X	X			X	X				X		
1st_Term_On_Campus1	X	X			X	X			X	X	X	X
1st_Term_TermAttemptedCreditsUgrad	X	X			X	X			X	X	X	X
1st_Term_TermCompletedCreditsUgrad		X	X	X		X	X	X	X	X	X	X
1st_Term_TermGPAUgrad		X	X	X		X	X	X	X	X	X	X

Chapter 7: Discussion

Our results from the simulation data show that Ridge, Lasso and Elastic net regression methods behave similarly to each other in terms of prediction accuracy. However, Lasso and Elastic net have substantial advantages over Ridge on variable selection; with the L1 regularization involved, Lasso and Elastic net tends to penalize the absolute size of the coefficients to zero, the larger the penalty applied, the further estimates are shrunk towards to zero. As the amount of shrinkage increase, the coefficients of less important variables reach zero first, which gives Lasso and Elastic net the feature of automatic feature selection and also yield a sparse model containing only a subset of the variables in the full model. As a result, the models generated from Lasso and Elastic net are much easier to interpret. Elastic net perform closely to Lasso regression on both prediction accuracy and variable selection, if there is no collinearity in the data. However if multi-collinearity present in the data, Elastic net will select group effects, in other words, it will keep the all the variables correlated to each other in model if all of contribute to the model significantly, Lasso tends to randomly select one of them. Ridge regression does not have the function of variable selection, no matter how big the shrinkage factor is, Ridge keeps all the variables in the model, it only shrinks the coefficients of less important variables very close to zero but will never be zero, because of the difficulties of variable selection and model interpretation, Ridge regression is not always the first choice compared to Lasso and Elastic net regression.

From the modeling fitting results of simulation data and two applications, EBIC does not hold outstanding advantages over BIC on model selection; EBIC tends to select a simple model by scarifying the prediction accuracy, which also means EBIC rules some important variables out and only keep the most dominant variables to make the model simpler. BIC has a penalty term on the number of parameters which is stricter than the terms of AIC, which leads to the simpler model favored by BIC. The simulation results shows the similar principles, AIC often risks choosing models with more variables. However for the most of the simulation results, the models with more variables chosen by AIC perform slightly better than others. If the primary goal is to select a model with high prediction accuracy, AIC is a better choice over EBIC and BIC; if the primary goal is to find important factors, BIC is a more reliable information criteria over AIC and EBIC, since BIC could effectively identify the important variables, and EBIC favors too small a model which risks ruling some important variables out. In the cases that a small set of variables are prominent, the variable selection outcomes between EBIC and BIC are very close, we observe this in case 7 and case 8 of the simulation data and the results of Arcene data. When there is no dominant variables existing in the data, EBIC would outperform BIC and it could effectively identify important variables, this conclusion is supported by the simulation results in Chen and Chen’s original study (Chen & Chen, 2012).

One thing we did not consider in the simulation study is the impact of the possible variations in the explanatory variables since all the data were generated from standard normal. However, in the real data analysis, we did consider standardizing the explanatory variables first such that each has mean of 0 and standard deviation of 1 and then fitted the models. We found that there was no obvious evidence of the normalizing effect. The standardized Retention data and the data on original scale share the same model fitting results. The only big difference we observed on the standardized Arcene data is that the model selected by EBIC contains 5 variables instead of 2, and as a result, EBIC and BIC agree on the variable selection.

There are some other regularization methods developed, these approaches are not in the scope of this paper, we will mention a few of them: Adaptive lasso is proposed by Hui Zou (2006), which is developed by assign adaptive weighted coefficients to L1 penalty, in Zou's Paper, the simulation results of Adaptive Lasso shows the advantage of computation efficiency over regular Lasso and estimates parameters consistently. SCAD is a variable selection method developed by Fan and Li (Fan & Li, 2001), which sets a boundary for the penalty function as a result of reducing bias.   It would be interesting to include them in the comparison in the future.

References

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. *Proceedings of the 2nd International Symposium on Information Theory*, pp. 257-281.

Chen, J., & Chen, Z. (2008). Extended Bayesian information criteria for model selection with large model spaces. *Biometrika, 95*(3), 759-771.

Chen, J., & Chen, Z. (2012). Extended BIC for small-n-large-p sparse GLM. *Statistica Sinica, 22*, 555-574.

Fan, J., & Li, R. (2001). Variable selection via non-concave penalized likelihood and its oracle properties. *Journal of American Statistical Association, 96*, 1348-1356.

