STATISTICAL METHODS FOR VARIABLE SELECTION IN THE CONTEXT OF HIGH-DIMENSIONAL DATA: LASSO AND EXTENSIONS

STATISTICAL METHODS FOR VARIABLE SELECTION IN THE CONTEXT OF HIGH-DIMENSIONAL DATA: LASSO AND EXTENSIONS

BY

XIAO DI YANG, B.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF MATHEMATICS & STATISTICS

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

Master of Science (2011) McMaster University
(Mathematics & Statistics) Hamilton, Ontario, Canada

TITLE:	STATISTICAL METHODS FOR VARIABLE SELECTION IN THE CONTEXT OF HIGH-DIMENSIONAL DATA: LASSO AND EXTENSIONS
AUTHOR:	XIAO DI YANG B.Sc., (Statistics) University of Waterloo, Waterloo, Canada
SUPERVISOR:	Dr. Joseph Beyene
NUMBER OF PAGES:	xi, 68

To my parents

Abstract

With the advance of technology, the collection and storage of data has become routine. Huge amount of data are increasingly produced from biological experiments. the advent of DNA microarray technologies has enabled scientists to measure expressions of tens of thousands of genes simultaneously. Single nucleotide polymorphism (SNP) are being used in genetic association with a wide range of phenotypes, for example, complex diseases. These high-dimensional problems are becoming more and more common. The "large p, small n" problem, in which there are more variables than samples, currently a challenge that many statisticians face. The penalized variable selection method is an effective method to deal with "large p, small n" problem. In particular, The Lasso (least absolute selection and shrinkage operator) proposed by Tibshirani has become an effective method to deal with this type of problem. the Lasso works well for the covariates which can be treated individually. When the covariates are grouped, it does not work well. Elastic net, group lasso, group MCP and group bridge are extensions of the Lasso. Group lasso enforces sparsity at the group level, rather than at the level of the individual covariates. Group bridge, group MCP produces sparse solutions both at the group level and at the level of the individual covariates within a group. Our simulation study shows that the group lasso forces complete grouping, group MCP encourages grouping to a rather slight extent, and group bridge is somewhere in between. If one expects that the proportion of nonzero group members to be greater than one-half, group lasso maybe a good choice; otherwise group MCP would be preferred. If one expects this proportion to be close to one-half, one may wish to use group bridge. A real data analysis example is also conducted for genetic variation (SNPs) data to find out the associations between SNPs and West Nile disease.

Acknowledgements

I am heartily thankful to my supervisor, Dr. Joseph Beyene, whose constant support, guidance and patience from the initial to the final level enabled me to complete this thesis.

I would also like to thank Dr. Narayanaswamy Balakrishnan and Dr. Aaron Childs for serving on my project examination committee and their suggestions on the thesis.

I am grateful to my parents and grandparents, for their continuous inspiration, encouragement and support throughout my graduate study.

Contents

List of Tables

List of Figures

Chapter 1

Introduction and Problem Statement

1.1 Background

In recent years, there has been a huge development of comprehensive, high-througput methods for molecular biology experimentation. The advent of DNA microarray technologies has enabled scientists to measure the expressions of tens of thousands of genes simultaneously. High-density DNA microarray technology allows researchers to monitor the interactions among thousands of gene transcripts in an organism on a single experimental medium. Prior to this technology, researchers were limited to examinations of much smaller numbers of genetic units per experiment and were able to assess interactions among genes under changing conditions on a much smaller scale.

Single nucleotide polymorphism (SNP) based association studies aim at identifying SNPs associated with phenotypes, for example, complex diseases. The SNPs may associate with disease individually as main effects or behave jointly as epistatic interactions.

For the analysis of high throughout data, the main difficulty is that the number of variables, for example, SNPs or genes far exceeds the number of samples due to the high cost of microarray experimens. This difficulty is amplified if we want to identify interactions. We need to eliminate the non-important variables and retain a subset of variables that explain the most important effects. West et. al (2001) defined it as the "large p, small n" problem in their paper. As the number of samples n is usually about tens or hundreds but the number of variables, p is usually about thousands or ten thousands, problems arise when fitting regression models.

1. *infinitely many solutions*: if $p > n$, there will be more unknowns than equations,and there maybe infinitely many solutions.

2. *model over-fitting*: the model will fit the training data well but not the testing data.

3. *multicollinearity*: many genes will show nearly identical patterns across the samples, so they supply no new information; some gene profiles can be linear combinations of the other gene profiles.

Variable selection methods have been studied extensively in the literature. See George and McCulloch, 1993; Foster and George, 1994; Breiman, 1995; Tibshirani, 1996; George and Foster, 2000; Fan and Li, 2001; Shen and Ye, 2002; Efron, Hastie, Johnstone and Tibshirani, 2004; Zou and Hastie, 2005; Lin and Zhang, 2006; and Wu, Boos and Stefanski, 2007. In particular, the Lasso proposed by Tibshirani has gained much attention in the past few years. The problem that we described above can be addressed by introducing a penalty into the regression model. Tibshirani (1996) proposed a new method for estimation in linear models. The Lasso minimizes the residual sum of squares subject to the sum of the absolute value of the coefficients being less than a constant. Because of the nature of this constraint, it forces some coefficients to be exactly 0 and hence eliminate the irrelevant genes or SNPs in the model. The ℓ_1-type penalty of the Lasso can also be applied to other models as for example Cox regression (Tibshirani, 1997), logistic regression (Lokhorst, 1999; Roth, 2004; Shevade and Keerthi, 2003; Genkin et al., 2007) or multinomial logistic regression (Krishnapuram et al., 2005) by replacing the residual sum of squares by the corresponding negative log-likelihood function.

Another problem encountered by researchers is that in practical problems, sometimes the predictors are dependent with each other and thus are "grouped". For example, in ANOVA factor analysis, a factor may have several levels and can be expressed via several dummy variables. The dummy variables of the factor become a group. Also, in additive models, each original predictors may be expanded into different order polynomials or a set of basis functions, then these polynomials (or basis functions) corresponding to the same original prediction variable become a "group" as well. In biological applications, genes that share a common biological function or participate in the same metabolic pathway have a high pairwise correlation with each other. So genes that share a common biological functions or participate in the same biological pathways become a group.

When genes are grouped, it is usually sub-optimal to ignore the group structure and apply the penalized regression. For example, suppose the k^{th} group is unimportant to the response, by using the lasso method, only the individual coefficient in the k^{th} group will be set to 0, rather than the whole group to be zero. So the lasso performs variable selection to the individual variable rather than to the group, often resulting in selecting more unimportant variables.

The group lasso (Bakin, 1999; Cai, 2001; Antoniadis and Fan, 2001; Yuan and Lin, 2006) overcomes these problems by introducing a suitable extension. Yuan & Lin (2006) proposed a group lasso which is an extension of the lasso to do variable selection on (predefined) groups of variables in linear regression models. The penalty function is comprised of ℓ_2 norms of the groups. This has the effect of encouraging sparsity at the group level while applying ridge regression-like penalty within a group. The group lasso has its weakness in that it is unable to do variable selection at the individual level and heavily shrinks large coefficients. Meier et al.(2008) extended this idea to logistic regression, and Zhao et al. (2006) extended the idea to hierarchical groups and overlapping cases.

Other methods that have been proposed and accommodate selection at the group level include: bridge (Frank and Friedman, 1993), smoothly clipped absolute deviation penalty (SCAD, Fan and Li (2001)) and minimax concave penalty (MCP, Zhang (2007)). These approaches all perform variable selection at group level but not at individual level.

Huang et al. (2009), in contrast, proposed a group bridge approach performing variable selection by encouraging sparse solutions at both the group and individual levels. The group bridge applies a bridge penalty to the ℓ_1 norm of the groups. Breheny and Huang (2009) defined selecting important groups as well as identifying important members of these groups as bi-level selection. They proposed a new method called group MCP which also performs variable selection at both the group and individual levels. They investigated the group lasso, group bridge and group MCP and introduced a new framework for thinking about group penalties. They also used the idea of a locally approximated coordinate descent to develop algorithms which are fast and stable for the three methods. The algorithms are included in the R package "grpreg" as "gMCP, gBridge and gLasso".

The recent successes in association mapping of disease genes have been propelled by logistic regression using cases and controls. Many researchers have applied penalized method to logistic regression. Park and Hastie (2007) used penalized logistic regression for detecting gene interactions. They used logistic regression with ridge regularization to fit gene-gene and gene-environment interaction models. Studies have shown that many common diseases are influenced by interaction of certain genes. Logistic regression models with ridge penalization not only can correctly characterize the influential genes along with their interaction structures, but also are good at handling high-dimensional, binary outcome data. Wu, Chen and Hastie (2009) did a genomewide association analysis by the lasso penalized regression. They evaluated the performance of lasso penalized logistic regression in case-control disease gene mapping with large number of SNP predictors. Recently, Meier, Geer and Bühlmann (2008) extended the group lasso to logistic regression and created an efficient algorithm for the method. Breheny and Huang (2009) extended the group lasso, group bridge, group MCP to logistic regression under their algorithm as well.

1.2 Scope of the Project

In the next few chapters, we will discuss the underlying theory and applications of the Lasso, elastic net, group lasso, group bridge, group MCP with logistic regression. Figure 1.1 is a graph describing the framework of different variable selection methods

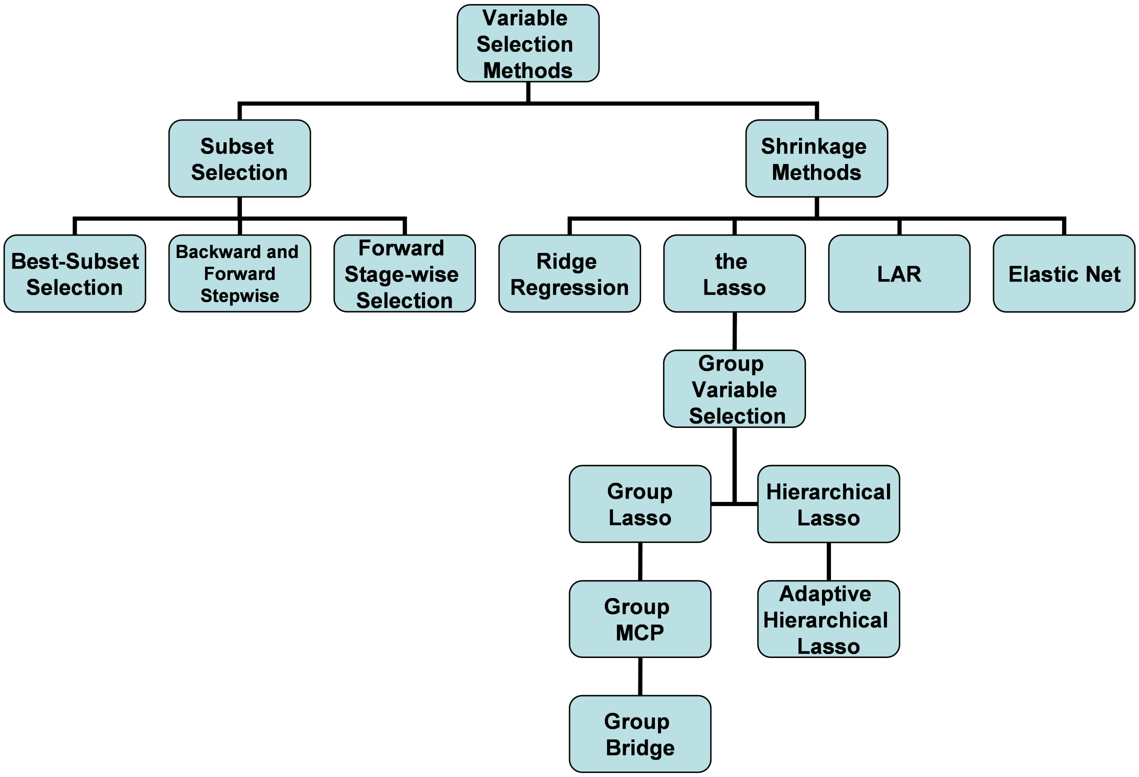

Figure 1.1: Framework of different variable selection methods

that we are going to discuss in this thesis.

Specifically, in Chapter 2, we conduct a literature review on penalized regression methods including the Lasso, elastic net, group lasso, group MCP, group bridge and hierarchical lasso. We describe each of them in terms of their penalties, penalty parameter selection and algorithm. We also discuss about advantages and disadvantages when they perform variable selection.

In chapter 3, we carry out a simulation study to compare the different variable selection methods. We compare them across a wide range of scenarios: Non-grouped and grouped variable cases with different correlation, magnitude of effects, grouping structure and sample sizes are considered. We find out that the group lasso, group bridge and group MCP are optimal for grouped variables selection and should be used depending on grouping structures of data.

In chapter 4, a real data analysis using the Lasso logistic regression is conducted for genetic variation (SNPs) data. We detect SNPs that are associated with West Nile virus disease.

Finally, in chapter 5, discussions and conclusions are given. Suggestions for further research are outlined.

Chapter 2

Preliminary Theory

2.1 Ordinary Least Squares

Suppose we have an input vector $X^T = (X_1, X_2, ..., X_p)$, and want to predict a real-valued output Y. The linear regression model has the form:

$$f(X) = \beta_0 + \textstyle\sum_{j=1}^{p} X_j \beta_j.$$

Here the β_j's are unknown parameters which can be estimated. The linear model either assumes that the regression function $E(Y|X)$ is linear, or that the linear model is a reasonable approximation.

