A Moving-window penalization method and its applications

Minli Bao
University of Iowa

Copyright © 2017 Minli Bao

This dissertation is available at Iowa Research Online: https://ir.uiowa.edu/etd/5710

Recommended Citation
https://ir.uiowa.edu/etd/5710.

Follow this and additional works at: https://ir.uiowa.edu/etd
Part of the Applied Mathematics Commons
A MOVING-WINDOW PENALIZATION METHOD AND ITS APPLICATIONS

by

Minli Bao

A thesis submitted in partial fulfillment of the requirements for the Doctor of Philosophy degree in Applied Mathematical and Computational Sciences in the Graduate College of The University of Iowa

August 2017

Thesis Supervisor: Professor Kai Wang
This is to certify that the Ph.D. thesis of

Minli Bao

has been approved by the Examining Committee for the thesis requirement for the Doctor of Philosophy degree in Applied Mathematical and Computational Sciences at the August 2017 graduation.

Thesis Committee:  
Kai Wang, Thesis Supervisor

Patrick Breheny

Weimin Han

Palle Jorgensen

Surjit Khurana
To my family
ACKNOWLEDGEMENTS

I would like to thank my advisor Professor Kai Wang for his help and support in completing this work. I truly appreciate the patience and guidance which he devoted to discussing with me. I am thankful to the members of my defense committee, Professor Patrick Breheny, Professor Weimin Han, Professor Palle Jorgensen and Professor Surjit Khurana, for their invaluable assistance. I would also like to thank Professor Weimin Han for his support during my study at the University of Iowa.

I want to thank many friends and teachers at the University of Iowa who assisted and supported my research, teaching, and life over the years.

Finally, I would like to thank my parents for their endless encouragement.
ABSTRACT

Genome-wide association studies (GWAS) has played an import role in identifying genetic variants underlying human complex traits. However, its success is hindered by weak effect at causal variants and noise at non-causal variants. Penalized regression can be applied to handle GWAS problems. GWAS data has some specificities. Consecutive genetic markers are usually highly correlated due to linkage disequilibrium.

This thesis introduces a moving-window penalized method for GWAS which smooths the effects of consecutive SNPs. Simulation studies indicate that this penalized moving window method provides improved true positive findings. The practical utility of the proposed method is demonstrated by applying it to Genetic Analysis Workshop 16 Rheumatoid Arthritis data.

Next, the moving-window penalty is applied on generalized linear model. We call such an approach as smoothed lasso (SLasso). Coordinate descent computing algorithms are proposed in details, for both quadratic and logistic loss. Asymptotic properties are discussed. Then based on SLasso, we discuss a two-stage method called MW-Ridge. Simulation results show that while SLasso can provide more true positive findings than Lasso, it has a side-effect that it includes more unrelated random noises. MW-Ridge can eliminate such a side-effect and result in high true positive rates and low false detective rates. The applicability to real data is illustrated by using GAW 16 Rheumatoid Arthritis data.
The SLasso and MW-Ridge approaches are then generalized to multivariate response data. The multivariate response data can be transformed into univariate response data. The causal variants are not required to be the same for different response variables. We found that no matter how the causal variants are matched, being fully matched or 60% matched, MW-Ridge can always over perform Lasso by detecting all true positives with lower false detective rates.
Public Abstract

Genome-wide association studies (GWAS) have played an important role in identifying genetic variants underlying human complex traits. However, its success is hindered by weak effect at causal variants and noise at non-causal variants. In an effort to overcome these difficulties, Liu et al. (2013) proposed a regularized regression method that penalizes on the difference of signal strength between two consecutive single-nucleotide polymorphisms (SNPs). We provide a generalization to this method so that more adjacent SNPs can be considered. The choice of the optimal number of markers is studied. Simulation studies indicate that this penalized moving window method provides improved true positive findings. The practical utility of the proposed method is demonstrated by applying it to Genetic Analysis Workshop 16 Rheumatoid Arthritis data.

The moving-window penalty is then extended to generalized linear model. We call such an approach as smoothed lasso (SLasso). The asymptotic properties are studied. A two-stage method called moving-window ridge (MW-Ridge) is then derived from the moving-window regularized regression approach. Simulation results show that while SLasso can provide more true positive findings than Lasso, it has a side-effect that it includes more unrelated random noises. MW-Ridge can eliminate such a side-effect and result in high true positive rates and low false detective rates. The practical utility of these two approaches is illustrated by applying them to GAW 16 Rheumatoid Arthritis data.
TABLE OF CONTENTS

LIST OF TABLES ......................................................... ix
LIST OF FIGURES ......................................................... xi

CHAPTER

1 INTRODUCTION ....................................................... 1
  1.1 Penalized Regression Methods ................................. 1
  1.2 Genetic Background ............................................ 5
  1.3 Smoothed Minimax Concave Penalization Method ............. 7
  1.4 Overview of This Thesis ........................................ 9

2 MOVING-WINDOW PENALIZATION METHOD FOR GENOME-WIDE ASSOCIATION STUDIES .......................... 11
  2.1 The Moving-window Penalization Method ....................... 11
  2.2 Computing Algorithm ........................................... 12
    2.2.1 Continuous Traits ......................................... 13
    2.2.2 Dichotomous Traits ......................................... 15
  2.3 Seletion of Tuning Parameters ................................ 17
    2.3.1 Selection of Tuning Parameters \( \lambda \) and \( \eta \) .... 17
    2.3.2 Selection of Tuning Parameter \( d \) ....................... 19
  2.4 Simulation ..................................................... 20
  2.5 Empirical Study of GAW 16 Rheumatoid Arthritis Data ....... 24

3 TWO-STAGE MOVING-WINDOW RIDGE METHOD .................. 32
  3.1 Generalized Linear Model with Moving-window Penalty ...... 32
    3.1.1 Model ....................................................... 33
    3.1.2 Computation ................................................ 33
      3.1.2.1 Continuous Outcome .................................. 33
      3.1.2.2 Binary Outcome ...................................... 34
    3.1.3 Asymptotic Properties .................................... 36
  3.2 A Two-stage Method: MW-Ridge ................................. 39
    3.2.1 Motivation ................................................ 39
    3.2.2 Method .................................................... 40
  3.3 Simulation ..................................................... 42
    3.3.1 Simulation 0: Experiment on GAW 16 RA Data ........... 42
3.3.2 Simulation 1: Continuous Outcome and Uncorrelated Predictors ........................................... 45
3.3.3 Simulation 2: Continuous Outcome and Correlated Predictors ............................................. 49
3.3.4 Simulation 3: Binary Outcome and Uncorrelated Predictors .................................................... 57
3.3.5 Simulation 4: Binary Outcome and Correlated Predictors ...................................................... 61
3.4 Application .................................................................................................................................. 66
3.5 Appendix ...................................................................................................................................... 69

4 MULTIVARIATE RESPONSE DATA ANALYSIS .............................................................................. 72

4.1 Method ......................................................................................................................................... 72
4.2 Simulation .................................................................................................................................... 75

5 CONCLUSION AND FUTURE WORK ........................................................................................... 83