Ghosh, J., Delampady, M., & Samanta, T. (2006). *An introduction to Bayesian analysis: Theory and methods*. Springer Science and Business Media, LLC.

Horel, A., & Kennard, R. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics, 12*(1), 55-67.

Siegmund, D. (2004). Model selection in irregular problems: Applications to mapping quantitative trait loci. *Biomertrika, 91*, 785-800.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society, 58*(1), 267-288.

Wahba, G., Golub, G., & Heath, M. (1979, May). Generalized cross-validation as a method for choosing a good ridge parameter. *Technometric, 21*(2).

Zou, H. (2006). The adaptive lasso and its oracle properties, *Journal of the American Statistical Association, 101*, 1418-1429.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, 67*, 301-332.

Appendix A

Variable Definitions for Retention Data

ACE: Academic Collegiate Excellence (ACE) program is designed to help students who did not meet the admission requirements but have potential to be successful students. We will use this as one of the variable in the model, if the student is in ACE
program, then ACE=1, otherwise, ACE=0.

Honors: The University Honors Program provides supportive and challenging learning environment for determined students to enhance the skills in analysis, synthesis and interpersonal communication, students admitted into this program have an outstanding academic background.

Female: This is the gender indictor, where 1 represents female and 0 is male.

International: A variable that tells which student is an international student.

StudentOfColor1: This dummy variable tells which student is not white.

Firstgeneration1: An indicator tells whether this student is the first college student in the family.

HS_MNSCU_region7,

HS_MNSCU_Region11,

HS_MNSCU_Region_outofstate,

HS_MNSCU_Region_unknown :

These four variables indicate the location of the high schools of students.

1st_term_on_Campus1: If the students live on campus for the 1st semester, the assign 1 to this variable, otherwise 0.

Appendix B

R Code

Case1:

```
########simulation data with 10 variables and 1000 observations.
set.seed(12345679)
x1=rnorm(500)
x2=rnorm(500)
x3=rnorm(500)
x4=rnorm(500)
x5=rnorm(500)
x6=rnorm(500)
x7=rnorm(500)
x8=rnorm(500)
x9=rnorm(500)
x10=rnorm(500)

########simulate the coefficients of the variables;
set.seed(12345679)
b1=10*runif(1)
b2=runif(1)
b3=runif(1)
b4=runif(1)
```

```
b5=runif(1)

b6=runif(1)

b7=runif(1)

b8=runif(1)

b9=runif(1)

b10=runif(1)


#####create function

z=runif(1)+b1*x1+

  b2*x2+

  b3*x3+

  b4*x4+

  b5*x5+

  b6*x6+

  b7*x7+

  b8*x8+

  b9*x9+

  b10*x10

######create reverse logit link function

pr = 1/(1+exp(-z))

y = rbinom(500,1,pr)

######create the dataset
```

```
train=data.frame(x1=x1,
        x2=x2,
        x3=x3,
        x4=x4,
        x5=x5,
        x6=x6,
        x7=x7,
        x8=x8,
        x9=x9,
        x10=x10,y)
#####create train and validation
library(caTools)
sample1=sample.split(train$y,SplitRatio=2/3)
train1=train[sample1,]
validation=train[!sample1,]
t.y=data.matrix(train1$y)
t.x=data.matrix(train1[,1:10])
v.y=data.matrix(validation$y)
v.x=data.matrix(validation[,1:10])
```

Case 2:

```
########simulation data with 10 variables and 1000 observations.

set.seed(12345679)

x1=rnorm(500)

x2=jitter(x1,factor=2500)

x3=jitter(x2,factor=2500)

x4=rnorm(500)

x5=jitter(x1,factor=7500)

x6=jitter(x1,factor=7500)

x7=rnorm(500)

x8=rnorm(500)

x9=rnorm(500)

x10=rnorm(500)


########simulate the coefficients of the variables;

set.seed(12345679)

b1=3*runif(1)

b2=runif(1)

b3=runif(1)

b4=runif(1)

b5=runif(1)

b6=runif(1)

b7=runif(1)
```

```
b8=runif(1)

b9=runif(1)

b10=runif(1)


#####create function z

z=runif(1)+b1*x1+

  b2*x2+

  b3*x3+

  b4*x4+

  b5*x5+

  b6*x6+

  b7*x7+

  b8*x8+

  b9*x9+

  b10*x10

######create reverse logit link function

pr = 1/(1+exp(-z))

y = rbinom(500,1,pr)



######create the dataset

train=data.frame(x1=x1,
```

```
        x2=x2,
        x3=x3,
        x4=x4,
        x5=x5,
        x6=x6,
        x7=x7,
        x8=x8,
        x9=x9,
        x10=x10,y=y)
#####create train and validation
library(caTools)
sample1=sample.split(train$y,SplitRatio=2/3)
train1=train[sample1,]
validation=train[!sample1,]

t.y=data.matrix(train1$y)
t.x=data.matrix(train1[,1:10])
v.y=data.matrix(validation$y)
v.x=data.matrix(validation[,1:10])
```