Typically we have a set of data $(x_1, y_1)...(x_n, y_n)$ from which we have to estimate the parameters β. Each $x_i = (x_{i1}, x_{i2}, ..., x_{ip})^T$ is a vector of feature measurements for the i^{th} case. The most popular estimation method is least squares, in which we choose the coefficients $\beta = (\beta_0, \beta_1, ..., \beta_p)^T$ to minimize the residual sum of squares

$$RSS(\beta) = \textstyle\sum_{i=1}^{n} (y_i - f(x_i))^2 = \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j)^2$$

From a statistical point of view, this criterion is reasonable if the training observations (x_i, y_i) represent independent random draws from their population. Even if the x_i's were not drawn randomly, the criterion is still valid if the y_i's are conditionally independent given the inputs x_i.

To minimize RSS, we denote by X the $N \times (p+1)$ matrix with each row an input vector (with a 1 in the first position), and similarly let y be the N-vector of outputs in the data set. Then we can write the residual sum-of-squares as

$$RSS(\beta) = (y - X\beta)^T (y - X\beta) \tag{2.1}$$

This is a quadratic function in the $p+1$ parameters. Differentiating with respect to β we obtain

$$\tfrac{\partial RSS}{\partial \beta} = -2X^T(y - X\beta)$$

$$\tfrac{\partial^2 RSS}{\partial \beta \partial \beta^T} = 2X^T X$$

Assuming (for the moment) that X has full column rank, and hence $X^T X$ is positive definite, we set the first derivative to zero

$$X^T(y - X\beta) = 0$$

to obtain the unique solution

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

2.1.1 The Bias-Variance Tradeoff

To fit a group of data, we usually come up with the model:

$$y = f(x) + \varepsilon, \quad \varepsilon \sim (0, \sigma^2)$$

In regression analysis, we want to find a good regression model $\hat{f}(x) = x^T\hat{\beta}$. We usually use least squares estimate for parameters: $\hat{\beta}^{ls}$. The predicted model is a good model if it satisfies two conditions:

1. $\hat{\beta}$ is close to the true β
2. $\hat{f}(x)$ fit future observations well

For the first condition, let's consider the mean squared error of our estimate $\hat{\beta}$. i.e. consider the squared distance of $\hat{\beta}$ to the true β

$$MSE(\hat{\beta}) = E[\|\hat{\beta} - \beta\|^2] = E[(\hat{\beta} - \beta)^T(\hat{\beta} - \beta)]$$

For example, in least squares(LS), we have:

$$E[(\hat{\beta}^{ls} - \beta)^T(\hat{\beta}^{ls} - \beta)] = \sigma^2 tr[(X^TX)^{-1}]$$

Knowing that the model fits the current data well is not enough. We also need to know whether the model fits a new group of data well. So if $\hat{f}(.)$ is a good model, $\hat{f}(.)$ should be close to the new data y_i's. This is the second condition and it's denoted by prediction error (PE). So good estimators should, on average, have small prediction errors.

The PE at a particular target point x_0 is:

$$\begin{aligned} PE(x_0) &= E_{Y|X=x_0}((Y - \hat{f}(X))^2 | X = x_0) \\ &= \sigma_\varepsilon^2 + Bias^2(f(x_0)) + Var(\hat{f}(x_0)) \end{aligned}$$

This is called bias-variance tradeoff. As model becomes more complex (more terms included), local structure/curvature can be picked up. However, as more terms included in the model, the coefficient estimates suffer from high variance. We can introduce a little bias in our estimate for β to reduce the variance and hence reduce the overall PE.

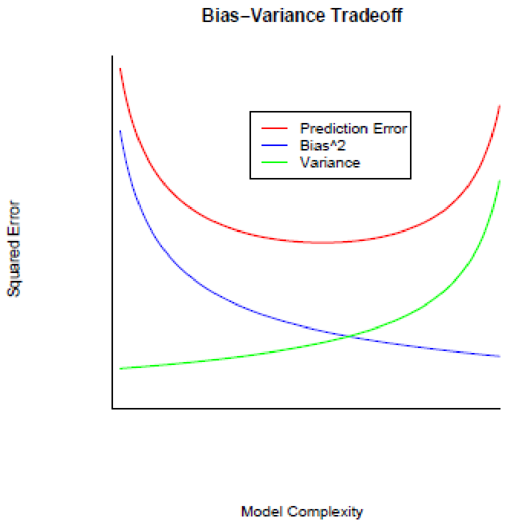

Figure 2.1: Bias-Variance Tradeoff.

[Source: adapted from “Ridge regression and the Lasso”, by Rudy Angeles, 2006, retrieved from website: http://www-stat.stanford.edu/ owen/courses/305/]

There are two challenges when we use the OLS estimates.

- *prediction accuracy*: We would like our model to accurately predict future data. The least squares estimates often have low bias but large variance. Prediction accuracy can often be improved by shrinking the regression coefficients. Shrinkage sacrifices some bias to reduce the variance of the predicted value and hence may improve the overall prediction.

- *interpretation*: With a large number of prediction variables, we would like to determine a small subset of variables that exhibits the strongest effects.

2.2 Subset Selection

2.2.1 Best-Subset Selection

In this section we describe a number of approaches to variable subset selection with linear regression. With subset selection we retain only a subset of the variables, and eliminate the rest from the model. Least squares regression is used to estimate the coefficients of the inputs that are retained. There are a number of different strategies for choosing the subset. In later sections we discuss shrinkage approaches for controlling variance.

Best subset selection regression finds for each $k \in \{0, 1, 2, ..., p\}$ the subset of size k that gives smallest residual sum of squares (RSS). This method uses the algorithm of leap-and-bound procedure of Furnival and Wilson (1974). This method is feasible for as many as 40 parameters. To determine k, we need to consider the tradeoff between bias and variance, along with the more subjective desire for parsimony. Typically, we choose the smallest model that minimizes an estimate of the expected prediction error.

2.2.2 Forward- and Backward-Stepwise Selection

The best-subset selection search through all possible subsets, becomes infeasible for p larger than 40. Instead of searching through all possible subsets, forward-stepwise selection starts with the intercept, and then sequentially adds into the model the predictor that most improves the fit. Like best-subset regression, forward-stepwise produces a sequence of models indexed by k, the subset size, which must be determined.

Forward-stepwise selection is a greedy algorithm, producing a nested sequence of models. In this sense it might seem sub-optimal compared to best-subset selection. However, it always can be computed even if the number of parameters is greater than observations($p > n$). Forward-stepwise is a more constrained search than best-subset selection in terms of selecting the best subset of each size. It will have low variance but perhaps more bias.

Backward-stepwise selection starts with the full model, and sequentially deletes the variables that are least important to the response. The candidate for dropping is the variable with the smallest Z-score. Backward selection can only be used when $n > p$, while forward-stepwise can always be used.

Some software packages implement hybrid stepwise-selection strategies that consider both forward and backward moves at each step, and select the "best" of the two. For example in the R package the step function uses the AIC criteria to select the "best" of the two in each step.

2.2.3 Forward-Stagewise Regression

Forward-stagewise regression is even more constrained than forward-stepwise regression. It starts like forward-stepwise regression, with an intercept equal to $\overline{y}$, and centered predictors with coefficients initially all 0. At each step, the algorithm identifies the variable most correlated with the current residual. It then computes the simple linear regression coefficient of the residual on this chosen variable, and then adds it to the current coefficient for that variable. This is continued till none of the variables have correlation with the residuals. i.e. the least-squares fit when $n > p$.

Unlike forward-stepwise regression, none of the other variables are adjusted when a term is added to the model. As a result, forward-stagewise regression can take more than p steps to reach the least squares fit, and historically has been dismissed as being inefficient.

2.3 Shrinkage Methods

By retaining a subset of the important predictors, subset selection produces a model that has low prediction error and is interpretable. However, it has a high variance due to its discrete process of either eliminating or retaining the variable. So it does not reduce the prediction error of the full model. Shrinkage methods are more continuous and do not suffer as much from high variability.

2.3.1 Ridge Regression

Ridge regression shrinks the regression coefficient by imposing a penalty parameter on them. The ridge regression minimize the sum of squares,

$$minimize \quad \textstyle\sum_{i=1}^{n}(y_i - \beta^T x_i)^2 \quad s.t. \textstyle\sum_{j=1}^{p} {\beta_j}^2 \leq t$$

which is equivalent to

$$minimize \quad (y - X\beta)^T(y - X\beta) \quad s.t. \textstyle\sum_{j=1}^{p} {\beta_j}^2 \leq t$$

We can write the ridge constraint as the following residual sum of squares(SS):

$$\begin{aligned} PRSS(\beta)_{\ell_2} &= \textstyle\sum_{i=1}^{n}(y_i - x_i^T\beta)^2 + \lambda \textstyle\sum_{j=1}^{p} {\beta_j}^2 \\ &= (y - X\beta)^T(y - X\beta) + \lambda \|\beta\|^2 \end{aligned}$$

The ridge solution have smaller average PE than $\hat{\beta}^{ls}$. Since $PRSS(\beta)_{\ell_2}$ is convex, it hence has a unique solution. To find the solution, we take derivative to $PRSS(\beta)_{\ell_2}$:

$$\frac{\partial PRSS(\beta)_{\ell_2}}{\partial \beta} = -2X^T(y - X\beta) + 2\lambda\beta$$

The solution to $PRSS(\hat{\beta})_{\ell_2}$ can be written as

$$\hat{\beta_\lambda}^{ridge} = (X^TX + \lambda I_p)^{-1}X^Ty$$

where X is standardized, and y is centered.

The parameter λ is the penalty parameter that controls the size of coefficients and amount of regularization. The λ forces coefficients to go to zero (but not equal to zero). The larger the penalty enforced, the smaller the coefficient will. As λ goes to zero, we obtain the least squares solutions. As λ goes to infinity, we have $\hat{\beta}^{ridge}_{\lambda=\infty} = 0$ (intercept-only model). For each λ, there is a solution. Hence, the λ's trace out a path of solutions.

Ridge regression is a continuous process that shrinks coefficients and hence is more stable: however, it does not set any coefficients to 0 and hence does not give an easily interpretable model.

2.3.2 The Lasso

Tibshirani (1996) proposed a new technique, called the Lasso, for 'least absolute shrinkage and selection operator'. It shrinks coefficients towards 0 and sets some of the coefficients exactly to 0. It tries to retain the good features of both subset selection and ridge regression. Lasso coefficients are solutions to the ℓ_1 optimization problem:

$$minimize \quad (y - X\beta)^T(y - X\beta) \quad s.t. \quad \Sigma_{j=1}^{p}|\beta_j| \leq t$$

We can write the Lasso constraint as the following penalized residual sum of squares (PRSS):

$$\begin{aligned} PRSS(\beta)_{\ell_1} &= \textstyle\sum_{i=1}^{n}(y_i - x_i^T\beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \\ &= (y - X\beta)^T(y - X\beta) + \lambda \|\beta\|_1 \end{aligned}$$

Quadratic programming techniques from convex optimization can be used to solve for values of $\hat{\beta}_\lambda^{lasso}$. $\hat{\beta}_\lambda$ has no closed form solution. The tuning parameter λ is the shrinkage parameter that controls the amount of regularization. If $t_0 = \sum_{j=1}^{p} |\hat{\beta}_j|$, (equivalently, $\lambda = 0$), there is no penalty put on the coefficients and hence we obtain the least squares solutions. If $\lambda \to \infty$, the penalty is infinitely large and thus forces all of the coefficients to be zero. Hence, we obtain a intercept-only model. Since $\Sigma_{j=1}^{p}|\beta_j| \leq t$, a path of solution is traced out by index t. Figure 2.2 below shows a Lasso coefficients paths for the brown fat data set which is publicly available at http://www.ssc.ca/en/education/archived-case-studies/ssc-case-studies-2011-fat#Data for identifying the factors that determine the existence and the volume of brown fat in humans.

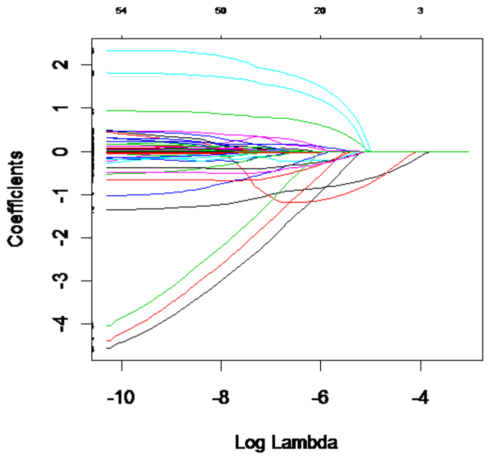

Figure 2.2: the Lasso coefficients paths for the brown fat data set

The tuning parameter λ controls the strength of penalty, which shrinks each β_j towards zero. Often, we believe that many of the $\beta_j's$ should be 0. Hence, the Lasso produces sparse solutions. Large enough λ will set some coefficients exactly equal to 0. So the Lasso will perform model selection. A ridge penalty $\lambda \sum_{j=1}^{p} \beta_j^2$ also shrinks parameter estimates towards zero, but it never set coefficients exactly equal to 0. So it is not actually performing variable selection as it is not forcing many estimates to vanish. This defect of the ridge penalty reflects the fact that $|\beta|$ is much larger than β^2 for small β.