REFERENCES .................................................................................................................................... 87
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>PPV and TPR over 100 replicates in mean(standard deviation)</td>
<td>22</td>
</tr>
<tr>
<td>2.2</td>
<td>PPV and TPR over 100 replicates in mean(standard deviation)</td>
<td>24</td>
</tr>
<tr>
<td>3.1</td>
<td>Median value of true positive, median value of model size, mean(standard deviation) of PPV and TPR, over 100 replicates.</td>
<td>43</td>
</tr>
<tr>
<td>3.2</td>
<td>Mean of predictive mean squared error over $\sigma^2$ (PMSE/$\sigma^2$), true positive rate (TPR), and false detective rate (FDR), for seven different methods with different $\sigma$ values, using 100 simulations each with $p = 50$, $p = 100$, and $p = 200$. Continuous outcome, uncorrelated predictors.</td>
<td>48</td>
</tr>
<tr>
<td>3.3</td>
<td>Mean of predictive mean squared error over $\sigma^2$ (PMSE/$\sigma^2$), true positive rate (TPR), and false detective rate (FDR), for seven different methods with different $\sigma$ values, using 100 simulations each with $p = 50$, $p = 100$, and $p = 200$. Continuous outcome, correlated predictors.</td>
<td>53</td>
</tr>
<tr>
<td>3.4</td>
<td>Mean of predictive error rate (PER), true positive rate (TPR), and false detective rate (FDR), for eight different methods and for different $c$ values, using 100 simulations each with $p = 50$, $p = 100$, and $p = 200$. Binary outcome, uncorrelated predictors.</td>
<td>59</td>
</tr>
<tr>
<td>3.5</td>
<td>Mean of predictive error rate (PER), true positive rate (TPR), and false detective rate (FDR), for eight different methods and for different $c$ values, using 100 simulations each with $p = 50$, $p = 100$, and $p = 200$. Binary outcome, correlated predictors.</td>
<td>64</td>
</tr>
<tr>
<td>3.6</td>
<td>Results of GAW 16 RA data.</td>
<td>68</td>
</tr>
<tr>
<td>4.1</td>
<td>Simulation results for the first response variable in mean(standard deviation) with all matched non-zero $\beta$s.</td>
<td>77</td>
</tr>
<tr>
<td>4.2</td>
<td>Simulation results for the second response variable in mean(standard deviation) with all matched non-zero $\beta$s.</td>
<td>78</td>
</tr>
<tr>
<td>4.3</td>
<td>Values of $\alpha$’s for SLasso and MW-R2 in mean(standard deviation) with all matched non-zero $\beta$s.</td>
<td>79</td>
</tr>
</tbody>
</table>
4.4 Simulation results for the first response variable in mean(standard deviation). 40% of the regression coefficients are not matched. . . . . . . . . . 80

4.5 Simulation results for the second response variable in mean(standard deviation). 40% of the regression coefficients are not matched. . . . . . . 81

4.6 Values of $\alpha$’s for SLasso and MW-R2 in mean(standard deviation). 40% of the regression coefficients are not matched. . . . . . . . . . . . . . . 82
LIST OF FIGURES

Figure

2.1 Estimated value of $|\beta|$ across the genome for the GAW 16 data using single SNP marginal logistic regression. .............................................. 25

2.2 Estimated value of $|\beta|$ across the genome for the GAW 16 data using LASSO. 26

2.3 Estimated value of $|\beta|$ across the genome for the GAW 16 data using moving-window regression with $d = 2$. ............................................. 27

2.4 Estimated value of $|\beta|$ across the genome for the GAW 16 data using moving-window regression with $d = 6$. ............................................. 28

2.5 Estimated values $|\beta|$ on chromosome 6 for the GAW 16 data. From top to bottom: 1) LASSO; 2) Moving-window method with $d = 2$; 3) Moving-window method with $d = 6$. ............................................. 29

2.6 Estimated values $|\beta|$ on chromosome 6 as $d = 2$. From top to bottom: 1) $\gamma_2 = 0.1$; 2) $\gamma_2 = 0.05$; 3) $\gamma_2 = 0.025$. ............................................. 30

2.7 Estimated values $|\beta|$ on chromosome 6 as $d = 6$. From top to bottom: 1) $\gamma_2 = 0.1$; 2) $\gamma_2 = 0.05$; 3) $\gamma_2 = 0.025$. ............................................. 31

3.1 The coefficients paths for three methods. From top to bottom: 1) LASSO; 2) MW-2; 3) MW-4. The paths for the five true positives are shown in different colors. The paths in pink are for the coefficients of the false positives. ............................................. 41

3.2 Median number of false positives, continuous outcome and uncorrelated predictors, with different values of $\sigma$. From top to bottom: 1) $p = 50$, relaxo covered by MW-R4 before $\sigma = 7$, MW-R2 covered by MW-R4 after $\sigma = 7$; 2) $p = 100$; 3) $p = 200$. ............................................. 50

3.3 Median number of true positives, continuous outcome and uncorrelated predictors, with different values of $\sigma$. From top to bottom: 1) $p = 50$; 2) $p = 100$; 3) $p = 200$, MW-R2 covered by MW-R4 after $\sigma = 7$, relaxo covered by MW-R2 before $\sigma = 4$. ............................................. 51
3.4 Median number of false positives, continuous outcome and correlated predictors, with different values of $\sigma$. From top to bottom: 1) $p = 50$, MW-R2 covered by MW-R4 before $\sigma = 7$, relaxo covered by MW-R2 after $\sigma = 7$; 2) $p = 100$; 3) $p = 200$.

3.5 Median number of true positives, continuous outcome and correlated predictors, with different values of $\sigma$. From top to bottom: 1) $p = 50$, MW-R2 fully covered by MW-R4; 2) $p = 100$, LASSO covered by MW-R2 before $\sigma = 7$, net covered by MW-R4 after $\sigma = 7$; 3) $p = 200$, LASSO covered by MW-R2 before $\sigma = 7$.

3.6 Median number of false positives, binary outcome and uncorrelated predictors, with different values of $c$. From top to bottom: 1) $p = 50$, MW-R2 covered by MW-R4 after $c = 0.5$; 2) $p = 100$; 3) $p = 200$.

3.7 Median number of true positives, binary outcome and uncorrelated predictors, with different values of $c$. From top to bottom: 1) $p = 50$, LASSO fully covered by net, net covered by SL after $c = 0.5$. MW-R2 covered by MW-R4 after $c = 0.5$; 2) $p = 100$, net covered by SL after $c = 0.5$; 3) $p = 200$, LASSO covered by net before $c = 1$, MW-R2 fully covered by MW-R4.

3.8 Median number of false positives, binary outcome and correlated predictors, with different values of $c$. From top to bottom: 1) $p = 50$, MW-R2 covered by MW-R4 before $c = 1$; 2) $p = 100$, MW-R2 fully covered by MW-R4; 3) $p = 200$, MW-R2 covered by MW-R4 between $c = 1$ and $c = 0.5$.

3.9 Median number of true positives, binary outcome and correlated predictors, with different values of $c$. From top to bottom: 1) $p = 50$, svm covered by MW-R2 after $c = 1$, net covered by MW-R4 after $c = 0.5$; 2) $p = 100$, LASSO and MW-R2 both fully covered by MW-R4; 3) $p = 200$, MW-R2 fully covered by MW-R4, LASSO covered by MW-R4 before $c = 0.5$, net covered by MW-R4 before $c = 1$.
CHAPTER 1
INTRODUCTION

Variable selection is an important problem in statistics. The purpose of variable selection is to select predictors and measure their effects on the response variable. Traditional methods for variable selection include forward, backward, and stepwise regression. In these methods, the model fitness is evaluated at each step. As the number of predictors increases, the computation expense grows quadratically. In the context of high-dimensional data where sample size is small and feature size is large, it is impossible to apply traditional variable selection methods due to collinearity. Regularized regression methods are becoming more and more popular for the variable selection problem arising from high-dimensional data.

1.1 Penalized Regression Methods

In classical statistics, consider a linear regression model as:

\[ y = X\beta + \epsilon, \]

where \( y \) is the response variable, \( X \) is the feature matrix, \( \beta \) is the regression coefficient vector and \( \epsilon \) is the error term. The least square estimate of \( \beta \) can be defined as:

\[ \hat{\beta} = \arg\min_{\beta} \{ ||y - X\beta||^2_2 \}. \]

We can work out \( \hat{\beta} \) if there is \( n > p \) where \( n \) is the sample size and \( p \) is the number of features. However, in high-dimensional data where \( n \ll p \), there is more than one solution of \( \hat{\beta} \). Regularized regression methods have become a feasible way to solve
the variable selection problems where \( n \ll p \). LASSO (least absolute shrinkage and selection operator) (28) is a widely used regularized regression method in which \( L_1 \) norm is imposed as a penalty for the regression coefficients. The LASSO method is to minimize

\[
\frac{1}{2n}||y - X\beta||_2^2 + \lambda||\beta||_1,
\]

where \( \lambda \geq 0 \) is the tuning parameter for the \( L_1 \) penalty which encourages sparsity of the coefficients. LASSO is able to select features which have strong effects on the response variable, and at the same time, shrink the coefficients estimates towards zero. However, LASSO has the following disadvantage: if the features are highly correlated, which is usually the case for real-world dataset, LASSO tends to select only one feature from a group of correlated features.