Case 3:

```
########simulation data with 10 variables and 1000 observations.

set.seed(12345679)

x1=rnorm(500)

x2=rnorm(500)

x3=rnorm(500)

x4=rnorm(500)

x5=rnorm(500)

x6=rnorm(500)

x7=rnorm(500)

x8=rnorm(500)

x9=rnorm(500)

x10=rnorm(500)

x11=rnorm(500)

x12=rnorm(500)

x13=rnorm(500)

x14=rnorm(500)

x15=rnorm(500)

x16=rnorm(500)

x17=rnorm(500)

x18=rnorm(500)

x19=rnorm(500)
```

```
x20=rnorm(500)
x21=rnorm(500)
x22=rnorm(500)
x23=rnorm(500)
x24=rnorm(500)
x25=rnorm(500)
x26=rnorm(500)
x27=rnorm(500)
x28=rnorm(500)
x29=rnorm(500)
x30=rnorm(500)
x31=rnorm(500)
x32=rnorm(500)
x33=rnorm(500)
x34=rnorm(500)
x35=rnorm(500)
x36=rnorm(500)
x37=rnorm(500)
x38=rnorm(500)
x39=rnorm(500)
x40=rnorm(500)
x41=rnorm(500)
```

```
x42=rnorm(500)
x43=rnorm(500)
x44=rnorm(500)
x45=rnorm(500)
x46=rnorm(500)
x47=rnorm(500)
x48=rnorm(500)
x49=rnorm(500)
x50=rnorm(500)
x51=rnorm(500)
x52=rnorm(500)
x53=rnorm(500)
x54=rnorm(500)
x55=rnorm(500)
x56=rnorm(500)
x57=rnorm(500)
x58=rnorm(500)
x59=rnorm(500)
x60=rnorm(500)
x61=rnorm(500)
x62=rnorm(500)
x63=rnorm(500)
```

```
x64=rnorm(500)

x65=rnorm(500)

x66=rnorm(500)

x67=rnorm(500)

x68=rnorm(500)

x69=rnorm(500)

x70=rnorm(500)

x71=rnorm(500)

x72=rnorm(500)

x73=rnorm(500)

x74=rnorm(500)

x75=rnorm(500)

x76=rnorm(500)

x77=rnorm(500)

x78=rnorm(500)

x79=rnorm(500)

x80=rnorm(500)

x81=rnorm(500)

x82=rnorm(500)

x83=rnorm(500)

x84=rnorm(500)

x85=rnorm(500)
```

```
x86=rnorm(500)
x87=rnorm(500)
x88=rnorm(500)
x89=rnorm(500)
x90=rnorm(500)
x91=rnorm(500)
x92=rnorm(500)
x93=rnorm(500)
x94=rnorm(500)
x95=rnorm(500)
x96=rnorm(500)
x97=rnorm(500)
x98=rnorm(500)
x99=rnorm(500)
x100=rnorm(500)

########simulate the coefficients of the variables;
set.seed(12345679)
b1=10*runif(1)
b2=10*runif(1)
b3=10*runif(1)
b4=runif(1)
```

```
b5=runif(1)

b6=runif(1)

b7=runif(1)

b8=runif(1)

b9=runif(1)

b10=runif(1)

b11=runif(1)

b12=runif(1)

b13=runif(1)

b14=runif(1)

b15=runif(1)

b16=runif(1)

b17=runif(1)

b18=runif(1)

b19=runif(1)

b20=runif(1)

b21=runif(1)

b22=10*runif(1)

b23=10*runif(1)

b24=10*runif(1)

b25=runif(1)

b26=runif(1)
```

```
b27=runif(1)
b28=runif(1)
b29=runif(1)
b30=runif(1)
b31=runif(1)
b32=runif(1)
b33=runif(1)
b34=runif(1)
b35=runif(1)
b36=runif(1)
b37=runif(1)
b38=runif(1)
b39=runif(1)
b40=runif(1)
b41=runif(1)
b42=runif(1)
b43=runif(1)
b44=runif(1)
b45=runif(1)
b46=runif(1)
b47=runif(1)
b48=runif(1)
```

```
b49=runif(1)
b50=runif(1)
b51=runif(1)
b52=runif(1)
b53=runif(1)
b54=runif(1)
b55=runif(1)
b56=runif(1)
b57=runif(1)
b58=runif(1)
b59=runif(1)
b60=runif(1)
b61=runif(1)
b62=runif(1)
b63=runif(1)
b64=runif(1)
b65=runif(1)
b66=runif(1)
b67=runif(1)
b68=runif(1)
b69=runif(1)
b70=runif(1)
```

```
b71=runif(1)
b72=runif(1)
b73=runif(1)
b74=runif(1)
b75=runif(1)
b76=runif(1)
b77=runif(1)
b78=runif(1)
b79=runif(1)
b80=runif(1)
b81=runif(1)
b82=runif(1)
b83=runif(1)
b84=runif(1)
b85=runif(1)
b86=runif(1)
b87=runif(1)
b88=runif(1)
b89=runif(1)
b90=runif(1)
b91=runif(1)
b92=runif(1)
```

```
b93=runif(1)
b94=runif(1)
b95=runif(1)
b96=runif(1)
b97=runif(1)
b98=runif(1)
b99=runif(1)
b100=runif(1)

#####create function z
set.seed(12345679)
z=runif(1)+b1*x1+
  b2*x2+
  b3*x3+
  b4*x4+
  b5*x5+
  b6*x6+
  b7*x7+
  b8*x8+
  b9*x9+
  b10*x10+
  b11*x11+
```