2.3.3 Cross-Validation

We need a disciplined way of choosing the tuning parameter λ. Obviously, we want to choose λ that minimizes the mean squared error. We compute our statistical model as $\hat{f}(*)$ from training set. The model $\hat{f}(*)$ is tested on a new independent set of data named test set. The prediction should be good if we have a good model. Ideally, we would separate our data set to training set and test sets. However, this is not always possible, for example, a data set with only a few observations will not allow one to divide the data into a training and test sets.

The tuning parameter λ can be determined by cross-validation method. The most common approach used is K-fold cross validation. In K-fold cross-validation, we randomly partition our data into K sub-sets, i.e. $D = (D_1, D_2, ..., D_K)$. Usually, $K = 5$ or $K = 10$. One of the sub-set from K sub-sets is retained as test set and the remaining $K - 1$ sub-set are used as training set. We fit our statistical model $\hat{f}_{-K}^{(\lambda)}(x)$ to the training set $D' = D_1, D_2, ...D_{K-1}$. Then we compute the fitted values of the model $\hat{f}_{-K}^{(\lambda)}(x)$ to the data of test set D_K. We also compute the cross-validation (CV) error of test set D_K. The cross-validation (CV) error for the K^{th} fold is

$$(CVError)_K^{(\lambda)} = |D_K|^{-1} \sum_{(x,y)\in D_K} (y - \hat{f}_{-K}^{\lambda}(x))^2$$

Therefore, the overall cross-validation error for the model is

$$(CVError)^{\lambda} = K^{-1} \sum_{k=1}^{K} (CVError)_k^{\lambda}$$

We select λ^{cv} as the one with minimum $(CVError)^{\lambda}$. Then, we can compute the chosen model $\hat{f}(x)^{\lambda^{cv}}$ on the entire training set $D = (D_1, D_2, ...D_K)$ and apply $\hat{f}(x)^{\lambda}$ to the test set to assess test error and prediction. Figure 2.3 is a plot of CV errors and standard error bands on the brown fat data set.

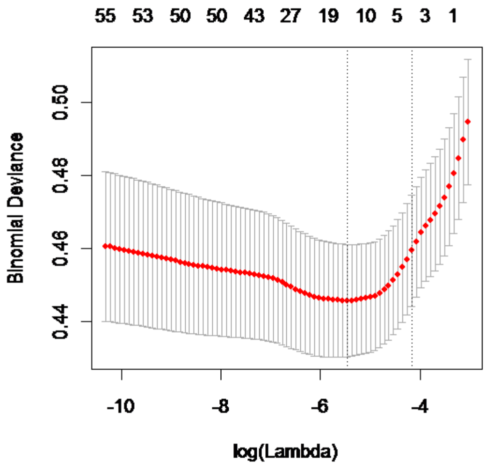

Figure 2.3: Cross-validation errors from a Lasso regression example on brown fat data set

2.3.4 Grouping Effect

In the "large p, small n" problem (West et al., 2001), the "grouped variables" situation is a particular concern. In some problems, the predictors belong to pre-defined groups.

For example, genes that belong to the same biological pathway, or collections of indicator (dummy) variables for representing the levels of a categorical predictor. In this situation it may be desirable to shrink and select the members of a group together.

Suppose we have data $(x_1, y_1), ..., (x_i, y_i), ..., (x_n, y_n)$, where $x_i = (x_{i1}, ..., x_{ip})$ are the predictors and y_i is the response. The linear regression to model the response y in terms of the predictors $x_1, ...x_p$ is:

$$y = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p + \varepsilon,$$

where ε is the error term. We now describe the cases where the variables are grouped. Suppose the predictor variables can be divided into K groups and the k^{th} group contains p_k variables. So the linear regression model becomes:

$$y_i = \beta_0 + \Sigma_{k=1}^{K} \Sigma_{j=1}^{p_k} \beta_{kj} x_{i,kj} + \epsilon_i.$$

So we are interested in finding which group of variables and individual variables have an important effect on the response. For example, $(x_{11}, ..., x_{1p_1}), ..., (x_{K1}, ..., x_{K_{pk}})$ may represent different biological roles for grouped genes and the response y may represent a certain disease. We want to find the association between genes and grouped genes with the disease.

Lasso (Tibshirani 1996) is an effective variable selection approach for “large p, small n” problem. We apply the Lasso to the “grouped” case. The lasso criterion penalizes the ℓ_1-norm of the regression coefficients:

$$max_{\beta_0, \beta_{kj}} - \frac{1}{2} \sum_{i=1}^{n} (y_i - \sum_{k=1}^{K} \sum_{j=1}^{p_k} \beta_{kj} x_{i,kj})^2 - \lambda \sum_{k=1}^{K} \sum_{j=1}^{p_k} |\beta_{kj}|,$$

where $\lambda \geq 0$ is a tuning parameter.

Lasso works well for the variables which can be treated individually. When the variables are grouped, the lasso does not work well. For example, suppose the k^{th} group is unimportant, lasso will only force individual coefficient in the k^{th} group to be zero. However, the whole group coefficients should be zero altogether because the k^{th} group is unimportant. Lasso tends to make selection based on the strength of individual variables rather than the strength of the group. It may result in selecting more groups than necessary. Therefore, for the grouped cases, it maybe desirable to shrink and select the members of a group together.

2.3.5 Elastic Net

Zou and Hastie (2003) proposed the elastic net which is also a penalized variable selection method. They found out that the elastic net often outperforms the lasso, while enjoying a similar sparsity of representation. In addition, the elastic net encourages a grouping effect, where strongly correlated predictors tend to be in or out of the model together. The elastic net is particularly useful when the number of predictors p is much bigger than the number of observations n.

2.4 Group Variable Selection Methods

2.4.1 Group Lasso

Yuan and Lin (2006) proposed the group lasso to address the grouping problem. They extended the lasso to group variable selection. Suppose that the p predictors are divided into K groups, with number of predictors p_k in group k. We use a matrix X_k to denote the predictors corresponding to the k^{th} group and coefficient vector β_k.

The group lasso minimizes the convex criterion

$$minimize \quad \|Y - \textstyle\sum_{k=1}^{K} X_k\beta_k\|_2^2 + \lambda \sum_{k=1}^{K} \sqrt{p_k} \parallel \beta_k \parallel_2$$

Yuan and Lin (2006) chose to penalize the ℓ_2 norm of the coefficients within each group, i.e. $\Sigma_{k=1}^{K} \parallel \beta_k \parallel_2$, where

$$\parallel \beta_k \parallel = \sqrt{\beta_{k1}^2 + ... + \beta_{kp_k}^2}$$

where $\lambda \geq 0$ is a tuning parameter and the term $\sqrt{p_k}$ represent the varying group sizes. The penalty function used is intermediate between the ℓ_1 penalty used in the Lasso and ℓ_2 penalty used in ridge regression. The $\|\cdot\|$ is the Euclidean norm (not squared). Since the Euclidean norm of a vector β_k is zero only if all of its components are zero, this procedure encourages sparsity at both the group and individual levels. That is, for some values of λ, certain tuning parameter λ can set the whole coefficient vector $\beta_k = 0$. So the kth group is removed from the fitted model. If the group sizes are all equals to 1, it reduces to the regular lasso.

This procedure was first proposed by Bakin (1999) as an extension of the Lasso for selecting groups of variables for which he also proposed a computational algorithm. Yuan and Lin generalized it. Generalizations include more general ℓ_2 norm $\parallel \eta \parallel_K = (\eta^T K \eta)^{1/2}$, as well as allowing overlapping groups of predictors (Zhao et al., 2008). There are also connections to methods for fitting sparse additive models (Lin and Zhang, 2006; Ravikumar et al., 2008).

2.4.2 Group Lasso, Group MCP, Group Bridge

Breheny and Huang (2009) considered the regression problems in which the covariates can be grouped. They are interested in selecting important groups as well as identifying important members of these groups. They define this as bi-level selection. They mentioned, the group lasso proposed by Yuan & Lin (2006) has its penalty function as comprised of ℓ_2 norms of the groups. This has the effect of encouraging sparsity at the group level while applying ridge regression-like shrinkage within a group. The group lasso performs variable selection at group level but not at individual level. However, the group bridge proposed by Huang et al. (2007) applies a bridge penalty to the ℓ_1 norm of the groups, performing variable selection by encouraging sparse solutions at both the group and individual level.

Group lasso and group bridge also have their own shortcomings. Group lasso is incapable of variable selection at the individual level and heavily shrinks large coefficients. Group bridge suffers from some practical difficulties because it is not everywhere differentiable. Furthermore, both methods make inflexible grouping assumptions that can cause the methods to suffer when groups are misspecified or sparsely represented.

The algorithms that have been proposed to fit models with grouped penalties are either inefficient for models with large number of predictors or limited to linear regression models, models with orthogonal group members. Therefore, Breheny and Huang (2009) felt there is a need to develop tools that perform bi-level group variable selection. In their paper, they proposed a new framework to better understand the behavior of group penalties. They also proposed a new method, group MCP which perform variable selection at both group and individual level and develop a fast algorithm called "grpreg" to fit group lasso,group bridge and group MCP.

Framework for Group Penalized Methods

Suppose we have data $(x_1, y_1), ..., (x_i, y_i), ..., (x_n, y_n), i = 1, 2, ..., n$, where $x_i = (x_{i1}, ..., x_{ip})$ is a p-dimensional predictor and y_i is the response variable. x_i contain an unpenalized intercept and J groups x_{ij}, with K_j be the size of group j. Covariates that do not belong to a group are considered as a group of one. We want to find a sparse estimates of coefficients of β by a loss function L which quantifies the discrepancy between an observation y_i and a linear predictor $\eta_i = x_i'\beta = \beta_0 + \Sigma_{j=1}^{j} x_{ij}'\beta_j$, where β_j is the coefficients in the j^{th} group. The covariates are standardized to make $\Sigma_{i=1}^{n} x_{ijk} = 0$ and $\frac{1}{n}\Sigma_{i=1}^{n} x_{ijk}^2 = 1$ to make the penalty applied equally. The covarties are standardized without loss of generality during the model fitting process and are transformed back to the original scale after model fitting.

The effect of a penalty on the coefficients are subject to the penalty's gradient. Penalties have their forms as $\lambda\beta^\gamma$. The ridge regression has $\gamma = 2$, so its rate of penalization increases with increase of β. The ridge penalty applies little to no penalization near 0 and large penalization to large coefficients. The lasso has its $\gamma = 1$, so its rate of penalization is constant. If $\gamma = 1/2$, the rate of penalization is very high near 0 but diminishes as β grows larger.

The group lasso minimizes the following objective function and β is the solution to the function

$$Q(\beta) = \frac{1}{2n}\|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda\Sigma_{j=1}^{J}\sqrt{K_j}\|\beta_j\|, \tag{2.2}$$

where $\| \cdot \|$ is the ℓ_2 norm. The group bridge estimate minimizes

$$Q(\beta) = \frac{1}{2n}\|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda \Sigma_{j=1}^{J} K_j^{\gamma} \|\beta_j\|_1^{\gamma}, \tag{2.3}$$

where $\| \cdot \|_1$ is the ℓ_1 norm. Throughout their paper, the group bridge has $\gamma = 1/2$.

To better understand the effect of penalties, the grouped penalties can be considered having a form of an outer penalty f_o applied to a sum of inner penalties f_I. So the penalty applied to a group of covariates is

$$f_o(\sum_{k=1}^{K_j} f_I(|\beta_{jk}|)) \tag{2.4}$$

and the partial derivative with respect to the jk^{th} covariate is

$$f_o'(\sum_{k=1}^{K_j} f_I|\beta_{jk|})f_I'(|\beta_{jk}|). \tag{2.5}$$

Both group lasso and group bridge fit into this framework. The group lasso has an outer bridge and inner ridge penalty. The group bridge has an outer bridge and inner lasso penalty. From (2.4), we can understand that the group penalization apply a rate of penalization to covariate that consist of two terms: one carrying the information about the group, and the other carrying the information about the individual. Variables can enter the model either by having a strong individual effect to the response or in a group with strong collective effects. Conversely, a variable with a strong individual effect can be excluded from a model through its association with a preponderance of weak group members. This framework is helpful in understanding the gradient and effect of a group penalty. However, casually combining penalties will lead to meaningless group penalization. For example, using the lasso as both inner and outer penalty is actually conventional lasso and make no use of grouping structure. Properties may emerge from combinations of these two penalties. The group lasso has a convex penalty even though the outer bridge penalty is nonconvex.

Zhang (2007) proposed a nonconvex penalty called MCP. The MCP penalty and its derivative are defined on $[0, \infty)$ by

$$f_{\lambda,a}(\theta) = \begin{cases} \lambda\theta - \frac{\theta^2}{2a}, & if \quad \theta \leq a\lambda; \\ 1/2a\lambda^2, & if \quad \theta > a\lambda. \end{cases}$$

$$f'_{\lambda,a}(\theta) = \begin{cases} \lambda - \frac{\theta}{a}, & if \quad \theta \leq a\lambda; \\ 0, & if \quad \theta > a\lambda. \end{cases}$$

for $\lambda \geq 0$. The MCP penalty applies the same rate of penalization as the lasso when $\theta \leq a\lambda$, and it continues to reduce the penalization to 0 when $\theta > a\lambda$. MCP is motivated by and rather similar to SCAD. The connections between MCP and SCAD are investigated by Zhang (2007). Both penalties perform variable selection by eliminating unimportant variables from the model while leaving the important variables in the model. This is the so-called "oracle" property which fitting an unpenalized model in which the truly nonzero variables are known in advance.