Elastic net (35) is able to accommodate correlated features by incorporating \( L_1 \) and \( L_2 \) penalty. The elastic net method minimizes:

\[
\frac{1}{2n}||y - X\beta||_2^2 + \lambda(\alpha||\beta||_1 + \frac{1}{2}(||\beta||_2^2)),
\]

in which \( \lambda \geq 0 \) and \( 0 \leq \alpha \leq 1 \). In elastic net, the penalty is imposed as a linear combination of the penalties in LASSO and ridge regression (14). Ridge regression will select all coefficients as nonzero regardless how large the tuning parameter \( \lambda \) is. Besides, although ridge regression doesn’t help in feature selection, it has better predictive performance than traditional linear regression. As a result, elastic net is able to both achieve sparse coefficient estimates and handle the case of collinearity.

Both LASSO and elastic net are convex regularized problems. There exists a biased effect and the coefficient estimates are shrunk towards zero. In recent years,
there are several non-convex regularized regression methods been developed, including bridge regression (12), smoothly clipped absolute deviation (SCAD) (8) and minimax concave penalty (MCP) (34). These non-convex methods are able to result in sparse coefficients and at the same time, reduce the shrinkage effects that cause bias.

However, these above methods do not take the spatial structure of the feature matrix into consideration. Usually, there is a natural spatial structure in some real world dataset. For example, in genome-wide association studies (GWAS), the order of genetic variants are correspondent to their locations in the chromosomes. Taking advantage of the spatial structure can help to discover causative factors as well as remove unrelated noisy variables.

If the predictors can be separated into different groups, then the group LASSO (33) is a reasonable choice. It imposes the L2-norm of the coefficients in the same groups. The correspondent convex optimization problem minimizes

$$\frac{1}{2n} ||y - \sum_{\ell=1}^{L} X_\ell \beta_\ell ||_2^2 + \lambda \sum_{\ell=1}^{L} ||\beta_\ell ||_2,$$

(5)

where $X_\ell$ and $\beta_\ell$ stand for the feature matrix and the regression coefficient for group $\ell$. Group LASSO can achieve the sparsity over groups and the coefficients in the same group will be estimated as either all zero or all nonzero. Furthermore, if the $L1$ penalty of LASSO is also included, then the nonzero coefficients will be sparse in each group. Similar as group LASSO, there are other group selection methods for high-dimensional data including group MCP and group bridge (15). Such methods are the natural extensions of MCP and bridge regression.

Before applying these above group selection methods, there comes a problem
difficult to accommodate for real world dataset: how to separate the features into groups. For example, in GWAS, the consecutive genetic variants are usually highly correlated (which is called as the linkage disequilibrium (LD) effect) thus it is difficult to separate them into groups. Unlike group LASSO, fused LASSO (29) is a penalized regression method which takes advantage of the sequential structure without the need to determine feature groups.

Fused LASSO introduces a smoothing penalty on the absolute values of the difference between two consecutive coefficients. This convex optimization problem minimizes:

\[
\frac{1}{2n} ||y - X\beta||_2^2 + \lambda ||\beta||_1 + \eta \sum_{j=1}^{p-1} |\beta_j - \beta_{j+1}|. \tag{6}
\]

Thus the distances between consecutive features are encouraged to be zero. There are some methods sharing some similarities with fused LASSO and smoothing the absolute values of the differences between the coefficients of two highly correlated predictors, such as the graph-structured general fused LASSO method (4; 5).

Smoothed minimax concave penalization (SMCP) (22) is developed as a regularized regression method which proposes a smoothing penalty different from fused LASSO. In this thesis, the penalty in SMCP is generalized to a moving-window penalty. Since SMCP is designed specifically for GWAS, I would introduce the genetic background before discussing the details of SMCP.
1.2 Genetic Background

Deoxyribonucleic acid (DNA) is a molecule that carries the genetics instructions used in the growth, development, functioning and reproduction of all known living organisms and many viruses. DNA molecules consist of two strands coiled around each other. The two DNA strands are known as polynucleotides. They are composed of nucleotides. Each nucleotide is composed of a nitrogen-containing nucleobase - either cytosine (C), guanine (G), adenine (A), or thymine (T). The nucleotides are joined to one another. The bases are paired under the rule: A with T while C with G. The two strands of DNA run in opposite directions to each other. The sequence of the bases contains the biological information. A gene is a region of DNA which contains such genetic information. Inside a living cell, DNA is organized into long structures called chromosomes. There are 23 pairs of chromosomes in human beings.

A genome is the genetic material of an organism which consists of DNA. A single nucleotide polymorphism (SNP) is a variation that occurs at a specific position in the genome. For example, at a specific base position in human genome, the base T might appear in most individuals while A in a minority. The two possible nucleotide variations (T or A) are called alleles. Mostly, chromosomes are paired. The score of SNP depends on the choice of reference allele. Assume A and T are alleles for one locus. The genotypes for such locus can be AA, AT, and TT. Given A as the reference allele, the genotypes are coded as 2, 1, and 0. However, given T as the reference allele, the genotypes are coded as 0, 1, and 2.

The alleles at different loci are associated by linkage disequilibrium (LD). LD
is influenced by many factors, including selection, the rate of mutation, genetic drift, the system of mating, population structure, and genetic linkage. The pattern of LD in a genome is a powerful signal of the population genetic processes that are structuring it.

Genome-wide association studies (GWAS) is the examination of many common genetic markers in different individuals to see whether the markers are associated with a trait. It has been found by numerous studies that many complex diseases are associated with genetic variants among populations. However, these identified variants explain only a small fraction of the heritability for most complex traits (23; 19; 7). Hence there is a need for improved statistical methods.

Usually, GWAS depends on the single marker analysis which testing the association between each SNP and the disease trait. Such analysis can be marginal linear regression or marginal logistic regression. However, the single marker analysis has the following problems: If a strong association does exist, then the single marker analysis is able to select out the markers. In reality, only few SNPs are associated with the trait. The single SNP analysis can find out a large set of potential effective signals and it can be pretty hard to find out the truly effective ones.

From a statistical point of view, identifying SNPs associated with the trait is a variable selection problem in a sparse, high-dimensional model setting. It is a sparse problem because those genetic variants are rare between individuals and only a small number of markers are truly positive. It is a high-dimensional problem because that the number of SNPs (hundreds of thousands) is usually much larger than the
sample sizes (thousands). Penalized regressions are powerful tools to deal with such problem. SMCP is designed specifically for GWAS. Further discussion is provided in the following section.

1.3 Smoothed Minimax Concave Penalization Method

Let $p$ be the number of SNPs and $n$ the total number of subjects. The SNPs are indexed in their chromosomal order. The genotype score of subject $i$ at SNP $j$ is denoted by $x_{ij}$. Let $n_j$ be the number of subjects whose genotype at SNP $j$ is non-missing and $\Theta_j$ the set of indices of such subjects. That is, $\sum_{i \in \Theta_j} 1 = n_j$. Genotype scores are normalized as usual such that $\sum_{i \in \Theta_j} x_{ij} = 0$ and $\sum_{i \in \Theta_j} x_{ij}^2 = n_j$. The phenotype of subject $i$ is denoted by $y_i$.

The loss function, denoted by $Q(\beta)$, is defined through a set of marginal models, one for each SNP, in order to magnify the effect of each SNP. Here $\beta$ is a vector of regression coefficients. Define the quadratic marginal loss function $Q(\beta)$ as:

$$Q(\beta) = \frac{1}{2} \sum_{j=1}^{p} \frac{1}{n_j} \sum_{i \in \Theta_j} (y_i - x_{ij} \beta_j)^2,$$

where $\beta = (\beta_1, \cdots, \beta_p)^T$. The marginal loss function is different from the conventional joint loss. The joint loss is defined as:

$$\sum_i (y_i - \sum_j x_{ij} \beta_j)^2$$

The marginal loss is based on single SNP analysis, while the joint loss is based on the joint linear regression such that

$$y = X\beta + \epsilon.$$
In this thesis, Chapter 2 focuses on marginal loss, while Chapter 3 and 4 focuses on joint loss.