b12*x12+

b13*x13+

b14*x14+

b15*x15+

b16*x16+

b17*x17+

b18*x18+

b19*x19+

b20*x20+

b21*x21+

b22*x22+

b23*x23+

b24*x24+

b25*x25+

b26*x26+

b27*x27+

b28*x28+

b29*x29+

b30*x30+

b31*x31+

b32*x32+

b33*x33+

b34*x34+

b35*x35+

b36*x36+

b37*x37+

b38*x38+

b39*x39+

b40*x40+

b41*x41+

b42*x42+

b43*x43+

b44*x44+

b45*x45+

b46*x46+

b47*x47+

b48*x48+

b49*x49+

b50*x50+

b51*x51+

b52*x52+

b53*x53+

b54*x54+

b55*x55+

b56*x56+

b57*x57+

b58*x58+

b59*x59+

b60*x60+

b61*x61+

b62*x62+

b63*x63+

b64*x64+

b65*x65+

b66*x66+

b67*x67+

b68*x68+

b69*x69+

b70*x70+

b71*x71+

b72*x72+

b73*x73+

b74*x74+

b75*x75+

b76*x76+

b77*x77+

b78*x78+

b79*x79+

b80*x80+

b81*x81+

b82*x82+

b83*x83+

b84*x84+

b85*x85+

b86*x86+

b87*x87+

b88*x88+

b89*x89+

b90*x90+

b91*x91+

b92*x92+

b93*x93+

b94*x94+

b95*x95+

b96*x96+

b97*x97+

b98*x98+

```
  b99*x99+
  b100*x100



######create reverse logit link function
pr = 1/(1+exp(-z))
y = rbinom(500,1,pr)



######create the dataset
train=data.frame(x1=x1,
                 x2=x2,
                 x3=x3,
                 x4=x4,
                 x5=x5,
                 x6=x6,
                 x7=x7,
                 x8=x8,
                 x9=x9,
                 x10=x10,
                 x11=x11,
                 x12=x12,
```