For the MCP penalty, the λ is the regularization parameter that controls the amount of penalization and a is a turning parameter that affects the range over which the penalty is applied. When a is small, the region in which MCP is not constant is small; when a is large, MCP penalty has a broader influence. So small values of a are best at retaining the unbiasedness of the MCP penalty for large coefficients, but they may create nonconvexity objective function that is difficult to optimize and in turn generate discontinuous solution with respect to λ. So it is better to choose an a that is large enough to avoid problems but not too large. For linear regression models, when the response and covariates are standardized to have standard deviation 1, they recommend using $a = 3$. As practical advice, they recommend always standardizing the variables and using $a = 3$.

The group MCP estimate minimizes

$$Q(\beta) = \frac{1}{2n}\|y - X\beta\|^2 + \Sigma_{j=1}^{J} f_{\lambda,b}(\Sigma_{k=1}^{K_j} f_{\lambda,a} \mid \beta_{jk} \mid),$$

where b, the tuning parameter of the outer penalty, is set to $K_j a\lambda/2$ to ensure that the group level penalty reaches its maximum if and only if each of its components are at their maximum. This means that the derivative of the outer penalty reaches 0 if and only if $|\beta_{jk}| \geq a\lambda$, for $k =\in 1, ..., K_j$.

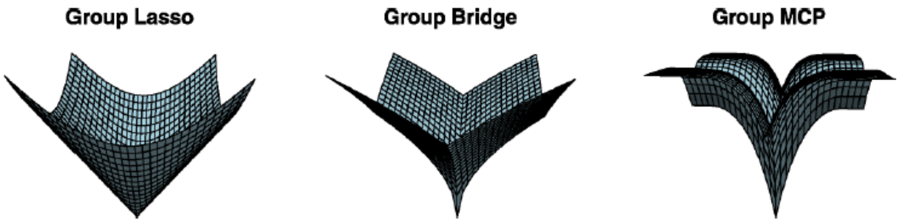

Figure 2.4: Penalties applied to a two-covariate group by the group lasso, group bridge, and group MCP methods

[Source: adapted from "Penalized methods for bi-level variable selection", by P. Breheny and J. Huang, 2009, Statistics And Its Interface, 2(1), p.369-380.]

Figure 2.4 plots the penalties applied to a two-covariate group by the group lasso, group bridge, and group MCP methods. Note that where the penalty comes to a point or edge, there is a possibility that the solution will take on a sparse value; all penalties come to a point at 0, encouraging group-level sparsity, but only group bridge and group MCP allow for bi-level selection. From Figure 2.4, the group MCP penalty is capped at both the individual covariate and group levels, while the group lasso and group bridge penalties are not. This illustrates the two rationales of group MCP: (1) to avoid over shrinkage by allowing covariates to grow large. (2) to allow groups to remain sparse internally. Group bridge allows the presence of a single large predictor to continually lower the entry threshold of the other variables in its group. This property, whereby a single strong predictor drags others into the model, prevents group bridge from achieving consistency for the selection of individual variables. Group MCP, on the other hand, limits the amount of signal that a single predictor can contribute towards the reduction of the penalty applied to the other member of the groups.

Local Coordinate Descent for Group MCP, Group Bridge, Group Lasso

Coordinate descent algorithms optimize a target function with respect to a single parameter at a time, iteratively cycling through all parameters until convergence is reached. The power of coordinate descent algorithms for optimizing penalized regression problems been gained much attention recently.

The procedure of the local group coordinate descent (LCD) algorithm is as follows: Let $\tilde{\beta}$ represents the current estimate of β,

1. Choose an initial estimate $\tilde{\beta} = \beta^{(0)}$
2. Approximate loss function, if necessary
3. Update covariates:
    (a) Update $\tilde{\beta}_0$

(b) For $j \in 1, ..., J$, update $\tilde{\beta}_j$

4. Repeat steps 2 and 3 until convergence

First, Breheny and Huang (2009) consider the updating of the intercept in step 3a. The partial residual for updating $\tilde{\beta}_0$ is $\tilde{y} = y - X_{-0}\tilde{\beta}_{-0}$, where the -0 subscript refers to what remains of X or $\tilde{\beta}$ after the 0^{th} column or element has been removed, respectively. The updated value of $\tilde{\beta}_0$ is therefore the simple linear regression solution

$$\tilde{\beta}_0 \longleftarrow \tfrac{x_0'\tilde{y}}{x_0'x_0} = \tfrac{1}{n}x_0'\tilde{y}.$$

They also use an equivalent approach with residuals for updating $\tilde{\beta}_0$ which is more efficient. They update $\tilde{\beta}_0$ by taking advantage of the current residuals $\tilde{r} = y - x\tilde{\beta}$. Since $\tilde{y} = \tilde{r} + x_0\tilde{\beta}_0$, the update becomes

$$\tilde{\beta}_0 \longleftarrow \frac{1}{n}x_0'\tilde{r} + \tilde{\beta}_0 \tag{2.6}$$

Updating $\tilde{\beta}_0$ in this way only need $2n$ operations: n operations to calculate $x_0'\tilde{r}$ and n operations to update $\tilde{r}$. In contrast, obtaining $\tilde{y}$ requires $n \times (p-1)$ operations. Meanwhile, for iteratively reweighed optimization, the updating step is

$$\tilde{\beta}_0 x_0' W\tilde{r}/x_0'Wx_0 + \tilde{\beta}_0, \tag{2.7}$$

requiring $3n$ operations. Updating $\tilde{\beta}_j$ in step 3b depends on the type of penalties. In the next, the updating steps for different group penalties, group bridge, group MCP and group lasso are being discussed.

Group MCP

Breheny and Huang (2009) begin by looking at the solutions of the Lasso. When the penalty being applied to a single parameter is $\lambda|\beta|$, the solution to the Lasso (Tibshirani, 1996) is

$$\beta = \tfrac{S(1/nx'y,\lambda)}{1/nx'x} = S(1/nx'y,\lambda),$$

where $S(z,c)$ is the soft-thresholding operator (Donoho and Johnstone,1994) defined for positive c by

$$S(z,c) = \begin{cases} z-c, & if \quad z > c; \\ 0, & if \quad |z| \leq c; \\ z+c, & if \quad z < -c. \end{cases}$$

Group MCP does not have a similarly convenient form for updating individual parameters. However, by taking the first order Taylor series approximation about $\tilde{\beta}_j$, the penalty as a function of β_{jk} is approximately proportional to $\tilde{\lambda}_{jk}|\beta_{jk}|$, where

$$\tilde{\lambda}_{jk} = f'^{\lambda,b}(\sum_{m=1}^{K_j} f_{\lambda,a}(|\tilde{\beta}_{jm}|))f'_{\lambda,a}(|\tilde{\beta}_{jk}|) \tag{2.8}$$

and f, f' were defined in group MCP penalty previously. Thus, in the local region where the penalty is well-approximated by a linear function, step 3b consists of simple updating steps based on the soft-thresholding cutoff $\tilde{\lambda}_{jk}$ for $k \in 1, ..., K_j$.

$$\tilde{\beta}_{jk} \longleftarrow S(1/nx'_{jk}\tilde{r} + \tilde{\beta}_{jk}, \tilde{\lambda}_{jk}) \tag{2.9}$$

or with weights,

$$\tilde{\beta}_{jk} \longleftarrow \frac{S(1/nx'^{jk}W\tilde{r} + 1/nx'_{jk}Wx_{jk}\tilde{\beta}_{jk}, \tilde{\lambda}_{jk})}{1/nx'_{jk}Wx_{jk}} \tag{2.10}$$

Group Bridge

The local coordinate descent algorithm for group bridge is similar to that for group MCP, but with

$$\tilde{\lambda}_{jk} = \lambda_{\gamma} K_j^{\gamma} \parallel \tilde{\beta}_j \parallel^{\gamma-1} \tag{2.11}$$

The difficulty in group bridge is that since the bridge penalty is not everywhere differentiable, $\tilde{\lambda}_{jk}$ is undefined at $\tilde{\beta}_j = 0$ for $y < 1$. 0 create a fundamental problem with the penalty itself. For any positive value of λ, 0 is a local minimum of the group bridge penalty and therefore complicates optimization.

To address this problem, they choose to begin with an initial value away from 0. If $\tilde{\beta}_j$ reaches 0 at any point during the iteration, they restrain $\tilde{\beta}_j$ at 0 thereafter. This causes the potential drawback of dropping groups that actually is nonzero when the solution converges. There are other approaches to address this problem such as adding a small constant to $\tilde{\beta}_j$ in (2.11) However, it would prevent the algorithm from taking advantage of sparsity and greatly reduce computational efficiency for large, sparse problems.

Group Lasso

Updating is more complicated in the group lasso because of its properties that grouped variables go to 0 all at once or not at all. Breheny and Huang (2009) choose to update $\tilde{\beta}_j$ at step 3b in two steps: (1) check whether $\tilde{\beta}_j = 0$; (2) if $\tilde{\beta} \neq 0$, update $\tilde{\beta}_{jk}$ for $k \in 1, ..., K_j$. Step (1) is performed under the condition that $\tilde{\beta}_j \neq 0$ if and only if

$$1/n \parallel X_j' \tilde{r} + X_j' X_j \tilde{\beta}_j \parallel > \sqrt{K_j} \lambda \tag{2.12}$$

The conditions above are the Karush-Kuhn-Tucker conditions for this problem, and were point out by Yuan and Lin (2006) first. The reason to do these is that if β_j cannot move in any direction away from 0 without increasing the penalty more than the movement improves the fit, then 0 is a local minimum. Since the group lasso penalty is convex, 0 is also the unique global minimum.

If this condition does not hold, then they set $\tilde{\beta}_j = 0$ and move on. Otherwise, they make a local approximation to the penalty and update the members of group j. However, instead of approximating the penalty as a function of $|\beta_{jk}|$, they can obtain a better approximation by considering the penalty as a function of β_{jk}^2. Therefore, the penalty applied to β_{jk} can be approximated by $\tilde{\lambda}_{jk}\beta_{jk}^2/2$, where

$$\tilde{\lambda}_{jk} = \frac{\lambda\sqrt{K_j}}{\|\tilde{\beta}_j\|} \tag{2.13}$$

This approach yields a shrinkage updating step instead of a soft-thresholding step

$$\tilde{\beta}_{jk} \longleftarrow \frac{1/nx'_{jk}\tilde{r} + \tilde{\beta}_{jk}}{1 + \tilde{\lambda}_{jk}} \tag{2.14}$$

or for weighted optimization,

$$\tilde{\beta}_{jk} \longleftarrow \frac{1/nx'^{jk}W\tilde{r} + 1/nx'_{jk}Wx_{jk}\tilde{\beta}_{jk}}{1/nx'_{jk}Wx_{jk} + \tilde{\lambda}_{jk}} \tag{2.15}$$

Note that, like (2.11), (2.13) is undefined at 0. However, this is merely a minor algorithmic inconvenience in group lasso. The penalty is differentiable but with its partial derivatives having a different form at 0. This issue can be avoided by adding a small positive quantity δ to the denominator in (2.10).

Meier et al. (2008) have also proposed a coordinate descent algorithm for fitting group lasso models. However, Meier et al. (2008) consider only the special case in which groups are orthonormal.

2.4.3 The Group Lasso for Logistic Regression

Suppose that we have data (x_i, y_i), $i = 1, ..., n$, of a p-dimensional vector x_i of G predictors. $y_i \in (0, 1)$ is the binary outcome. Both categorical and continuous predictors are allowed. Let df_p be the degrees of freedom of the gth predictor. For example, the main effect of a factor with four levels has $df = 3$ and a continuous predictor has $df = 1$ only.

Linear logistic regression models the probability $p_i = P(Y = 1|x_i)$ by

$$\eta(x_i) = log(\frac{p_i}{1-p_i}) = \beta_0 + \sum_{g=1}^{G} x_{i,g}^T \beta_g \tag{2.16}$$

where β_0 is the intercept and β_g is the parameter vector corresponding to the g^{th} predictor. We denote the whole parameter vector by $\beta = (\beta_0, \beta_1^T, ..., \beta_G^T)^T$. The logistic group lasso estimator $\hat{\beta_\lambda}$ is given by the minimizer of the convex fucntion

$$S_\lambda(\beta) = -\ell(\beta) + \lambda \Sigma_{g=1}^{G} s(df_g) \|\beta_g\|_2 \tag{2.17}$$

where $\ell(\cdot)$ is the log-likelihood function, i.e.

$$\ell(\beta) = \Sigma_{i=1}^{n} y_i \eta(x_i) - log[1 + exp\{\eta(x_i)\}]. \tag{2.18}$$

The tuning parameter $\lambda \geq 0$ controls the amount of penalization. We do not penalize the intercept. The minimum in equation (2.17) is attained. The function $s(\cdot)$ is used to rescale the penalty with respect to the dimensionality of the parameter vector β_g. They use $s(df_g) = df_g^{1/2}$ to ensure that the penalty term is of the order of the number of parameters df_g. The same rescaling was used in Yuan and Lin (2006). The "groupwise" ℓ_2-norm in equation (2.17) is an intermediate between the lasso and the ridge penalty function.