For dichotomous traits, the loss function is the negative log likelihood loss defined as

$$Q(\alpha, \beta) = - \sum_{j=1}^{p} \frac{1}{n_j} \sum_{i \in \Theta_j} (y_{ij} \log p_{ij} + (1 - y_{ij}) \log (1 - p_{ij})),$$

where $p_{ij} = \Pr(y_{ij} = 1|x_{ij}) = (e^{\alpha_j + x_{ij} \beta_j}) / (1 + e^{\alpha_j + x_{ij} \beta_j})$, $\alpha = (\alpha_1, \cdots, \alpha_p)^T$, $\beta = (\beta_1, \cdots, \beta_p)^T$.

In order to achieve sparsity, SMCP imposes the following MCP penalty

$$\rho(\beta_j; \lambda, \gamma) = \lambda \int_{0}^{[\beta_j]} (1 - x/(\gamma \lambda))_+ dx.$$

Here $\lambda$ is the penalty parameter, $\gamma$ is the regularization parameter which controls concavity and $x_+ = x 1_{\{x \geq 0\}}$. It approaches the $L1$ norm penalty in LASSO as $\gamma \to \infty$ and approaches the hard-thresholding as $\gamma \to 1+$.

It is expected that adjacent SNPs in high LD should have similar strength of association with the phenotypes. In SMCP, the following penalty is proposed in order to encourage smoothness between consecutive $|\beta|$s:

$$S(\beta_j; \eta) = \frac{\eta}{2} \zeta_j(|\beta_j| - |\beta_{j+1}|)^2,$$

where $\eta$ is the tuning parameter and $\zeta_j$ measures the strength of LD, defined as $\zeta_j = |\text{Corr}(X_j, X_{j+1})|$. The smoothing penalty is only encouraged when there exists strong correlation between adjacent predictors. The estimate $\hat{\beta}$ is the minimizer of
the following objective function:

\[ Q(\beta) + \sum_{j=1}^{p} \rho(\beta_j; \lambda, \gamma) + \sum_{j=1}^{p-1} S(\beta_j; \eta). \]

Replacing \( Q(\beta) \) by the negative log likelihood loss \( Q(\alpha, \beta) \) leads to the objective function for dichotomous traits.

The score of SNP depends on the choice of reference allele. A genotype can be coded as either 0 or 2, depending on the reference alleles. The effect of association will be switched to its opposite if the choice of reference allele is altered. The difference between two coefficients is sensitive to the choices of reference alleles, while the magnitudes of coefficients are invariant with such choices. Therefore, fused LASSO is not suitable for GWAS. Since SMCP shrinks the squared differences between the magnitudes of two adjacent coefficients, it is applicable to GWAS.

1.4 Overview of This Thesis

In Chapter 2, I introduce the moving-window penalty which is a general form of the smoothing penalty in SMCP. The moving-window penalty is applied to genome-wide screening. A coordinate descent algorithm is developed for coefficient estimation. I discuss how to select the values of tuning parameters. The moving-window method is applied to genetic analysis workshop (GAW) 16 rheumatoid arthritis (RA) data.

In Chapter 3, I apply the moving-window penalty in a predictive model. The method is called as smoothed LASSO (SLasso). A coordinate descent algorithm is proposed in details. I discuss the asymptotic properties. Furthermore, a two-stage method: moving-window ridge (MW-Ridge) is derived from SLasso. These two
methods are applied to GAW 16 RA data.

In Chapter 4, SLasso and MW-Ridge are applied to multivariate response data. I discuss how to transform multivariate response data into univariate response data, and study the behaviors of these two methods through simulations.

The moving-window method proposed in Chapter 2 is implemented for both linear and logistic regression models, in the R package \texttt{MWLasso}. MW-Ridge discussed in Chapter 3 is implemented for both linear and logistic regressions, in the R package \texttt{MWRidge}.
CHAPTER 2
MOVING-WINDOW PENALIZATION METHOD FOR GENOME-WIDE ASSOCIATION STUDIES

This chapter is organized as follows. Section 2.1 introduces the moving-window penalty. Section 2.2 discusses the computing algorithm. Section 2.2.1 discusses the case for continuous trait and Section 2.2.2 discusses the case for dichotomous trait. Section 2.3 discusses the selection of tuning parameters. Section 2.4 presents the simulation experiments. In Section 2.5, the moving-window method is applied to GAW 16 rheumatoid arthritis data.

2.1 The Moving-window Penalization Method

In SMCP, the strength of effects between two adjacent SNPs are smoothed. In the context of GWAS, the effect of LD may well extend beyond two adjacent SNPs. Based on this consideration, we replace the smoothing penalty in SMCP by a penalty that involves $d$ consecutive SNPs, where the value of $d$ is determined by data and can be larger than two. Consider a moving window of size $d$ that scans all SNPs from the beginning to the end. For SNPs in the same window, they are considered to be close enough and are expected to have similar strength of effects. Therefore, there effect size in terms of $|\beta|$ are expected to be similar. Let $W_s$ denote the set of SNP indices in the $s$th moving window. The total number of $W_s$ is $p - d + 1$: $W_1 = \{1, \cdots, d\}$, $W_2 = \{2, \cdots, d+1\}, \cdots, W_{p-d+1} = \{p-d+1, \cdots, p\}$. The smoothing penalty within
\( W_s, s = 1, 2, \ldots, p - d + 1 \) is defined as:

\[
S(W_s; \eta) = \eta \cdot \frac{1}{2(d-1)} \sum_{k,j \in W_s, k < j} \zeta_{k,j} (|\beta_k| - |\beta_j|)^2,
\]

where the weight \( \zeta_{i,j} \) is defined as \( \zeta_{i,j} = \text{Corr}(X_i, X_j) \). Each pair of adjacent SNPs will be scanned by \( d-1 \) windows, so \( \frac{1}{(d-1)} \) is used to standardize the smoothing effect and make sure that for any size \( d \), the overall weight for \( (|\beta_j| - |\beta_{j+1}|)^2 \) is fixed as \( \zeta_{j,j+1} \). Especially, when \( d = 2 \), \( S(W_s; \eta) \) is exactly the same as the smoothing penalty of SMCP.

As for the penalty responsible for SNP selection, we choose the LASSO penalty instead of MCP. This is because the LASSO penalty is easier to deal with and it is a limiting case of MCP. The LASSO penalty is defined as:

\[
\rho(\beta_j; \lambda) = \lambda |\beta_j|.
\]

So the objective function is defined as:

\[
L_n(\beta) = Q(\beta) + \sum_{j=1}^{p} \rho(\beta_j; \lambda) + \sum_{s=1}^{p-d+1} S(W_s; \eta),
\]

where \( Q(\beta) \) is the quadratic loss in the objective function of SMCP. Replacing \( Q(\beta) \) by \( Q(\alpha, \beta) \) (negative log likelihood loss in the objective function of SMCP) leads to the objective function for dichotomous traits, denoted by \( L_n(\alpha, \beta) \).

### 2.2 Computing Algorithm

As in other high dimensional problems, a major challenge in estimating the model parameters is to find out a computational feasible way to optimize the objective function \( L_n(\beta) \) or \( L_n(\alpha, \beta) \). LAR has been discussed as a feasible computation
method for LASSO (6). Coordinate descent algorithm has been applied for both LASSO and elastic net (10) and some non-convex problems such as SCAD and MCP (3). Block coordinate descent algorithm can be applied to grouped LASSO (11; 9). For fused LASSO, the coordinate descent algorithm diverges. Alternative direction method of multipliers (ADMM) (31) and majorization-minimization (MM) algorithm (5) can be applied to minimize the objective function for fused LASSO. For SMCP, the coordinate descent algorithm is appropriate and there is an explicit solution in updating each \( \beta_j \) (22). For the moving-window regression, the coordinate descent algorithm is applicable. Details are described below.