x13=x13,

x14=x14,

x15=x15,

x16=x16,

x17=x17,

x18=x18,

x19=x19,

x20=x20,

x21=x21,

x22=x22,

x23=x23,

x24=x24,

x25=x25,

x26=x26,

x27=x27,

x28=x28,

x29=x29,

x30=x30,

x31=x31,

x32=x32,

x33=x33,

x34=x34,

x35=x35,

x36=x36,

x37=x37,

x38=x38,

x39=x39,

x40=x40,

x41=x41,

x42=x42,

x43=x43,

x44=x44,

x45=x45,

x46=x46,

x47=x47,

x48=x48,

x49=x49,

x50=x50,

x51=x51,

x52=x52,

x53=x53,

x54=x54,

x55=x55,

x56=x56,

x57=x57,

x58=x58,

x59=x59,

x60=x60,

x61=x61,

x62=x62,

x63=x63,

x64=x64,

x65=x65,

x66=x66,

x67=x67,

x68=x68,

x69=x69,

x70=x70,

x71=x71,

x72=x72,

x73=x73,

x74=x74,

x75=x75,

x76=x76,

x77=x77,

x78=x78,

x79=x79,

x80=x80,

x81=x81,

x82=x82,

x83=x83,

x84=x84,

x85=x85,

x86=x86,

x87=x87,

x88=x88,

x89=x89,

x90=x90,

x91=x91,

x92=x92,

x93=x93,

x94=x94,

x95=x95,

x96=x96,

x97=x97,

x98=x98,

x99=x99,

x100=x100,y=y)

```
#####create train and validation
library(caTools)
sample1=sample.split(train$y,SplitRatio=2/3)
train1=train[sample1,]
validation=train[!sample1,]
t.y=data.matrix(train1$y)
t.x=data.matrix(train1[,1:100])
v.y=data.matrix(validation$y)
v.x=data.matrix(validation[,1:100])
```

Case 4:

```
########simulation data with 10 variables and 1000 observations.
set.seed(12345679)
x1=rnorm(500)
x2=rnorm(500)
x3=rnorm(500)
x4=jitter(x1,factor=2500)
x5=jitter(x1,factor=2500)
x6=jitter(x1,factor=2500)
x7=jitter(x1,factor=2500)
x8=jitter(x1,factor=2500)
x9=jitter(x3,factor=7500)
```

```
x10=jitter(x3,factor=7500)

x11=jitter(x3,factor=7500)

x12=jitter(x3,factor=7500)

x13=jitter(x3,factor=7500)

x14=rnorm(500)

x15=rnorm(500)

x16=rnorm(500)

x17=rnorm(500)

x18=rnorm(500)

x19=rnorm(500)

x20=rnorm(500)

x21=rnorm(500)

x22=rnorm(500)

x23=rnorm(500)

x24=rnorm(500)

x25=rnorm(500)

x26=rnorm(500)

x27=rnorm(500)

x28=rnorm(500)

x29=rnorm(500)

x30=rnorm(500)

x31=rnorm(500)
```

```
x32=rnorm(500)

x33=rnorm(500)

x34=rnorm(500)

x35=rnorm(500)

x36=rnorm(500)

x37=rnorm(500)

x38=rnorm(500)

x39=rnorm(500)

x40=rnorm(500)

x41=rnorm(500)

x42=rnorm(500)

x43=rnorm(500)

x44=rnorm(500)

x45=rnorm(500)

x46=rnorm(500)

x47=rnorm(500)

x48=rnorm(500)

x49=rnorm(500)

x50=rnorm(500)

x51=rnorm(500)

x52=rnorm(500)

x53=rnorm(500)
```

```
x54=rnorm(500)
x55=rnorm(500)
x56=rnorm(500)
x57=rnorm(500)
x58=rnorm(500)
x59=rnorm(500)
x60=rnorm(500)
x61=rnorm(500)
x62=rnorm(500)
x63=rnorm(500)
x64=rnorm(500)
x65=rnorm(500)
x66=rnorm(500)
x67=rnorm(500)
x68=rnorm(500)
x69=rnorm(500)
x70=rnorm(500)
x71=rnorm(500)
x72=rnorm(500)
x73=rnorm(500)
x74=rnorm(500)
x75=rnorm(500)
```

```
x76=rnorm(500)

x77=rnorm(500)

x78=rnorm(500)

x79=rnorm(500)

x80=rnorm(500)

x81=rnorm(500)

x82=rnorm(500)

x83=rnorm(500)

x84=rnorm(500)

x85=rnorm(500)

x86=rnorm(500)

x87=rnorm(500)

x88=rnorm(500)

x89=rnorm(500)

x90=rnorm(500)

x91=rnorm(500)

x92=rnorm(500)

x93=rnorm(500)

x94=rnorm(500)

x95=rnorm(500)

x96=rnorm(500)

x97=rnorm(500)
```

```
x98=rnorm(500)
x99=rnorm(500)
x100=rnorm(500)

########simulate the coefficients of the variables;
set.seed(12345679)
b1=10*runif(1)
b2=10*runif(1)
b3=10*runif(1)
b4=b1
b5=b1
b6=b1
b7=b1
b8=b1
b9=b3
b10=b3
b11=b3
b12=b3
b13=b3
b14=runif(1)
b15=runif(1)
b16=runif(1)
```

```
b17=runif(1)
b18=runif(1)
b19=runif(1)
b20=runif(1)
b21=runif(1)
b22=10*runif(1)
b23=10*runif(1)
b24=10*runif(1)
b25=runif(1)
b26=runif(1)
b27=runif(1)
b28=runif(1)
b29=runif(1)
b30=runif(1)
b31=runif(1)
b32=runif(1)
b33=runif(1)
b34=runif(1)
b35=runif(1)
b36=runif(1)
b37=runif(1)
b38=runif(1)
```

```
b39=runif(1)

b40=runif(1)

b41=runif(1)

b42=runif(1)

b43=runif(1)

b44=runif(1)

b45=runif(1)

b46=runif(1)

b47=runif(1)

b48=runif(1)

b49=runif(1)

b50=runif(1)

b51=runif(1)

b52=runif(1)

b53=runif(1)

b54=runif(1)

b55=runif(1)

b56=runif(1)

b57=runif(1)

b58=runif(1)

b59=runif(1)

b60=runif(1)
```

```
b61=runif(1)
b62=runif(1)
b63=runif(1)
b64=runif(1)
b65=runif(1)
b66=runif(1)
b67=runif(1)
b68=runif(1)
b69=runif(1)
b70=runif(1)
b71=runif(1)
b72=runif(1)
b73=runif(1)
b74=runif(1)
b75=runif(1)
b76=runif(1)
b77=runif(1)
b78=runif(1)
b79=runif(1)
b80=runif(1)
b81=runif(1)
b82=runif(1)
```

```
b83=runif(1)
b84=runif(1)
b85=runif(1)
b86=runif(1)
b87=runif(1)
b88=runif(1)
b89=runif(1)
b90=runif(1)
b91=runif(1)
b92=runif(1)
b93=runif(1)
b94=runif(1)
b95=runif(1)
b96=runif(1)
b97=runif(1)
b98=runif(1)
b99=runif(1)
b100=runif(1)

#####create function z
set.seed(12345679)
z=runif(1)+b1*x1+
```