Yuan and Lin (2006) proposed an algorithm to solve a system of non-linear equations which corresponds to a groupwise minimization of the penalized residual sum of squares. They did not give a numerical convergence. The algorithm that they proposed is a revision of block co-ordinate descent algorithm. In this group Lasso for logistic regression paper, Meier, Geer and Bühlmann (2008) also used the block co-ordinate descent algorithm to solve more complicated logistic regression models.

2.5 Hierarchical Lasso

The existing successful group variable selection methods such as Antoniadis and Fan (2001), Yuan and Lin (2006) and Zhao, Rocha and Yu (2009) have the limitation of selecting variables in an "all-in-all-out" fashion, i.e. when one variable in a group is selected, all other variables in the same group are also selected. Zhou and Zhu (2010) realized this problem and proposed an extension of group lasso for group variable selection, which they call it hierarchical lasso (HLasso). This method not only removes unimportant groups, but also selects variables within a group. They also showed that the new method has the potential to achieve the theoretical "oracle" property as in Fan and Li (2001) and Fan and Peng (2004).

The original lasso (Tibshirani, 1996) penalizes the ℓ_1-norm of the regression coefficients:

$$max_{\beta_0,\beta_{kj}} - 1/2\sum_{i=1}^{n}(y_i - \sum_{k=1}^{K}\sum_{j=1}^{p_k}\beta_{kj}x_{i,kj})^2 - \lambda\sum_{k=1}^{K}\sum_{j=1}^{p_k}|\beta_{kj}|$$

Due to the singularity at $\beta_{kj} = 0$, the ℓ_1-norm penalty can shrink some of the fitted coefficients to be zero when the penalty is large enough. The Lasso works well for the variables that can be treated individually. When the variables are grouped, the lasso does not work well. For example, suppose the k^{th} group is unimportant, lasso will only force individual coefficient in the k^{th} group to be zero. However, the whole group coefficients should be zero altogether because the k^{th} group is unimportant. The Lasso tends to make selection based on the strength of individual variables rather than the strength of the group. Therefore, for the grouped cases, it maybe desirable to shrink and select the members of a group together.

Antoniadis and Fan (2001), Yuan and Lin (2006) and Zhao, Rocha and Yu (2009) have proposed methods to solve the group variable selection problem. Antoniadis and Fan (2001) proposed to use a blockwise additive penalty in the setting of wavelet approximations. To increase the estimation precision, empirical wavelet coefficients were thresholded or shrunken in blocks (or groups) rather than individually.

Yuan and Lin (2006) and Zhao, Rocha and Yu (2009) extended the Lasso for group variable selection. Yuan and Lin (2006) chose to penalize the ℓ_2-norm of the coefficients within each group, i.e., $\sum_{k=1}^{K}||\beta_k||_2$, where

$$||\beta_k||_2 = \sqrt{\beta_{k1}^2 + ... + \beta_{kp_k}^2}$$

Unlike Yuan and Lin (2006) using the ℓ_2-norm penalty, Zhao, Rocha and Yu (2009) used the ℓ_∞-norm penalty, i.e., $\sum_{k=1}^{K}||\beta_k||_\infty$, where

$$||\beta_k||_\infty = max(|\beta_{k1}|, ..., |\beta_{kp_k}|)$$

Because of the singularity of $||\beta_k||_2$ at $\beta_k = 0$, appropriate tuning parameter λ can make the whole coefficient vector $\beta_k = 0$, hence the k^{th} group is unimportant and removed from the fitted model.

Similar to the ℓ_2-norm, the $||\beta_k||_\infty$ is also singular at $\beta_k = 0$; hence when λ is appropriately tuned, the ℓ_∞-norm can also effectively remove unimportant groups.

Both the ℓ_2-norm penalty and the ℓ_∞-norm penalty have the problem of selecting variables in an "all-in-all-out" fashion, i.e., when one variable in a group is selected, all other variables in the same group are also selected. The reason is that both $||\beta_k||_2$ and $||\beta_\infty||$ are singular only when the whole vector $\beta_k = 0$. Once a component of β_k is non-zero, the two norm functions are no longer singular. For the ℓ_2-norm, it is the ridge penalty that is under the square root. Since the ridge penalty can not do variable selection (as in ridge regression), once the ℓ_2-norm is non-zero (or the corresponding group is selected), all components will be non-zero. For the ℓ_∞-norm, if the "$max(\cdot)$"is non-zero, there is no increase in the penalty for letting all the individual components move away from zero. Hence if one variable in a group is selected, all other variables are also automatically selected.

In many practical problems, however, we may want to keep the flexibility of selecting variables within a group. For example, genes in the same biological pathway form a group and the group maybe related to a certain biological process. However, not all genes in the group maybe related to biological process, we want to identify the genes that are important to the biological process.

To solve this problem, Zhou and Zhu (2009) proposed the hierarchical lasso which they reparameterize β_{kj} as

$$\beta_{kj} = d_k \alpha_{kj}, k = 1, ..., K; j = 1, ..., p_k,$$

where $d_k \geq 0$ (for identifiability reasons). In this case, $\beta_{kj}, j = 1, ..., p_k$, all belong to the k^{th} group. The β_{kj} are comprised of d_k and α_{kj}. d_k is at the first level of the hierarchy, controlling $\beta_{kj}, j = 1, ..., p_k$, as a group; α_{kj}'s are at the second level of the hierarchy, reflecting differences within the k^{th} group.
So they consider the penalized least squares criterion

$$max - \tfrac{1}{2}\textstyle\sum_{i=1}^{n}(y_i - \sum_{k=1}^{K} d_k \sum_{j=1}^{p_k} \alpha_{kj} x_{i,kj})^2 - \lambda_1 \sum_{k=1}^{K} d_k - \lambda_2 \sum_{k=1}^{K} \sum_{j=1}^{p_k} |\alpha_{kj}|$$

subject to $d_k \geq 0, k = 1, ..., K$,
where $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$ are tuning parameters. λ_1 controls the estimates at the group level, and it can effectively remove unimportant groups: if d_k is shrunken to zero, all β_{kj} in the kth group will be set to zero. λ_2 controls the estimates at the variable-specific level: if d_k is not equal to zero, some of the α_{kj} and $\beta_{kj}, j = 1, ..., p_k$, still have the chances of being zero. So the hierarchical penalty keeps the flexibility of the ℓ_1 norm penalty.

From the above, the hierarchical penalty seems complicated to implement in practice. However, the λ_1 and λ_2 can be combined into one penalty. Let $\lambda = \lambda_1 * \lambda_2$, the above function showed by a Lemma to be is equivalent to, but maybe with different d_k and α_{kj}:

$$max_{d_k,\alpha_{kj}} - \tfrac{1}{2}\textstyle\sum_{i=1}^{n}(y_i - \sum_{k=1}^{K} d_k \sum_{j=1}^{p_k} \alpha_{kj} x_{i,kj})^2 - \sum_{k=1}^{K} d_k - \lambda \sum_{k=1}^{K} \sum_{j=1}^{p_k} |\alpha_{kj}|$$

subject to $d_k \geq 0, k = 1, ..., K$.

2.5.1 Adaptive Hierarchical Lasso

Zhou and Zhu (2009) also applied the adaptive idea which has been used in Breiman (1995), Wang, Li and Tsai (2006), Zhang and Lu (2007), and Zou (2006) as an extension to hierarchical lasso method. This function penalizes different coefficients differently:

$$max_{\beta_n,kj} - \frac{1}{2}\sum_{i=1}^{n}(y_{ni} - \sum_{k=1}^{K_n}\sum_{j=1}^{p_k} x_{ni,kj}\beta_{n,kj})^2 -$$
$$n\lambda_n \sum_{k=1}^{K_n} \sqrt{\omega_{n,k1}|\beta_{n,k1}| + \omega_{n,k2}|\beta_{n,k2}| + ... + \omega_{n,kp_k}|\beta_{n,kp_{nk}}|}$$

where $\omega_{n,kj}$ are pre-specified weights. In this case, if the effect of a variable is strong, the corresponding weight is set to be small, hence the coefficient is slightly penalized. If the effect of a variable is weak, the corresponding weight is set to be large and hence the corresponding coefficient is heavily penalized. In practise, the weight is computed by ordinary least squares estimates or the ridge regression estimates, such as

$$\omega_{n,kj} = \frac{1}{|\hat{\beta}^{ols}_{n,kj}|^{\gamma}} \text{ or } \omega_{n,kj} = \frac{1}{|\hat{\beta}^{ridge}_{n,kj}|^{\gamma}}$$

where γ is a positive constant. They also showed that the adaptive hierarchical lasso enjoys an "Oracle" property. i.e. it performs as if the true sub-model is known in advance.

They conducted simulation studies to compare hierarchical lasso with the ℓ_2-norm group lasso and the ℓ_∞-norm group lasso. They also compared the adaptive hierarchical lasso with other "Oracle" property but not group selection methods: SCAD and the adaptive lasso.

They found out that all shrinkage methods are better than OLS. Therefore, the regularization is necessary for prediction accuracy. In terms of prediction accuracy, they found that when variables in a group follow the "all-in-all-out" pattern, where all the variables in a group are important or non-important to the response, the group lasso performs slightly better than the hierarchical lasso. However, when variables in a group do not follow the “all-in-all-out” pattern, the hierarchical lasso method performs slightly better than the group lasso. In terms of variable selection, they look at it by percentage of correctly identified important variables. The lasso, the group lasso, and the hierarchical lasso all perform similarly. However, the group lasso and the hierarchical lasso are more effective at removing unimportant variables than lasso.

They also assessed the performance of adaptive hierarchical lasso where they used the adaptive weights. They compared the adaptive hierarchical lasso with other Oracle properties but not group variable selection methods: SCAD and adaptive lasso. From the results, in the “all-in-all-out” case, the adaptive hierarchical lasso removes unimportant variables more effectively than SCAD and adaptive lasso; the adaptive hierarchical lasso outperforms SCAD and adaptive lasso significantly in terms of prediction accuracy. In the “not-all-in-all-out” case, the advantage of knowing the grouping structure information is reduced, but the adaptive hierarchical lasso still performs slightly better than SCAD and adaptive lasso, especially in terms of removing unimportant variables.

The hierarchical lasso method not only effectively removes unimportant groups, but also keeps the flexibility of selecting variables within a group. They also showed that the improved hierarchical lasso method enjoys an “Oracle property”. Numerical results indicate that their methods works well, especially when variables in a group are associated with the response in a “not-all-in-all-out” fashion.

Chapter 3

A Simulation Study

The purpose of this simulation study is to compare the variable selection methods: the Lasso, elastic net, group lasso, group bridge, group MCP. We tested these five methods in the logistic regression setting with binary outcome. We measured the performance of each method in terms of prediction accuracy and number of correctly identified important or non-important variables. The simulated data are generated from the logistic regression model:

$$logit(P_i) = log(\frac{P_i}{1-P_i}) = \alpha + \beta * \mathbf{X}$$

where $X_i's$ are the predictors generated from standard normal distribution with mean 0, Y_i's are the binomially distributed data where $Y_i \sim Binomial(n_i, P_i)$ for $i = 1...n$. The simulation examples are conducted based on following criterions: see Table 3.1

For the criterion of different sizes of effect, we considered odds ratio. For example, let X_i be a binary outcome with $X_i = 1$ indicating female, while $X_i = 0$ indicating male, we have:

$$log(odds_F) = log(\frac{P_{iF}}{1-P_{iF}}) = \beta_0 + \beta_1 * 1$$

Table 3.1: Simulation criterions for different examples

simulation criterions	levels	numeric values
effect	large	$log(1.8)$
	medium	$log(1.6)$
	low	$log(1.2)$
grouping structure	all-in-all-out	
	not-all-in-all-out	
large p, small n	p is greater than n	p=40 n=20
	p is less than n	p=40 n=100
correlation	high	0.8
	medium	0.6
	low	0.2

$$log(odds_M) = log(\frac{P_{iM}}{1-P_{iM}}) = \beta_0 + \beta_1 * 0$$

$$\text{Thus, } log(\frac{odds_F}{odds_M}) = \beta_1$$

$$odds \quad ratio = \frac{odds_F}{odds_M} = exp(\beta_1)$$

$$\beta_1 = log(odds \quad ratio)$$

In the logistic regression setting, odds ratio between 1.1 to 1.5 is considered to be small effect, 1.6 to 1.7 is considered to be medium effect, 1.8 to 2 is considered to be large effect. So we used $log(1.2), log(1.6), log(1.8)$ for small, medium and large effect in our case.

Four examples are presented here. The first was used in the original lasso paper (Tibshirani, 1996), to compare the prediction performance of the lasso and ridge regression systematically. The second and fourth examples were adopted from a paper about group variable selection methods (Xie and Zeng, 2008). The third example was adopted from a paper about elastic net (Zou and Hastie, 2003). The last three examples created a grouped variable situation which we intend to compare the original lasso and group methods. We also created several extended examples for comparison.