### 2.2.1 Continuous Traits

Given current values \( \beta_k, k \neq j \), \( \beta_j \) is updated by the minimizer of \( \tilde{L}_n(\beta_j) \) which is defined as

\[
\tilde{L}_n(\beta_j) = \frac{1}{2n_j} \sum_{i \in \Theta_j} (y_i - x_{ij}\beta_j)^2 + \lambda |\beta_j| + \tilde{S}_n(\beta_j)
\]

where

\[
\tilde{S}_n(\beta_j) = \frac{\eta}{2(d-1)} \sum_{s=j-d+1}^{j-1} \sum_{k \in W_s, k \neq j} \zeta_{k,j} (|\beta_k| - |\beta_j|)^2.
\]

It is straightforward to verify that

\[
\tilde{L}_n(\beta_j) = P_j \beta_j^2 + Q_j \beta_j + R_j |\beta_j| + C,
\]
where $C$ represents a term free of $\beta_j$ and

$$P_j = \frac{1}{2}(\frac{1}{n_j} \sum_{i \in \Theta_j} x_{ij}^2 + \frac{\eta}{d-1} \sum_{s=j-d+1}^{j-1} \sum_{k \in W_s, k \neq j} \zeta_{k,j}),$$

$$Q_j = -\frac{1}{n_j} \sum_{i \in \Theta_j} x_{ij} y_i,$$

$$R_j = \lambda - \frac{\eta}{d-1} \sum_{s=j-d+1}^{j-1} \sum_{k \in W_s, k \neq j} \zeta_{k,j} |\beta_k|.$$ 

To minimize $\tilde{L}_n(\beta_j)$ is equivalent to minimize $P_j \beta_j^2 + Q_j \beta_j + R_j |\beta_j|$ over $\beta_j$. The minimizer is

$$\hat{\beta}_j = -\text{sgn}(Q_j) \cdot \frac{(|Q_j| - R_j)_+}{2P_j}.$$ 

We note that $P_j$ and $Q_j$ are free of $\beta_k$, $k = 1, \ldots, p$. They can be computed in advance.

The coordinate descent algorithm proceeds as Algorithm 2.1.

**Algorithm 2.1 Coordinate Descent Method for Continuous Traits**

1. Compute $P_j, Q_j, j = 1, \ldots, p$ for $t = 0$
2. Input the initial values $(\hat{\beta}_1^{(0)}, \ldots, \hat{\beta}_p^{(0)})$
3. repeat
4. for $j = 1, \ldots, p$ do
5. Fix $\hat{\beta}_k^{(t)}, k \neq j$
6. Compute $R_j$
7. Update $\hat{\beta}_j^{(t)}$
8. end for
9. $t \leftarrow t + 1$
10. until $\hat{\beta}$ converges

The convergence of the coordinate descent algorithm can be shown as follows: the objective function can be written in the form of $f_0(\beta_1, \ldots, \beta_p) + f_1(\beta_1, \ldots, \beta_p)$.
Here $f_0$ is the summation of the loss function and the smoothing penalty, while $f_1 = \lambda \sum_{j=1}^{p} |\beta_j|$. Given Lemma 3.1 in (30), $f_0$ is regular in the sense of (5) in (30). Since $f_1$ is separable, the coordinate descent algorithm will converge to a stationary point, which should be a coordinatewise minimal point of the objective function (22).

2.2.2 Dichotomous Traits

For dichotomous traits, we can use the quadratic loss function and hence apply Algorithm 2.1 described in the previous subsection. Now we introduce the coordinate descent method for the logistic loss function that applies only to dichotomous traits.

In the marginal logistic regression which tests the strength of association between the $j$th SNP and the phenotype, we define $\alpha_j$ as the coefficient for the constant effect and $\beta_j$ as the coefficient for the SNP effect. Let $\alpha = (\alpha_1, \ldots, \alpha_p)^T$ and $\beta = (\beta_1, \ldots, \beta_p)^T$. The coordinate descent algorithm to minimize $L_n(\alpha, \beta)$ depends on iteratively reweighted least squares. We use the following quadratic approximation to $Q(\alpha, \beta)$:

$$Q(\alpha, \beta) = \sum_{j=1}^{p} \frac{1}{2n_j} \sum_{i \in \Theta_j} w_{ij} (z_{ij} - \alpha_j - x_{ij}\beta_j)^2,$$

where

$$z_{ij} = \alpha_j + x_{ij}\beta_j + \frac{y_i - p_{ij}}{p_{ij}(1 - p_{ij})},$$
$$w_{ij} = p_{ij}(1 - p_{ij}).$$

Here $p_{ij} = 1/(1 + \exp(-\alpha_j - x_{ij}\beta_j))$. So the ‘working’ objective function is

$$\tilde{L}_n(\alpha, \beta) = Q(\alpha, \beta) + \lambda \sum_{j=1}^{p} |\beta_j| + \sum_{s=1}^{p-d+1} S(W_s; \eta).$$
The optimization problem is transformed to minimize an iteratively reweighted penalized linear squared function. Note that in this function, the intercept terms \( \{\alpha_j\}_{j=1,...,p} \) cannot be omitted. This is different from the case of continuous traits discussed earlier. To ease computation burden, we fixed the values of \( \{\alpha_j\}_{j=1,...,p} \) at their respective estimates computed from their marginal logistic regressions. The optimization of \( \tilde{L}_n(\alpha, \beta) \) is conducted with respect to \( \beta \) only as follows.

Given the current values \( \{\beta_k\}_{k \neq j} \), \( \beta_j \) is updated by the minimizer of \( \tilde{L}_n(\beta_j) \) which is given by

\[
\tilde{L}_n(\beta_j) = \frac{1}{2n_j} \sum_{i \in \Theta_j} w_{ij}(z_{ij} - \alpha_j - x_{ij}\beta_j)^2 + \lambda|\beta_j| + \frac{\eta}{2(d-1)} \sum_{s=j-d+1}^j \sum_{k,j \in W_s,k \neq j} \zeta_{k,j}(|\beta_k| - |\beta_j|)^2
\]

\[
= P_j \beta_j^2 + Q_j \beta_j + R_j |\beta_j| + C
\]

where \( C \) is a term free of \( \beta_j \),

\[
P_j = \frac{1}{2} \left( \frac{1}{n_j} \sum_{i \in \Theta_j} w_{ij}x_{ij}^2 + \frac{\eta}{d-1} \sum_{s=j-d+1}^j \sum_{k,j \in W_s,k \neq j} \zeta_{k,j} \right),
\]

\[
Q_j = -\frac{1}{n_j} \sum_{i \in \Theta_j} w_{ij}x_{ij}(z_{ij} - \alpha_j),
\]

\[
R_j = \lambda - \frac{\eta}{d-1} \sum_{s=j-d+1}^j \sum_{k,j \in W_s,k \neq j} \zeta_{k,j} |\beta_k|.
\]

The minimizer of \( \tilde{L}_n(\beta_j) \) is

\[
\hat{\beta}_j = -\text{sgn}(Q_j) \cdot \frac{(|Q_j| - R_j)_+}{2P_j}.
\]

The coordinate descent algorithm proceeds as Algorithm 2.2.

Such algorithm can be considered as two nested loops: the outer loop is to update the quadratic approximation \( \tilde{L}_n \) by using the current parameters \( \hat{\alpha} \) and \( \hat{\beta} \),
Algorithm 2.2 Coordinate Descent Method for Dichotomous Traits

1. Estimate \((\alpha_1, \ldots, \alpha_p)\) from the marginal logistic regressions. These values are then fixed.
2. \(t \leftarrow 0\)
3. Input the initial values \((\hat{\beta}_1^{(0)}, \ldots, \hat{\beta}_p^{(0)})\)
4. Compute \(z_{ij}, w_{ij}, j = 1, \ldots, p\)
5. Compute \(P_j, Q_j, j = 1, \ldots, p\)
6. repeat
7. repeat
8. for \(j = 1, \ldots, p\) do
9. Fix \(\hat{\beta}_k^{(t)}, k \neq j\)
10. Compute \(R_j\)
11. Update \(\hat{\beta}_j^{(t)}\)
12. end for
13. until \(\hat{\beta}\) converges
14. Update \(z_{ij}\) and \(w_{ij}\) by using the current estimates \(\hat{\beta}_j^{(t)}\)
15. Update \(P_j, Q_j, j = 1, \ldots, p\)
16. \(t \leftarrow t + 1\)
17. until \(\hat{\beta}\) converges

while the inner loop is to run the coordinate descent algorithm on the penalized weighted quadratic function.