b2*x2+

b3*x3+

b4*x4+

b5*x5+

b6*x6+

b7*x7+

b8*x8+

b9*x9+

b10*x10+

b11*x11+

b12*x12+

b13*x13+

b14*x14+

b15*x15+

b16*x16+

b17*x17+

b18*x18+

b19*x19+

b20*x20+

b21*x21+

b22*x22+

b23*x23+

b24*x24+

b25*x25+

b26*x26+

b27*x27+

b28*x28+

b29*x29+

b30*x30+

b31*x31+

b32*x32+

b33*x33+

b34*x34+

b35*x35+

b36*x36+

b37*x37+

b38*x38+

b39*x39+

b40*x40+

b41*x41+

b42*x42+

b43*x43+

b44*x44+

b45*x45+

b46*x46+

b47*x47+

b48*x48+

b49*x49+

b50*x50+

b51*x51+

b52*x52+

b53*x53+

b54*x54+

b55*x55+

b56*x56+

b57*x57+

b58*x58+

b59*x59+

b60*x60+

b61*x61+

b62*x62+

b63*x63+

b64*x64+

b65*x65+

b66*x66+

b67*x67+

b68*x68+

b69*x69+

b70*x70+

b71*x71+

b72*x72+

b73*x73+

b74*x74+

b75*x75+

b76*x76+

b77*x77+

b78*x78+

b79*x79+

b80*x80+

b81*x81+

b82*x82+

b83*x83+

b84*x84+

b85*x85+

b86*x86+

b87*x87+

b88*x88+

b89*x89+

```
 b90*x90+
 b91*x91+
 b92*x92+
 b93*x93+
 b94*x94+
 b95*x95+
 b96*x96+
 b97*x97+
 b98*x98+
 b99*x99+
 b100*x100


######create reverse logit link function
pr = 1/(1+exp(-z))
y = rbinom(500,1,pr)

######create the dataset
train=data.frame(x1=x1,
          x2=x2,
          x3=x3,
          x4=x4,
```