For each example, we simulated 100 data sets. Each data set consists of a training set and a test set. The training set was used to select tuning parameter. Models then fit to the test set and the median squared error is computed on the test set. We also recorded how frequently the important variables were selected and how frequently the unimportant variables were removed. The results are summarized in tables below.

The first criteria to test different method is the prediction. We used three different measures to quantify the error of the prediction:

(1) The amount of the median mean squared error which is defined as:

$$MSE = (\hat{\beta} - \beta^{(0)})^T(\hat{\beta} - \beta^{(0)})$$

(2) The model errors is:

$$ME = E(E(Y|x) - \hat{\mu})^2,$$

where $E(Y|x) = p(x) = \frac{exp(x\beta)}{1+expx\beta}$ and $\hat{\mu}(x) = \hat{p}(x) = \frac{exp(x\hat{\beta})}{1+expx\hat{\beta}}$.

(3) The classification or counting error:

$$CE = 1 \quad \text{if } Y_{new} = 1 \text{ and } \hat{p} < 1/2 \text{ or } Y_{new} = 0 \text{ and } \hat{p} > 1/2,$$
$$CE = 1/2 \quad \text{if } \hat{p} = 1/2$$
$$CE = 0 \quad \text{otherwise.}$$

The second criteria of testing a method is the percentage of correctly removed unimportant variables and percentage of correctly identified important variables. The four examples are given by:

- Example 1. We simulated 100 data sets consisting of 100 and 200 observations in the training and test sets respectively and 8 predictors. We let the true parameter $\beta = (log(1.8), log(1.2), 0, 0, log(1.6), 0, 0, 0)$. The pairwise correlation between x_i and x_j was set to be $corr(i,j) = 0.5^{(|i-j|)}$. So the covariates have a first-order autoregressive correlation.

- Example 2. We simulated 100 data sets consisting of 100 and 400 observations in the training and test sets respectively and 40 predictors. We set

$$\beta = (\underbrace{log(1.6), ..., log(1.6)}_{15}, \underbrace{log(1.2), ..., log(1.2)}_{5}, \underbrace{0, ..., 0}_{20})$$

  This example created one group with highly correlated 15 variables. The within group correlation is set to be high as 0.8. The between group correlations of variables are 0. The 5 variables with coefficient 1.5 are independent to others and have a small effect to the response. The rest of 20 variables are independent to others and have no effect on the response.

- Example 3.1 We simulated 100 data sets consisting of 100 and 400 observations in the training and test sets and 40 predictors. We chose

$$\beta = (\underbrace{log(1.2), ..., log(1.2)}_{5}, \underbrace{log(1.2), ..., log(1.2)}_{5}, \underbrace{log(1.2), ..., log(1.2)}_{5}, \underbrace{0, ..., 0}_{25})$$

  This example created three groups with highly correlated 5 variables in each group. The effects of grouped variables are set to small in each group. The within group correlation is set to be high as 0.8. The between group correlations of variables are 0. The variables in each group are equally important to the response. There are also 25 variables with non-important effects to the response.

- Example 3.2 Same as Example 3.1 except grouped variables have medium effects. We chose

$$\beta = (\underbrace{log(1.6), ..., log(1.6)}_{5}, \underbrace{log(1.6), ..., log(1.6)}_{5}, \underbrace{log(1.6), ..., log(1.6)}_{5}, \underbrace{0, ..., 0}_{25})$$

- Example 3.3 Same as Example 3.1 except grouped variables have large effects. We chose

$$\beta = (\underbrace{log(1.8), ..., log(1.8)}_{5}, \underbrace{log(1.8), ..., log(1.8)}_{5}, \underbrace{log(1.8), ..., log(1.8)}_{5}, \underbrace{0, ..., 0}_{25})$$

- Example 3.4 Same as Example 3.2 except grouped correlation is set to be low as 0.2. We chose

$$\beta = (\underbrace{log(1.6), ..., log(1.6)}_{5}, \underbrace{log(1.6), ..., log(1.6)}_{5}, \underbrace{log(1.6), ..., log(1.6)}_{5}, \underbrace{0, ..., 0}_{25})$$

- Example 3.5 Same as Example 3.2 except grouped correlation is set to be medium as 0.4. We chose

$$\beta = (\underbrace{log(1.6), ..., log(1.6)}_{5}, \underbrace{log(1.6), ..., log(1.6)}_{5}, \underbrace{log(1.6), ..., log(1.6)}_{5}, \underbrace{0, ..., 0}_{25})$$

- Example 3.6 Same as Example 3.2 except the number of observations is set to 20 to create a case that $p > n$. We chose

$$\beta = (\underbrace{log(1.6), ..., log(1.6)}_{5}, \underbrace{log(1.6), ..., log(1.6)}_{5}, \underbrace{log(1.6), ..., log(1.6)}_{5}, \underbrace{0, ..., 0}_{25})$$

- Example 4.1 We simulated 100 data sets consisting of 100 and 200 observations in the training and test sets and 40 predictors. We set

$$\beta =$$
$$(\underbrace{log(1.2), ..., log(1.2), 0, 0}_{5}, \underbrace{log(1.2), ..., log(1.2), 0, 0}_{5}, \underbrace{log(1.2), ..., log(1.2), 0, 0}_{5}, \underbrace{0, ..., 0}_{25})$$

This example also created three groups with highly correlated variables. The effects of grouped variables are set to small in each group. The within group correlation is set to be high as 0.8. However, in each group there are three important variables and two zero-effect variables. The rest of 25 variables are independent zero-effect variables.

- Example 4.2. Same as Example 4.1 except grouped variables have medium effects. We set

$$\beta =$$

$$(\underbrace{log(1.6), ..., log(1.6), 0, 0,}_{5}\underbrace{log(1.6), ..., log(1.6), 0, 0,}_{5}\underbrace{log(1.6), ..., log(1.6), 0, 0,}_{5}\underbrace{0, ..., 0}_{25})$$

- Example 4.3. Same as Example 4.1 except grouped variables have large effects. We set

$$\beta =$$

$$(\underbrace{log(1.8), ..., log(1.8), 0, 0,}_{5}\underbrace{log(1.8), ..., log(1.8), 0, 0,}_{5}\underbrace{log(1.8), ..., log(1.8), 0, 0,}_{5}\underbrace{0, ..., 0}_{25})$$

- Example 4.4. Same as Example 4.1 except that there are only two variables having large effects and the rest having zero effect. We set

$$\beta =$$

$$(\underbrace{log(1.8), ..., log(1.8), 0, 0, 0,}_{5}\underbrace{log(1.8), ..., log(1.8), 0, 0, 0,}_{5}\underbrace{log(1.8), ..., log(1.8), 0, 0, 0,}_{5}\underbrace{0, ..., 0}_{25})$$

Table 3.2 shows the median (mean) squared-error from 100 simulations. We notice that the elastic net has smaller MSE than the lasso in all examples. So the elastic net is more accurate in estimating coefficients than the lasso under collinearity. The group variable selection methods: group lasso and group MCP perform significantly better than both the lasso and the elastic net in all examples, although the group MCP does not work well for the "large p, small n" example. The group bridge has very large MSE in some examples (example 2, 3, 4), so it is not a very good method. The group lasso is even more accurate than group MCP and thus is the most accurate method in estimating coefficients for grouped variable selection. Its performance is very stable across all examples. The reductions of group lasso compared to lasso in MSE in examples 1, 2.1, 3.1, and 4.1 are 62.82%, 58.32%, 67.78%, 24.12%. Table 3.4 shows the model error (ME) from 100 simulations. The group lasso still perform best and has the smallest ME in most examples. Table 3.6 shows the misclassification error (MCE). The group lasso has smallest MCE than other methods.

Table 3.8 shows the number of correctly identified important variables and number of correctly removed non-important variables. Lasso produces very sparse model but does not work for collinearity. Elastic net improves the Lasso when predictors are correlated. But elastic net misses the five important effects with the small coefficients $log(1.2)$ in example 2. In example 2, group lasso improves elastic net in terms of number of correctly identified important variables.

Example 3 is a case in which all the variables in a group are important. This shows an "all-in-all-out" fashion, i.e., when one variable in a group is selected, all other variables in the same group should also be selected. The group lasso selected all of important variables, while lasso, group MCP, group Bridge, all missed out some of the variables in a group. Therefore, we can see that the group lasso forces complete grouping, group MCP encourages grouping to a rather slight extent, and group bridge is somewhere in between. In example 3, we compared how the small, medium and large effect affect the group variable selection. In example 3.3, where we have five collective small effects within a group, the group lasso successful selects all of the small effects while lasso only selects two, elastic net, group bridge and group MCP select three. Therefore, the group lasso is the best method for complete grouping. It works well for the collective small effects within a group contributing to a large effect.

Example 4 creates a sparse situation where there are 5 variables in a group and 3 of them are important effects. This shows an "not-all-in-all-out" fashion. The group lasso selects all of the variables from each group even some of them are zero effects. For group MCP, approximately 2 variables are selected per group while group bridge selects 4 variables per group. As well as in example 4.4, although there are 5 variables in a group and 2 of them are large effects with $log(2)$, the group lasso selects all 5 variables. It selects variables based on group level but not on individual level. The group MCP selects the 3 groups and also identifies the 3 important variables of the group.

Since group MCP makes rather cautious assumptions about grouping, the method works well if there are larger number of rather sparse groups. It works well when the underlying model has less grouping. However, when the important effects are tightly clustered into groups, the group MCP tends to select too many groups and does not use the grouping information sufficiently. Group lasso, which is opposite to group MCP, tends to over shrink individual coefficients when group are sparsely populated.

Therefore, when applying the group penalization method, the data structure need to be considered. If one expects that the proportion of nonzero group members to be greater than one-half, use group lasso; otherwise use group MCP. If one expects this proportion to be close to one-half, one may wish to use group bridge.

We also tested the methods with different grouped correlation and sample size. The results are shown in the subsequent tables. From example 3.4 and 3.5 where correlations are set small and medium, we found out that there is no big difference for different methods in terms of prediction accuracy and number of variables correctly selected. However, if we set the sample size to small as in example 3.6. The prediction accuracy significantly reduced for each method. In terms of number of correctly selected variables, we can see that the collinearity issue for lasso is more obvious. The group lasso only select 1 important variable from group while group MCP select 3 and group lasso select 5. Therefore, the group lasso still outperformed the rest of the methods in "all-in-all-out", large p, small n situation.

Table 3.2: MSE: Median, mean, first and third quartiles of mean-squared error from 100 simulations

examples	methods	median	mean	Q1(25%)	Q3(75%)
1.1	lasso	0.39	0.41	0.32	0.54
	elastic net	0.38	0.40	0.30	0.52
	group MCP	0.18	0.20	0.11	0.27
	group Bridge	0.12	0.19	0.07	0.27
	group Lasso	0.15	0.17	0.09	0.25
2.1	lasso	1.96	1.94	1.75	2.09
	elastic net	1.48	1.49	1.26	1.68
	group MCP	2.13	2.40	1.82	2.86
	group Bridge	12.82	28.85	8.18	25.32
	group Lasso	0.82	1.14	0.69	1.06
3.1	lasso	0.38	0.37	0.33	0.42
	elastic net	0.31	0.32	0.27	0.35
	group MCP	0.35	0.37	0.30	0.39
	group Bridge	0.40	0.69	0.32	0.72
	group Lasso	0.11	0.13	0.09	0.14
3.2	lasso	1.73	1.75	1.54	1.99
	elastic net	1.44	1.45	1.20	1.64
	group MCP	1.56	1.64	1.26	1.80
	group Bridge	3.99	4.38	2.66	5.37
	group Lasso	0.67	0.85	0.56	0.85
3.3	lasso	2.65	2.64	2.29	2.96
	elastic net	2.35	2.26	1.97	2.61
	group MCP	2.01	2.15	1.69	2.38
	group Bridge	5.99	6.59	3.90	7.44
	group Lasso	1.23	1.37	0.93	1.58
3.4	lasso	2.22	2.25	1.95	2.58
	elastic net	2.10	2.06	1.75	2.33
	group MCP	1.79	1.63	1.34	2.03
	group Bridge	1.33	1.44	1.08	1.72
	group Lasso	0.83	0.82	0.59	0.99

Table 3.3: MSE: Median, mean, first and third quartiles of mean-squared error from 100 simulations (Continued...)

examples	methods	median	mean	Q1(25%)	Q3(75%)
3.5	lasso	1.89	1.92	1.68	2.21
	elastic net	1.71	1.73	1.45	1.98
	group MCP	1.39	1.28	0.81	1.64
	group Bridge	1.64	1.83	1.27	2.24
	group Lasso	0.72	0.78	0.58	0.89
3.6	lasso	3.18	3.90	3.05	3.31
	elastic net	3.04	3.10	2.76	3.26
	group MCP	21.51	40.37	12.54	45.45
	group Bridge	7.48	9.85	5.38	11.58
	group Lasso	3.55	4.72	2.52	6.23
4.1	lasso	0.29	0.28	0.27	0.30
	elastic net	0.28	0.27	0.25	0.30
	group MCP	0.25	0.27	0.23	0.30
	group Bridge	0.31	0.40	0.30	0.37
	group Lasso	0.22	0.23	0.18	0.25
4.2	lasso	1.25	1.27	1.12	1.44
	elastic net	1.13	1.15	1.02	1.31
	group MCP	1.21	1.40	0.98	1.53
	group Bridge	6.38	8.70	1.45	10.81
	group Lasso	0.95	1.48	0.84	1.09
4.3	lasso	1.78	1.85	1.59	2.10
	elastic net	1.72	1.73	1.54	1.94
	group MCP	1.82	2.31	1.51	2.66
	group Bridge	13.00	17.53	7.34	19.11
	group Lasso	1.37	1.75	1.26	1.50
4.4	lasso	1.33	1.37	1.19	1.57
	elastic net	1.33	1.33	1.18	1.48
	group MCP	1.13	1.38	0.93	1.47
	group Bridge	1.73	6.18	1.03	7.49
	group Lasso	1.25	1.32	1.16	1.37