2.3 Selection of Tuning Parameters

2.3.1 Selection of Tuning Parameters \(\lambda\) and \(\eta\)

There are various ways for tuning parameter selection. Common methods include AIC, BIC, and cross-validation. These methods are intended to measure the prediction ability of the covariates. In GWAS, it is very likely that the covariates, i.e., the genotypes of disease-causing variants, are not directly observed although they are expected to be in LD with the observed SNPs. Furthermore, the goal of GWAS is to
identify associated SNPs rather than prediction. Most importantly, the loss functions considered in this report are both marginal. Given these considerations, AIC, BIC, and cross-validation are not suitable for the proposed method. A bisection method has been used to search for the values of tuning parameters (32; 22). We use the same method here. This method needs a specification of a number of SNPs to be selected up front. Let \( m \) denote this number. The tuning parameters are then determined so that the number of SNPs is no less than \( m \) (while keeping it as close to \( m \) as possible).

We begin by reparameterization the tuning parameters \( \lambda \) and \( \eta \) as follows:

\[
\gamma_1 = \lambda + \eta, \quad \gamma_2 = \lambda / \gamma_1.
\]

It is straightforward to verify that when \( \gamma_2 = 1 \) the proposed moving-window method is equivalent to the 'LASSO' method as the smoothing penalty disappeared at \( \eta = 0 \). To proceed, \( \gamma_2 \) is fixed at 0.05. This is the values used for SMCP (22). The value of \( \gamma_1 \) is determined by the bisection method. Let \( \gamma_{1\text{max}} \) be the largest value of \( \gamma_1 \) under which at least one SNP selected by LASSO. It is known that \( \gamma_{1\text{max}} = \max_j |\sum_{i \in \Theta_j} x_{ij}y_i|/(n_j \gamma_2) \). Since \( \gamma_1 \) cannot be zero, the lower bound of \( \gamma_1 \) is set to be \( \gamma_{1\text{min}} = \epsilon \gamma_{1\text{max}} \). We set \( \epsilon = 0.1 \). The bisection method involves an iterative process. Set \( \gamma_{1u} = \gamma_{1\text{max}} \) and \( \gamma_{1\ell} = \gamma_{1\text{min}} \). We compute \( \gamma_{1\text{mid}} = \frac{1}{2}(\gamma_{1u} + \gamma_{1\ell}) \). Let \( r(\gamma_1) \) be the number of non-zero predictors selected under \( \gamma_1 \). If \( r(\gamma_{1\text{mid}}) < m \), replace \( \gamma_{1u} \) by \( \gamma_{1\text{mid}} \). Otherwise, if \( r(\gamma_{1\text{mid}}) > m \), replace \( \gamma_{1\ell} \) by \( \gamma_{1\text{mid}} \). Repeat the process until \( r(\gamma_{1\text{mid}}) = m \). As a result, we are able to select the value of tuning parameters \( \lambda \) and \( \eta \).
2.3.2 Selection of Tuning Parameter $d$

The window size $d$ controls the number of SNPs to be included in a window. When $d = 1$, the moving-window model is equivalent to the LASSO model. When $d = 2$, the smoothing penalty reduces to the one used in SMCP. Here we apply the empirical mean of the absolute value of lag-$(d-1)$ autocorrelations to select $d$. Let

$$s(d) = \frac{1}{p - d + 1} \sum_{j=1}^{p-d+1} |\text{Corr}(X_j, X_{j+d-1})|.$$ 

$s(d)$ is expected to be a non-increasing function of $d$. If so, its largest value occurs at $d = 2$. For any given value $\rho$ which is not too small but is less than $s(2)$, there would be a value $d' > 2$ such that $s(d') < \rho$. The value of $d$ is determined by $d = \max\{d : s(d) \geq \rho\}$. Technically, the value of $\rho$ is restricted to be no smaller than $\rho_t$, which is equal to $\epsilon \cdot s(2)$ for a small value $\epsilon$. The value of $\rho$ is pre-specified. In the simulation study, we will try $\rho = 0.4, 0.35, 0.3, 0.25,$ and $0.2$. In the empirical study to be reported later, we use $\rho = 0.3$ and $0.4$.

Since $d$ takes only integer values, it should be pretty fast to select its value. If necessary, this procedure can be sped up by using a bisection method. Starting with $\rho_u = s(2)$ and $\rho_l = \epsilon \rho_u$, set $d = 2$ as $d_l$. $s(d)$ are evaluated at $d = 2, 2^2, 2^3, \ldots$. The smallest $d$ that satisfies $s(d) < \rho$ is denoted by $d_u$. Then we apply the bisection method. We set $d_{\text{mid}} = (d_u + d_l)/2$. If $s(d_{\text{mid}}) < \rho$, replace $d_u$ by $d_{\text{mid}}$. Otherwise if $s(d_{\text{mid}}) \geq \rho$, replace $d_l$ by $d_{\text{mid}}$. As a result, we will find out the largest $d$ such that $s(d) \geq \rho$ faster.
2.4 Simulation

In the simulation study, genotypes are taken from rheumatoid arthritis (RA) study which was provided by the Genetic Analysis Workshop (GAW) 16 (1). There are 2062 subjects. 400 subjects are randomly selected. 5000 genotypes are randomly selected from chromosome 6. In order to maintain the sequential structure, the 5000 genotypes are consecutive. The genotypes are standardized in advance such that $\sum_i x_{ij} = 0$ and $\sum_i x_{ij}^2 = 400$. The continuous trait $y$ is generated from the linear model:

$$y_i = \mathbf{x}_i^T \boldsymbol{\beta} + \epsilon_i, \ i = 1, ..., 400.$$  \hspace{1cm} (1)

where $\mathbf{x}_i \in \mathbb{R}^{5000}$ is a vector of SNP scores of subject $i$, $\boldsymbol{\beta}$ is the vector of genetic effects for these SNPs. $\epsilon_i$ is the random residual sampled from a normal distribution with mean 0 and variance 3. The entries of $\boldsymbol{\beta}$ are all 0, except that $(\beta_{1501}, \ldots, \beta_{1512}) = (-0.3, 0.2, -0.25, 0.2, -0.6, 0.7, -0.5, 0.1, -0.5, 0.3)$ and $(\beta_{1514}, \ldots, \beta_{1532}) = (0.25, -0.4, 0.2, -0.1, -0.25, 0.3, -0.4, 0.15, 0.3, -0.4, 0.4, -0.5, 0.2, -0.3, 0.16, 0.36, -0.2, 0.1)$. That is, the number of truly non-zero $\beta$s is 31.

For dichotomous trait, the linear part in the logistic regression model is sampled in the same way as continuous trait. Then $y_i$ are generated from the Bernoulli distribution: $\Pr(y_i = 1|x_i) = 1/(1 + \exp(- (\alpha + \mathbf{x}_i^T \boldsymbol{\beta})))$, where $\alpha = 0$.

For continuous trait, we used the marginal quadratic loss. For dichotomous trait, both the marginal quadratic loss and the marginal logistic loss are used.

People usually use AIC, BIC, and mean squared error to evaluate the model performance. However, these methods are appropriate to predictive models and are
inappropriate in this study. For GWAS data, we chose to use the marginal linear regression model. It is not a predictive model thus the above criterion cannot be used. Therefore, we use positive predictive value (PPV) and true positive rate (TPR) to evaluate the model performances. They are derived from the number of TP (true positive), FP (false positive), and FN (false negative). The definitions are shown as follows.

\[
PPV = \frac{TP}{TP + FP},
\]

\[
TPR = \frac{TP}{TP + FN}.
\]

PPV and TPR are also known as precision and recall, respectively. The tuning parameter \( \gamma_2(= \lambda/\gamma_1) \) is fixed at 0.05. After several trials, the tuning parameter \( \gamma_1(= \lambda + \eta) \) is chosen such that the number of selected SNPs (those with non-zero coefficients) is \( m = 45 \). The values of \( \rho \) are set as \((0.4, 0.35, 0.3, 0.25, 0.2)\). Respectively, \( d = (2, 4, 6, 10, 15) \). The simulation results based on 100 replicates are shown in Table 2.1.