x5=x5,

x6=x6,

x7=x7,

x8=x8,

x9=x9,

x10=x10,

x11=x11,

x12=x12,

x13=x13,

x14=x14,

x15=x15,

x16=x16,

x17=x17,

x18=x18,

x19=x19,

x20=x20,

x21=x21,

x22=x22,

x23=x23,

x24=x24,

x25=x25,

x26=x26,

x27=x27,

x28=x28,

x29=x29,

x30=x30,

x31=x31,

x32=x32,

x33=x33,

x34=x34,

x35=x35,

x36=x36,

x37=x37,

x38=x38,

x39=x39,

x40=x40,

x41=x41,

x42=x42,

x43=x43,

x44=x44,

x45=x45,

x46=x46,

x47=x47,

x48=x48,

x49=x49,

x50=x50,

x51=x51,

x52=x52,

x53=x53,

x54=x54,

x55=x55,

x56=x56,

x57=x57,

x58=x58,

x59=x59,

x60=x60,

x61=x61,

x62=x62,

x63=x63,

x64=x64,

x65=x65,

x66=x66,

x67=x67,

x68=x68,

x69=x69,

x70=x70,

x71=x71,

x72=x72,

x73=x73,

x74=x74,

x75=x75,

x76=x76,

x77=x77,

x78=x78,

x79=x79,

x80=x80,

x81=x81,

x82=x82,

x83=x83,

x84=x84,

x85=x85,

x86=x86,

x87=x87,

x88=x88,

x89=x89,

x90=x90,

x91=x91,

x92=x92,

```
        x93=x93,
        x94=x94,
        x95=x95,
        x96=x96,
        x97=x97,
        x98=x98,
        x99=x99,
        x100=x100,y=y)
#####create train and validation
library(caTools)
sample1=sample.split(train$y,SplitRatio=2/3)
train1=train[sample1,]
validation=train[!sample1,]
t.y=data.matrix(train1$y)
t.x=data.matrix(train1[,1:100])
v.y=data.matrix(validation$y)
v.x=data.matrix(validation[,1:100])
```

For cases 5, 6, 7, 9, the simulation code is set up in the same way as code for cases 3 and 4 with increased number of variables.

Code for model fitting and information criteria:

```
#import important packages;
library(MASS)
library(gplots)
library(ROCR)
library(Matrix)
library(glmnet)
library(elasticnet)




#ridge model
#########select ridge model tuning  parameter with AIC, BIC and EBIC
ridge.t=glmnet(t.x,t.y,alpha=0,lambda=seq(0,1,by=0.001),family="binomial")



dev=deviance(ridge.t)


nvar=ridge.t$df


lambda=ridge.t$lambda
```

```
#lasso model
#########select lasso model tuning  parameter with AIC, BIC and EBIC
lasso.t=glmnet(t.x,t.y,alpha=1,lambda=seq(0,1,by=0.001),family="binomial")
dev=deviance(lasso.t)

nvar=lasso.t$df

lambda=lasso.t$lambda

#elastic net model
#########select ridge model tuning  parameter with AIC, BIC and EBIC
enet.t=glmnet(t.x,t.y,alpha=0.5,lambda=seq(0,1,by=0.001),family="binomial")

dev=deviance(enet.t)

nvar=enet.t$df

lambda=enet.t$lambda


#Calcualte AIC, BIC and EBIC
#p is the total number of variables for each simulation data.
```

```
p=10

#n is the number of observations in each simulation data

n=100


dat=data.frame(deviance=dev,lambda=lambda,nvar=nvar)


dat$aic=dat$deviance+2*dat$nvar


dat$bic=dat$deviance+dat$nvar*log(0.333*n)



dat$ebic=dat$bic+2*0.25**log(choose(p,dat$nvar))


dat1=subset(dat,nvar!=0)





#EBIC

ebic1=subset(dat1,ebic!="-Inf")

ebic=subset(ebic1,ebic==min(ebic1$ebic))

ebic
```

```
#BIC
bic=subset(dat1,bic==min(dat1$bic))
bic

#AIC
aic=subset(dat1,aic==min(dat1$aic))
aic



#Calcualte AIC, BIC and EBIC
dat=data.frame(deviance=dev,lambda=lambda,nvar=nvar)

dat$aic=dat$deviance+2*dat$nvar

dat$bic=dat$deviance+dat$nvar*log(2858)

dat$k=log((dat$nvar),base=2858)

dat$theta=1-1/(2*(dat$k))

dat$ebic=dat$bic+2*(dat$theta)*log(choose(2858,dat$nvar))
```

```
dat1=subset(dat,nvar!=0)

#EBIC
ebic1=subset(dat1,ebic!="-Inf")
ebic=subset(ebic1,ebic==min(ebic1$ebic))
ebic

#BIC
bic=subset(dat1,bic==min(dat1$bic))
bic

#AIC
aic=subset(dat1,aic==min(dat1$aic))
aic
```