Table 3.4: ME: Median, mean, first and third quartiles of Model Error from 100 simulations

examples	methods	median	mean	Q1(25%)	Q3(75%)
1.1	lasso	4.17	4.44	3.17	6.00
	elastic net	4.01	4.21	2.83	5.83
	group MCP	1.44	1.77	0.92	2.34
	group Bridge	1.03	1.51	0.55	2.28
	group Lasso	1.11	1.34	0.66	1.88
2.1	lasso	7.80	7.82	5.07	10.32
	elastic net	6.26	6.55	4.26	8.29
	group MCP	3.59	3.80	3.09	4.27
	group Bridge	7.64	7.85	6.65	8.82
	group Lasso	2.47	2.74	1.90	3.42
3.1	lasso	13.45	15.20	10.59	19.44
	elastic net	12.75	13.57	10.07	15.78
	group MCP	7.22	7.21	5.61	8.86
	group Bridge	7.29	9.40	5.25	11.92
	group Lasso	3.39	3.74	2.26	4.50
3.2	lasso	11.82	12.48	9.08	15.88
	elastic net	10.03	10.67	7.16	13.19
	group MCP	4.24	5.08	3.22	5.61
	group Bridge	7.21	7.37	6.09	8.47
	group Lasso	3.42	3.71	2.46	4.60
3.3	lasso	10.89	11.69	8.65	14.37
	elastic net	10.04	9.93	7.31	12.42
	group MCP	3.68	4.19	3.22	4.63
	group Bridge	7.11	7.32	6.00	8.42
	group Lasso	3.75	4.09	2.68	5.21
3.4	lasso	23.16	24.19	18.72	29.08
	elastic net	20.57	20.92	15.72	24.65
	group MCP	16.47	15.24	11.07	19.97
	group Bridge	8.17	9.51	6.58	9.94
	group Lasso	5.68	5.93	4.03	7.09

Table 3.5: ME: Median, mean, first and third quartile of model error from 100 simulations (Continued...)

examples	methods	median	mean	Q1(25%)	Q3(75%)
3.5	lasso	17.92	18.75	14.28	22.75
	elastic net	14.64	15.88	11.35	18.83
	group MCP	10.71	10.07	4.93	13.80
	group Bridge	7.61	8.44	6.52	8.87
	group Lasso	4.32	4.81	3.47	5.77
3.6	lasso	2.11	2.06	1.32	2.70
	elastic net	1.99	1.96	1.35	2.59
	group MCP	1.48	1.60	1.06	2.13
	group Bridge	1.34	1.50	0.98	1.91
	group Lasso	1.92	1.24	0.79	1.58
4.1	lasso	6.23	5.83	5.06	6.89
	elastic net	5.99	5.73	4.91	6.89
	group MCP	3.45	3.55	2.53	4.46
	group Bridge	5.75	5.61	4.19	6.96
	group Lasso	2.61	2.87	1.92	3.86
4.2	lasso	6.75	6.79	4.95	8.67
	elastic net	5.99	6.10	4.47	7.78
	group MCP	3.54	3.74	2.86	4.63
	group Bridge	6.87	6.44	4.28	8.11
	group Lasso	2.71	3.08	2.06	3.58
4.3	lasso	6.40	6.70	5.06	8.23
	elastic net	5.79	6.09	4.56	7.24
	group MCP	3.53	3.80	2.95	4.62
	group Bridge	7.28	7.29	6.02	8.89
	group Lasso	2.58	2.87	2.02	3.22
4.4	lasso	7.01	7.24	4.96	8.82
	elastic net	6.39	6.67	5.09	7.81
	group MCP	3.68	3.90	2.86	4.73
	group Bridge	6.70	5.95	2.77	8.49
	group Lasso	3.11	3.38	2.66	3.96

Table 3.6: MCE: Median, mean, first and third quartile of misclassification error from 100 simulations

examples	methods	median	mean	Q1(25%)	Q3(75%)
1.1	lasso	0.1900	0.1856	0.1500	0.22125
	elastic net	0.1800	0.1787	0.1500	0.2125
	group MCP	0.1700	0.1713	0.1400	0.2000
	group Bridge	0.1700	0.1756	0.1475	0.2100
	group Lasso	0.1700	0.1686	0.1400	0.2000
2.1	lasso	0.0400	0.0403	0.0350	0.0475
	elastic net	0.0400	0.0397	0.0325	0.0475
	group MCP	0.0375	0.0368	0.0300	0.0425
	group Bridge	0.0300	0.0285	0.0200	0.0375
	group Lasso	0.0375	0.0374	0.0300	0.0450
3.1	lasso	0.1375	0.1430	0.1269	0.1575
	elastic net	0.1325	0.1374	0.1225	0.1475
	group MCP	0.1325	0.1327	0.1200	0.1425
	group Bridge	0.1325	0.1401	0.1200	0.1525
	group Lasso	0.1313	0.1300	0.1169	0.1425
3.2	lasso	0.0650	0.0670	0.0575	0.0731
	elastic net	0.0650	0.0660	0.0575	0.0725
	group MCP	0.0600	0.0619	0.0525	0.0700
	group Bridge	0.0550	0.0561	0.0450	0.0656
	group Lasso	0.0638	0.6333	0.0525	0.0725
3.3	lasso	0.0550	0.0541	0.0469	0.0625
	elastic net	0.0550	0.0532	0.0469	0.0600
	group MCP	0.0500	0.0489	0.0400	0.0575
	group Bridge	0.0438	0.0442	0.0350	0.0550
	group Lasso	0.0500	0.0507	0.0450	0.0575
3.4	lasso	0.1125	0.1144	0.0950	0.1275
	elastic net	0.1025	0.1031	0.0900	0.1150
	group MCP	0.1013	0.1002	0.0875	0.1150
	group Bridge	0.0875	0.0891	0.0750	0.1025
	group Lasso	0.0900	0.0890	0.0794	0.1000

Table 3.7: MCE: Median, mean, first and third quartile of misclassification error from 100 simulations (Continued...)

examples	methods	median	mean	Q1(25%)	Q3(75%)
3.5	lasso	0.0875	0.8898	0.0769	0.1000
	elastic net	0.0838	0.0832	0.0725	0.0950
	group MCP	0.0825	0.0815	0.0694	0.0925
	group Bridge	0.0750	0.0751	0.0600	0.0875
	group Lasso	0.0775	0.0771	0.0650	0.0881
3.6	lasso	0.0500	0.0770	0.0000	0.1500
	elastic net	0.0500	0.0595	0.0000	0.1000
	group MCP	0.0000	0.0085	0.0000	0.0000
	group Bridge	0.0000	0.0020	0.0000	0.0000
	group Lasso	0.0000	0.0070	0.0000	0.0000
4.1	lasso	0.1950	0.1973	0.1650	0.2325
	elastic net	0.1800	0.1943	0.1650	0.2263
	group MCP	0.1650	0.1682	0.1500	0.1850
	group Bridge	0.1950	0.1988	0.1650	0.2400
	group Lasso	0.1600	0.1623	0.1400	0.1850
4.2	lasso	0.0950	0.0976	0.0800	0.1113
	elastic net	0.0950	0.0950	0.0800	0.1100
	group MCP	0.0875	0.0901	0.0750	0.1063
	group Bridge	0.0800	0.0772	0.0550	0.0950
	group Lasso	0.0875	0.0894	0.0750	0.1013
4.3	lasso	0.0800	0.0793	0.0650	0.0900
	elastic net	0.0800	0.0792	0.0650	0.0900
	group MCP	0.0700	0.0705	0.0600	0.0850
	group Bridge	0.0600	0.0593	0.0400	0.0750
	group Lasso	0.0700	0.0736	0.0600	0.0850
4.4	lasso	0.1100	0.1116	0.0950	0.1313
	elastic net	0.0104	0.1086	0.0900	0.1263
	group MCP	0.1050	0.1028	0.0850	0.1250
	group Bridge	0.0850	0.0935	0.0750	0.1163
	group Lasso	0.1050	0.1023	0.0850	0.1163

Table 3.8: Percentage of correctly identified important variables and correctly removed unimportant variables

examples	methods	non-zero var.(%)	zero var.(%)
1.1	lasso	2(66.7%)	5(100%)
	elastic net	2(66.7%)	5(100%)
	group mcp	2(66.7%)	5(100%)
	group bridge	2(66.7%)	5(100%)
	group lasso	2(66.7%)	4(80%)
2.1	lasso	11,0 (55%)	20(100%)
	elastic net	15,0 (75%)	20(100%)
	group mcp	11,1 (60%)	18(90%)
	group bridge	15,4 (95%)	4(20%)
	group lasso	15,2 (85%)	16 (80%)
3.1	lasso	2,2,2,(40%)	25 (100%)
	elastic net	4,4,4,(80%)	25 (100%)
	group mcp	2,3,3,(53%)	25 (100%)
	group bridge	3,3,3,(60%)	25 (100%)
	group lasso	5,5,5,(100%)	24 (96%)
3.2	lasso	4,4,4, (80%)	25 (100%)
	elastic net	5,5,5, (100%)	25 (100%)
	group mcp	4,4,4, (80%)	18 (72%)
	group bridge	5,5,5, (100%)	6 (24%)
	group lasso	5,5,5, (100%)	22 (88%)
3.3	lasso	4,4,4, (80%)	25(100%)
	elastic net	5,5,5 (100%)	25(100%)
	group mcp	4,4,5, (87%)	19(76%)
	group bridge	5,5,5 (100%)	7(28%)
	group lasso	5,5,5 (100%)	22(88%)
3.4	lasso	4,4,4, (80%)	25(100%)
	elastic net	5,5,5 (100%)	25(100%)
	group mcp	5,5,5,(100%)	25(100%)
	group bridge	5,5,5 (100%)	5(20%)
	group lasso	5,5,5 (100%)	22(88%)

Table 3.9: Percentage of correctly identified important variables and correctly removed unimportant variables (Continued...)

examples	methods	non-zero var.(%)	zero var.(%)
3.5	lasso	4,5,4, (87%)	25(100%)
	elastic net	5,5,5 (100%)	25(100%)
	group mcp	5,5,5,(100%)	25(100%)
	group bridge	5,5,5 (100%)	4(16%)
	group lasso	5,5,5 (100%)	21(84%)
3.6	lasso	1,1,1,(20%)	25(100%)
	elastic net	1,2,1,(27%)	24(96%)
	group mcp	1,1,1,(20%)	23(92%)
	group bridge	3,3,3,(60%)	19(76%)
	group lasso	5,5,5,(100%)	18(72%)
4.1	lasso	0,0,0, (0%)	2,2,2, + 25 = 31 (100%)
	elastic net	1,0,0, (0%)	2,2,2, + 25 = 31 (100%)
	group mcp	1,1,1, (33%)	2,2,2, + 24 = 30 (97%)
	group bridge	0,0,0, (0%)	2,2,2, + 25 = 31 (100%)
	group lasso	3,3,3, (100%)	0,0,0, + 21 = 21 (68%)
4.2	lasso	2,2,2, (67%)	2,1,2, + 25 = 30 (97%)
	elastic net	3,3,3, (100%)	1,1,1, + 25 = 28 (90%)
	group mcp	2,2,2, (33.33%)	2,1,2, + 24 = 29 (94%)
	group bridge	3,3,3, (100%)	0,0,0, + 6 = 6 (20%)
	group lasso	3,3,3, (100%)	0,0,0, + 20 = 20 (65%)
4.3	lasso	2,2,2, (68%)	2,1,2,+ 25 = 30 (98%)
	elastic net	3,3,3, (100%)	1,1,1, +25= 28 (90%)
	group mcp	2,2,2, (68%)	1,1,2, +22= 26 (84%)
	group bridge	3,3,3, (100%)	0,0,0, + 6= 6 (20%)
	group lasso	3,3,3, (100%)	0,0,0, +19= 19 (61%)
4.4	lasso	2,1,2, (83.33%)	3,2,3, +25=33 (97%)
	elastic net	2,2,2, (100%)	1,2,2, +25=30 (88%)
	group mcp	2,2,2, (100%)	2,2,2, +24=30 (88%)
	group bridge	2,2,2 (100%)	1,1,1, +24=27 (79%)
	group lasso	2,2,2 (100%)	0,0,0, +21=21 (62%)

Table 3.10: Number of variables selected per group in different examples

examples	methods	number of variables selected per group
4.1	lasso	0, 0, 0
	elastic net	1, 0, 0
	group mcp	1, 1, 1
	group bridge	0, 0, 0
	group lasso	5, 5, 5
4.2	lasso	2, 3, 2
	elastic net	4, 4, 4
	group mcp	2, 3, 2
	group bridge	5, 5, 5
	group lasso	5, 5, 5
4.3	lasso	2, 3, 2
	elastic net	4, 4, 4
	group mcp	3, 3, 2
	group bridge	5, 5, 5
	group lasso	5, 5, 5
4.4	lasso	2, 2, 2
	elastic net	4, 3, 3
	group mcp	3, 3, 3
	group bridge	4, 4, 4
	group lasso	5, 5, 5

Chapter 4

A Real Data Example

4.1 Application to SNP Data Analysis

The methods that we introduced are particularly useful in high-dimensional data, for instance, genome-wide association studies. Genome-wide association studies are a method used to identify genes involved in human disease. This method searches the genome for small variations, called single nucleotide polymorphisms or SNPs, that occur more frequently in people with a particular disease than in people without the disease. Each study can look at hundreds or thousands of SNPs at the same time. Researchers use data from this type of study to pinpoint genes that may contribute to a person's risk of developing a certain disease.