First of all, there is indeed a performance improvement as more SNPs are included in the smoothing window. From LASSO, which can be regarded as a special case of the proposed moving-window method with \( d = 1 \), to the moving-window method with \( d = 6 \), both TPR and PPV are increasing for both the continuous trait and the binary trait with either quadratic loss or logistic loss. For the binary trait, this trend continues up to \( d = 10 \). After \( d = 6 \) the performance for the continuous trait starts to decrease. The same is also true for the binary trait for \( d > 10 \). This phenomenon is probably due to over-smoothing. Overall, \( d = 6 \) seems to be a suitable
Table 2.1. PPV and TPR over 100 replicates in mean(standard deviation).

<table>
<thead>
<tr>
<th>PPV or TPR</th>
<th>method</th>
<th>continuous trait</th>
<th>binary trait *</th>
<th>binary trait **</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPV</td>
<td>LASSO</td>
<td>0.4460(0.0423)</td>
<td>0.4312(0.0399)</td>
<td>0.4313(0.0400)</td>
</tr>
<tr>
<td></td>
<td>d=2</td>
<td>0.6024(0.0453)</td>
<td>0.4940(0.0561)</td>
<td>0.5787(0.0536)</td>
</tr>
<tr>
<td></td>
<td>d=4</td>
<td>0.6611(0.0349)</td>
<td>0.5612(0.0526)</td>
<td>0.6399(0.0392)</td>
</tr>
<tr>
<td></td>
<td>d=6</td>
<td>0.6790(0.0229)</td>
<td>0.6352(0.0429)</td>
<td>0.6753(0.0260)</td>
</tr>
<tr>
<td></td>
<td>d=10</td>
<td>0.6779(0.0247)</td>
<td>0.6498(0.0441)</td>
<td>0.6756(0.0275)</td>
</tr>
<tr>
<td></td>
<td>d=15</td>
<td>0.6649(0.0421)</td>
<td>0.6378(0.0520)</td>
<td>0.6580(0.0440)</td>
</tr>
<tr>
<td>TPR</td>
<td>LASSO</td>
<td>0.6477(0.0618)</td>
<td>0.6261(0.0580)</td>
<td>0.6261(0.0580)</td>
</tr>
<tr>
<td></td>
<td>d=2</td>
<td>0.8745(0.0658)</td>
<td>0.7171(0.0812)</td>
<td>0.8400(0.0778)</td>
</tr>
<tr>
<td></td>
<td>d=4</td>
<td>0.9597(0.0507)</td>
<td>0.8145(0.0764)</td>
<td>0.9293(0.0570)</td>
</tr>
<tr>
<td></td>
<td>d=6</td>
<td>0.9887(0.0332)</td>
<td>0.9223(0.0624)</td>
<td>0.9803(0.0378)</td>
</tr>
<tr>
<td></td>
<td>d=10</td>
<td>0.9855(0.0424)</td>
<td>0.9432(0.0640)</td>
<td>0.9809(0.0399)</td>
</tr>
<tr>
<td></td>
<td>d=15</td>
<td>0.9652(0.0611)</td>
<td>0.9258(0.0755)</td>
<td>0.9551(0.0638)</td>
</tr>
</tbody>
</table>

* binary trait with quadratic loss.

** binary trait with logistic loss.

choice of the moving window size for this data.

At $d = 6$, the PPV for continuous trait is increased by 52%, compared to LASSO (from 0.446 to 0.679) and by 12% compared to the case of $d = 2$ (from 0.6024 to 0.679). The TPR also increases 52% and 12%, respectively, compared to LASSO (from 0.6477 to 0.9887) and the case of $d = 2$ (from 0.8745 to 0.9887).

For the binary trait with quadratic loss, at $d = 6$ the PPV increases 47% and 29%, respectively, compared to the LASSO (from 0.4312 to 0.6352) and the case of $d = 2$ (from 0.4940 to 0.6352) while the TPR is increased by 47% and 29%, respectively. For the binary trait with logistic loss, the PPV in increased by 57% and 17%, respectively, compared to LASSO and the case of $d = 2$ while the TPR is increased by 57% and 17%, respectively, as well. The performance with logistic loss
is better than the performance with quadratic loss in terms of both PPV and TPR. The downside of the logistic loss is that it involves more computation time.

Next, we consider a simulation study in which neighboring SNPs do not have similar coefficients in the true model. The genotypes are the same as the previous simulation study. The elements of $\beta$ are all 0, except that $(\beta_{1501}, \beta_{1524}, \beta_{1530}, \beta_{2400}, \beta_{2403}) = (-0.8, 0.4, -0.4, 1.2, -0.8)$. The number of truly non-zero $\beta$s is 5. Unlike the previous simulation study, the truly non-zero $\beta$s are not consecutive. $\beta_{1524}$ and $\beta_{1530}$ will be smoothed only if $d \geq 7$, while $\beta_{2400}$ and $\beta_{2403}$ will be smoothed only if $d \geq 4$. The tuning parameter $\gamma_2 (= \lambda/\gamma_1)$ is fixed at 0.05. The tuning parameter $\gamma_1 (= \lambda + \eta)$ is chosen such that the number of selected SNPs (i.e., those with non-zero coefficients) is $m = 7$. Simulation results based on 100 simulation replicates are shown in Table 2.2.

In this simulation study, the assumption that neighboring SNPs have similar $\beta$ values is not true. From lasso to the moving-window method with $d = 4$, both TPR and PPV decrease slightly for both the continuous trait and the binary trait with either quadratic loss or logistic loss. For the continuous trait, the PPV decreased from 0.5557 (lasso) to 0.5343 (d=4). The TPR decreased from 0.778 (lasso) to 0.748 (d=4), respectively. After $d = 4$, the PPV and TPR remain stable up to $d = 15$, for both the continuous trait and the binary trait with either quadratic loss or logistic loss. The differences in the PPV and TPR are not significant, suggesting that the moving-window method performs no worse than lasso even the assumption is misspecified.
Table 2.2. PPV and TPR over 100 replicates in mean(standard deviation).

<table>
<thead>
<tr>
<th>PPV or TPR</th>
<th>method</th>
<th>continuous trait</th>
<th>binary trait *</th>
<th>binary trait **</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPV</td>
<td>LASSO</td>
<td>0.5557(0.1283)</td>
<td>0.5786(0.1258)</td>
<td>0.5786(0.1258)</td>
</tr>
<tr>
<td></td>
<td>d=2</td>
<td>0.5557(0.1283)</td>
<td>0.5743(0.1251)</td>
<td>0.5736(0.1254)</td>
</tr>
<tr>
<td></td>
<td>d=4</td>
<td>0.5343(0.1229)</td>
<td>0.5586(0.1302)</td>
<td>0.5593(0.1296)</td>
</tr>
<tr>
<td></td>
<td>d=6</td>
<td>0.5357(0.1241)</td>
<td>0.5600(0.1246)</td>
<td>0.5571(0.1227)</td>
</tr>
<tr>
<td></td>
<td>d=10</td>
<td>0.5314(0.1285)</td>
<td>0.5586(0.1237)</td>
<td>0.5571(0.1243)</td>
</tr>
<tr>
<td></td>
<td>d=15</td>
<td>0.5343(0.1278)</td>
<td>0.5586(0.1237)</td>
<td>0.5571(0.1227)</td>
</tr>
<tr>
<td>TPR</td>
<td>LASSO</td>
<td>0.778(0.1796)</td>
<td>0.810(0.1761)</td>
<td>0.810(0.1761)</td>
</tr>
<tr>
<td></td>
<td>d=2</td>
<td>0.778(0.1796)</td>
<td>0.804(0.1752)</td>
<td>0.802(0.1764)</td>
</tr>
<tr>
<td></td>
<td>d=4</td>
<td>0.748(0.1720)</td>
<td>0.782(0.1822)</td>
<td>0.782(0.1822)</td>
</tr>
<tr>
<td></td>
<td>d=6</td>
<td>0.750(0.1738)</td>
<td>0.784(0.1745)</td>
<td>0.780(0.1717)</td>
</tr>
<tr>
<td></td>
<td>d=10</td>
<td>0.744(0.1800)</td>
<td>0.782(0.1731)</td>
<td>0.780(0.1741)</td>
</tr>
<tr>
<td></td>
<td>d=15</td>
<td>0.748(0.1789)</td>
<td>0.782(0.1731)</td>
<td>0.780(0.1717)</td>
</tr>
</tbody>
</table>

* binary trait with quadratic loss.
** binary trait with logistic loss.