Calculate AUC for validation data:

```
###########fit ridge model with selected tuning  parameter
ridge.t=cv.glmnet(t.x,t.y,alpha=0,lambda=seq(0,1,by=0.001),nfolds=5,family="binomial",type.measure="class")

coeft=coef(ridge.t$glmnet.fit,s=bic$lambda)
coeft
```

```
#Plot the ROC curve
predictridget=predict(ridge.t,newx=v.x,type="response",s=bic$lambda)
write.csv(predictridget,file="F:\\Thesis\\predictridget.csv")

pred <- prediction(predictridget[,1], v.y)

perf <- performance(pred, measure = "tpr", x.measure = "fpr")
title(main="ROC curve for ridge regression")
plot(perf, col=rainbow(10))

auc.tmp <- performance(pred,"auc")

auc <- as.numeric(auc.tmp@y.values)

auc

###########fit ridge model with selected tuning  parameter
lasso.t=cv.glmnet(t.x,t.y,alpha=1,lambda=seq(0,1,by=0.001),nfolds=5,family="binomial",typ
e.measure="class")

#export the coef
coeft.lasso=coef(lasso.t$glmnet.fit,s=bic$lambda)
```

```
coeft.lasso<-data.matrix(coeft.lasso)

write.csv(coeft.lasso,file="F:\\Thesis\\coeflasso.csv")




#Plot the ROC curve

predictlasso=predict(lasso.t,newx=v.x,type="response",s=bic$lambda)

write.csv(predictlasso,file="F:\\Thesis\\predictlasso.csv")


pred <- prediction(predictlasso[,1], v.y)


perf <- performance(pred, measure = "tpr", x.measure = "fpr")

title(main="ROC curve for ridge regression")

plot(perf, col=rainbow(10))


auc.tmp <- performance(pred,"auc")


auc <- as.numeric(auc.tmp@y.values)


auc
```

```
###########fit ridge model with selected tuning  parameter
enet.t=cv.glmnet(t.x,t.y,alpha=0.5,lambda=seq(0,2,by=0.001),nfolds=5,family="binomial",ty
pe.measure="class")



#export the coef
coeft.enet=coef(enet.t$glmnet.fit,s=bic$lambda)
coeft.enet<-data.matrix(coeft.enet)
write.csv(coeft.enet,file="F:\\Thesis\\coefenet.csv")



#Plot the ROC curve
predictenet=predict(enet.t,newx=v.x,type="response",s=aic$lambda)
write.csv(predictenet,file="F:\\Thesis\\predictenet.csv")

pred <- prediction(predictenet[,1], v.y)

perf <- performance(pred, measure = "tpr", x.measure = "fpr")
title(main="ROC curve for ridge regression")
plot(perf, col=rainbow(10))
```

```
auc.tmp <- performance(pred,"auc")

auc <- as.numeric(auc.tmp@y.values)

auc

#code for standardized data
#ridge model
#########select ridge model tuning  parameter with AIC, BIC and EBIC
ridge.t=glmnet(t.x,t.y,alpha=0,lambda=seq(0,1,by=0.001),family="binomial",standardize =
TRUE)


dev=deviance(ridge.t)

nvar=ridge.t$df

lambda=ridge.t$lambda

#lasso model
#########select lasso model tuning  parameter with AIC, BIC and EBIC
```

```
lasso.t=glmnet(t.x,t.y,alpha=1,lambda=seq(0,1,by=0.001),family="binomial",standardize =
TRUE)
dev=deviance(lasso.t)

nvar=lasso.t$df

lambda=lasso.t$lambda

#elastic net model
#########select ridge model tuning  parameter with AIC, BIC and EBIC
enet.t=glmnet(t.x,t.y,alpha=0.5,lambda=seq(0,1,by=0.001),family="binomial",standardize =
TRUE)

dev=deviance(enet.t)

nvar=enet.t$df

lambda=enet.t$lambda



#Calcualte AIC, BIC and EBIC
#p is the total number of variables for each simulation data.
```

```
p=47

#n is the number of observations in each simulation data

n=4287


dat=data.frame(deviance=dev,lambda=lambda,nvar=nvar)


dat$aic=dat$deviance+2*dat$nvar


dat$bic=dat$deviance+dat$nvar*log(0.667*n)



dat$ebic=dat$bic+2*0.25**log(choose(p,dat$nvar))


dat1=subset(dat,nvar!=0)


#EBIC

ebic1=subset(dat1,ebic!="-Inf")

ebic=subset(ebic1,ebic==min(ebic1$ebic))

ebic
```

```
#BIC

bic=subset(dat1,bic==min(dat1$bic))

bic


#AIC

aic=subset(dat1,aic==min(dat1$aic))

aic
```