Since genome-wide association studies examine SNPs across the genome, they represent a promising way to study complex, common diseases in which many genetic variations contribute to a person's risk. This method has already identified SNPs related to some diseases including diabetes, heart abnormalities, Parkinson disease. Researchers hope that future genome-wide association studies will identify more SNPs associated with chronic diseases, as well as variations that affect a persons response to certain drugs and influence interactions between a persons genes and the environment.

The example we use here to demonstrate our methods involves genetic variation (SNPs) data from a case-control study of West Nile virus disease with 177 cases and 262 controls. We want to find the possible associations between certain SNPs and West Nile virus disease. The data is collected at Nabraska center with all individuals being white Caucasian. There are 439 individuals participated in the study and 500 SNP markers genotyped for individuals.

We code each SNP as 0, 1, 2 for homozygous ("AA"), heterozygous ("Aa"), and mutation (rare) homozygous ("aa") genotypes respectively according to the genotype frequency. We first screen data to exclude SNPs that have a call rate less than 95% (missing rate greater than 5%) or minor allele frequency less than 5%. The number of SNP markers left is reduced to 407 after screening. Our response variable is the case-control binary outcome of West Nile virus infection status. Our predictors are the 407 SNP markers. We first performed linkage disequilibrium test to look at the dependence structure of SNP markers between each other. Since the test result shows that the correlation between loci are relatively low (from 0.01-0.1), we choose to apply the lasso method to relatively independent SNP markers instead of grouping them and applying group variable selection methods. Our model is fitted by the Lasso Logistic regression with binary outcome. The choice of tuning parameter is determined by cross-validation method based on area of ROC curve.

Results from applying the Lasso logistic regression were obtained. 18 out of 407 SNPs were detected as associated with West Nile virus disease. The Lasso forces the rest of the irrelevant SNP markers' coefficients to be 0's. The coefficients of

Table 4.1: the predictor estimates by lasso logistic

SNP	Gene Symbol	Locus ID	Chromosome	Chromosome Position	Estimate
rs10036567	KIAA0141	9812	5	141307833	-0.1394971
rs1007863	PARVB	29780	22	44395451	0.01348572
rs10129889	DYNC1H1	1778	14	102508056	-0.1951717
rs10131813	HOMEZ	57594	14	23745533	-0.05694708
rs10282929	EEF1D	1936	8	144681777	0.008469059
rs1031257	LOC100420743	9842	6	127945594	0.08571146
rs10403787	C3P1	388503	19	10166375	0.09808576
rs10407445	ZIM3	114026	19	57649962	-0.0985149
rs10410943	ACTL9	284382	19	8808900	0.02246633
rs1043620	HSPA1L	3305	6	31783755	-0.2422276
rs1044240	ALCAM	214	3	105258861	0.09130686
rs10445686	RAB3GAP1	22930	2	135893372	-0.01607454
rs1046480	TMEM159	57146	16	21185384	0.0722214
rs10466026	CDH23	64072	10	73550969	0.0414965
rs1046844	SLC25A26	115286	3	66428282	0.05953302
rs1048369	GPC4	2239	X	132437337	-0.02237237
rs1050348	LAMA4	3910	6	112493872	-0.05538143

the relevant SNP markers are given in Table 4.1. The coefficient estimators of all SNP markers are very small, in the scale of 0.1, suggesting small effects of SNPs to West Nile virus disease. rs1044240 which lies on gene ALCAM (activated leukocyte cell adhesion molecule) has coefficient 0.091, by taking exp(0.091)= 1.095, having a minor allele “a” result in an increase in the odds of having West Nile virus disease to exp(0.081) = 1.095 times than without having one. rs1043620 which lies on gene HSPA1L (heat shock 70kDa protein 1-like) has coefficient -0.24, by taking exp(-0.24), having a minor allele “a” result in a decrease in the odds of having West Nile virus disease to exp(-0.24)=0.79 times than without having one.

Chapter 5

Conclusions and Discussion

5.1 Discussion

With the advance of technology, the collection and storage of data become easy to obtain. Huge amount of data are produced from biological experiment for statistician to analyze. The high-dimensional problems is becoming more and more common. The "large p, small n" problem, in which there are more number of variables than samples, is the main difficulty. Traditional approaches to regression breaks down and new method is in need. The penalized variable selection method is an effective method. In particular, the Lasso proposed by Tibshirani has gained much attention in the past few years.

Lasso works well for the variables which can be treated individually. When the variables are grouped, the lasso does not work well. For example, suppose the k^{th} group is unimportant, lasso will only force individual coefficient in the k^{th} group to be zero. However, the whole group coefficients should be zero altogether because the k^{th} group is unimportant. Lasso tends to make selection based on the strength of individual variables rather than the strength of the group. Therefore, lasso works well when variables are have low correlation with each other. It does not work well for the situation of collinearity.

Breheny and Huang introduce a framework that sheds light on the behavior of grouped penalization methods. Methods that take into account grouping information have recently begun to appear in the penalized regression literature. The group lasso enforces sparsity at the group level, rather than at the level of the individual covariates. Within a group, the covariates are either all equal to zero or else all nonzero. The group lasso has some attractive qualities, such as the fact that its objective function is convex (i.e., there are no local minima, only a single global minimum). However, the group lasso also has disadvantages. It produces a strong bias towards zero, over selects the true number of groups, and it is incapable of selecting important elements within a group. The group bridge produces sparse solutions both at the group level and at the level of the individual covariates within a group. Its solutions tend to exhibit less bias than those of the group lasso and have been shown to be asymptotically consistent for group selection. Unlike the group lasso, however, the group bridge objective function is nonconvex and not differentiable at $|\beta_j| = 0$, which in practice can lead to problems with model fitting. They also develop a new group penalty, group MCP, which its grouping assumptions are less severe than those of group lasso and group bridge. It also performs better than those competing methods in situations with substantial (greater than 50%) within-group sparsity.

The difference between the methods is the extent to which the form of the penalty enforces grouping: group lasso forces complete grouping, group MCP encourages grouping to a rather slight extent, and group bridge is somewhere in between. Therefore, when applying the group penalization method, the data structure need to be considered. Therefore, under real data analysis situation, if one expects that the proportion of nonzero group members to be greater than one-half, use group lasso; otherwise use group MCP. If one expects this proportion to be close to one-half, one may wish to use group bridge.

5.2 Future Research

There is still many future research can be pursued in this field of study. Firstly, a limitation of the current group variables methods is that they does not consider the situation where grouped variables overlap to each other. This may cause problem in gene expression studies where genes may be grouped by pathways that are not mutually exclusive. Therefore, a future research area would include overlap groups for variable selection. Secondly, Group lasso, group bridge and group MCP work well in certain situations and not in others. The future research area would be to develop a more robust method that performs well across a variety of situations. Thirdly, the statistical inference research on penalized regression models other than the lasso is also valuable. Fourthly, the applications of grouped penalization methods to specific field of study needs to be conducted. One application of particular interest is genome-wide association studies, where the SNPs are in groups by the gene they belong to or with high correlation to each other. Further study is needed regarding the impact of issues inherent in genetics (such as linkage, penetrance, and the genetic basis of the disease) upon the performance of penalized regression and other approaches of analyzing these data. In summary, there is still plenty of statistical challenges that need to be addressed by researchers in this field.

Bibliography

[1] A. Antoniadis and J. Fan. Regularization of Wavelet Approximations. *Journal of the American Statistical Association*, 96, 939-967, 2001

[2] K. Ayers and H. Cordell. SNP selection in genome-wide and candidate gene studies via penaalized logistic regression. *Genetic Epidemiology*, 34: 879-891, 2010.

[3] P. Breheny. *grpreg: Regularization paths for regression models with grouped covariates.* R package version 1.1, 2009.

[4] P. Breheny and J. Huang. Penalized methods for bi-level variable selection. *Statistics And Its Interface*, 2 (1): 369-380, 2009.

[5] L. Breiman. Better Subset Regression Using the Nonnegative Garrote. *Technometrics*, 37: 373-384, 1995.

[6] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least Angle Regression. *The Annals of Statistics*, 32: 407-499, 2004.

[7] J. Fan and R. Li. Variable selection via Nonconcave Penalized Likelihood and Its Oracle Properties. *Journal of the American Statistical Association*, 96: 1348-1360, 2001.

[8] J. Friedman, T. Hastie, and R. Tibshirani. A Note on the Group Lasso and a Sparse Group Lasso. 2010.

[9] J. Friedman, T. Hastie, and R. Tibshirani. Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software*, 33 (1): 1-22, 2010

[10] E. George and D. Foster. Calibration and Empirical Bayes Variable Selection. *Biometrika*, 87:731-747, 2000.

[11] E. George and R. McCulloch. Variable Selection via Gibbs Sampling. *Journal of the American Statistical Association*, 88: 881-889, 1993.

[12] T. Hastie, R. Tibshirani, and J. Friedman. *Elements of Statistical Learning: Data Mining, Inference and Prediction (Second Edition)*. Springer-Verlag, New York, 2009.

[13] C. J. Hoggart1, J. C. Whittaker, M. D. Iorio and D. J. Balding. Simultaneous Analysis of All SNPs in Genome-Wide and Re-Sequencing Association Studies. *PLoS Genetics*, 4, 2008

[14] J. Huang, S. Ma, H. Xie, and C. Zhang. A Group Bridge Approach for Variable Selection. *Biometrika*, 96: 339-355, 2009.

[15] S. Ma, X. Song, and J. Huang. Supervised group lasso with applications to microarray data analysis. 2007.

[16] G. J. Mclachlan, K-A. Do, and C. Ambroise. *Analyzing Microarray Gene Expression Data*. John Wiley & Sons, New Jersey, 2004.

[17] L. Meier. grplasso: *Fitting user specified models with Group Lasso penalty*, 2009. R package version 0.4-2.

[18] L. Meier, S. van der Geer, and P. Buhlmann. The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70: 53-71, 2008.

[19] M. Y. Park and T. Hastie. Penalized logistic regression for detecting gene interactions. *Biostatistics*, Oxford, England, 9: 30-50, 2008.

[20] C. Phillips. Online resources for SNP analysis. A review and route map. *Molecular Biotechnology*, 2007.

[21] L. Shen and E. C. Tan. Dimension reduction-based penalized logistic regression for cancer classification using microarray data. *IEEE/ACM Trans Comput Biol Bioinform*, 2 (2), 2005.

[22] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 58: 267-288, 1996.

[23] M. West, et al. Predicting the clinical status of human breast cancer using gene expression. *PNAS*, 98: 11462-11467, 2001.

[24] T. T. Wu, Y. F. Chen, T. Hastie, E. Sobel, and K. Lange. Genomewide Association Analysis by Lasso Penalized Logistic Regression. *Bioinformatics*, 25 (6): 714-721, 2009.

[25] Y. Wu, D. D. Boos, and L. A. Stefanski. Controlling Variable Selection by the Addition of Pseudovariables. *Journal of the American Statistical Association*, 102 (477): 235-243, 2007.

[26] J. Xie and L. M. Zeng. Group Variable Selection Methods and Their Applications in Analysis of Genomic Data. *Frontiers in Computational and Systems Biology*, *Computational Biology*, 15: 231, 2010.

[27] C. Yang, X. Wan, Q. Yang, H. Xue, and W. Yu. Identifying main effects and epistatic interactions from large-scale SNP data via adaptive group Lasso. *BMC Bioinformatics*, 11 (Suppl 1): S18, 2010.

[28] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68: 49-67, 2006.

[29] P. Zhao, G. Rocha, and B. Yu. Grouped and Hierarchical Model Selection through Composite Absolute Penalties. *The Annals of Statistics*, 37 (6): 3468-3497, 2009.

[30] J. Zhao and Z. Chen. A two-stage penalized logistic regression approach to case-control genome-wide association studies. 2010.

[31] N. F. Zhou and J. Zhu. Group variable selection via a hierarchical lasso and its oracle property. 2007.

[32] H. Zhou, JS. Sinsheimer, K. Lange and M. E. Sehl. Association screening of common and rare genetic variants by penalized regression. *Bioinformatics*, 26 (19): 2375-2382 (8), 2010.

[33] H. Zou. The Adaptive Lasso and Its Oracle Properties. *Journal of the American Statistical Association*, 101: 1418-1429, 2006.

[34] H. Zou and T. Hastie. Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society, Seires B,* 67 (2): 301-320, 2005.