2.5 Empirical Study of GAW 16 Rheumatoid Arthritis Data

Rheumatoid arthritis (RA) is a complex human disorder with a prevalence ranging from around 0.8% in Caucasians to 10% in some native American groups (1). Several studies showed that rheumatoid arthritis was associated with genetic markers (16) (26). GWA 16 data is from the North American Rheumatoid Arthritis Consortium (NARAC). It is the initial batch of whole genome association data for the NARAC cases (N=868) and controls (N=1194) after removing duplicated and contaminated samples. There are 531689 markers across 22 chromosomes. The phenotype \( y \) is binary, where \( y = 0 \) for controls and \( y = 1 \) for cases. The SNP scores were standardized in advance. We first did a regular genome-wide association study in which each SNP was tested for association individually. The \( \beta \) values computed
from a simple logistic regression were presented in Figure 2.1. A strong association
signal is present on chromosome 6. At the same time, there are also many $\beta$ values
that are small, representing noise signals. We hope to obtain a sparse set of sig-
nals. We apply LASSO and the proposed moving-window method for the purpose of
selecting SNPs and compare their results.

For LASSO, the tuning parameter $\lambda$ is 0.0516 and $\eta$ is 0 in order to choose
$m = 800$ SNPs. The 800 SNPs chosen by LASSO are identical to the 800 SNPs which
are most strongly correlated with the phenotype in the marginal regression shown in
Figure 2.1. The estimated $\beta$s are presented in Figure 2.2. Most of the selected SNPs
are located on chromosome 6. On all other chromosomes, there are few non-zero $\beta$s.
Here is what we did with the proposed moving-window method. Simulation study in the previous section suggests $\rho = 0.3$ is an appropriate choice for GAW 16 data. In the simulation study only chromosome 6 data were used to determine the size $d$ of the moving window and we got $d = 6$ for $\rho = 0.3$. Now we use data from all chromosome to determine $d$. The value for $d$ is turned out to be again 6 given that $s(6) = 0.3068$ and $s(7) = 0.2891$. In addition, we also considered the case of $\rho = 0.4$. This is also a value of $\rho$ considered in the simulation study where the corresponding $d$ was 2. Interestingly, the value of $d$ for $\rho = 0.4$ remains 2 when data from all genomes are used. The tuning parameter $\gamma_2 (= \lambda/\gamma_1)$ is set at 0.05 and the tuning parameter $\gamma_1 (= \lambda + \eta)$ was chosen such that the number of selected SNPs (i.e., those whose $\beta$s are not equal to 0) is 800. When $d = 2$, the tuning parameters are $\lambda = 0.0524$ and $\eta = 0.9964$. When $d = 6$, we have $\lambda = 0.0531$ and $\eta = 1.0094$. The estimated $\beta$s for
these two situations are presented in Figure 2.3 and Figure 2.4, respectively.

Figure 2.3. Estimated value of $|\beta|$ across the genome for the GAW 16 data using moving-window regression with $d = 2$.

There is a shrinkage effect on the estimates of $|\beta|$. For LASSO, the max value of $|\beta|$ is around 0.19. It becomes 0.14 and 0.11 for $d = 2$ and $d = 6$, respectively. Such effect comes from the smoothing penalty. As window size $d$ increases, the effect of the smoothing penalty becomes stronger. The moving-window method also has a clustering effect. It tends to choose adjacent SNPs with high LD together. For instance, as $d$ increases from 0 (i.e., LASSO), 2, to 6, more and more selected SNPs are located on chromosome 6. The number of selected SNPs is 489 for LASSO, 513 for the case of $d = 2$, and 540 for $d = 6$. In order to show the shrinkage effect and clustering effect more clearly, the $\beta$ estimates on chromosome 6 are displayed in
Figure 2.4. Estimated value of $|\beta|$ across the genome for the GAW 16 data using moving-window regression with $d = 6$.

Figure 2.5 for LASSO, $d = 2$, and $d = 6$.

In order to explore the effect of different $\gamma_2$ values, the $\beta$ estimates on chromosome 6 under different $\gamma_2$ values are displayed in Figure 2.6, with $d = 2$. As $d = 6$, the $\beta$ estimates on chromosome 6 under different $\gamma_2$ values are displayed in Figure 2.7. A lower $\gamma_2$ value indicates stronger smoothing effect, resulting $\beta$ estimates closer to zero.
Figure 2.5. Estimated values $|\beta|$ on chromosome 6 for the GAW 16 data. From top to bottom: 1) LASSO; 2) Moving-window method with $d = 2$; 3) Moving-window method with $d = 6$. 
Figure 2.6. Estimated values $|\beta|$ on chromosome 6 as $d = 2$. From top to bottom: 1) $\gamma_2 = 0.1$; 2) $\gamma_2 = 0.05$; 3) $\gamma_2 = 0.025$. 
Figure 2.7. Estimated values $|\beta|$ on chromosome 6 as $d = 6$. From top to bottom: 1) $\gamma_2 = 0.1$; 2) $\gamma_2 = 0.05$; 3) $\gamma_2 = 0.025$. 
CHAPTER 3
TWO-STAGE MOVING-WINDOW RIDGE METHOD

In Chapter 2, the moving-window smoothing penalty has been introduced and applied in genome-wide association studies to select SNPs. Marginal regression is used in order to magnify the effect of each genetic variant. In this chapter, the moving-window penalty is applied to the generalized linear model.

This chapter is organized as follows. Section 3.1 introduces a regularized regression method called smoothed LASSO (SLasso). Section 3.2 discusses a two-stage method called moving-window ridge (MW-Ridge). In Section 3.3, the performances of SLasso, MW-Ridge and several other regularized methods are compared in simulation experiments. In Section 3.4, both SLasso and MW-Ridge are applied to GAW 16 RA data.

3.1 Generalized Linear Model with Moving-window Penalty

In classical linear regression model,

\[ y_i = \beta_0 + \sum_{j=1}^{p} X_{ij} \beta_j + \epsilon_i, i = 1, \ldots, n, \]  

(1)

where \( y_i \) is the outcome of the \( i \)th sample, \( X_{ij} \) is \( j \)th feature of the \( i \)th sample, \( \beta_j \) is the coefficient of the \( j \)th column of the feature matrix, and \( \epsilon_i \) is iid random noise. One might combine the \( L_1 \) penalty and the moving-window penalty when the feature number \( p \) is much larger than the sample size \( n \).
3.1.1 Model

As defined in Chapter 2, suppose the window size is \( d \), then the smoothing penalty for the \( s \)th moving window \( W_s, s = 1, \ldots, p - d \) is written as:

\[
\frac{1}{2(d - 1)} \sum_{k,j \in W_s, k < j} \zeta_{k,j}(|\beta_k| - |\beta_j|)^2.
\] (2)

The weight of the smoothing effect is defined as \( \zeta_{k,j} = |\text{Corr}(X_k, X_j)| \). One needs to minimize the objective function:

\[
\frac{1}{2n} \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| + \frac{\eta}{2(d - 1)} \sum_{s=1}^{p-d+1} \sum_{k,j \in W_s, k < j} \zeta_{k,j}(|\beta_k| - |\beta_j|)^2.
\] (3)

In Chapter 2, the loss function is the marginal loss so it is only good for variable selection but not good for prediction. Here we use the joint loss instead. With window size as two, we call such method smoothed LASSO (SLasso).

3.1.2 Computation

We implement the coordinate descent algorithm for both linear model and binary outcome logistic model.

3.1.2.1 Continuous Outcome

Given current values \( \beta_k, k \neq j, \beta_j \) is updated by the minimizer of \( \tilde{L}_n(\beta_j) \) which is defined as

\[
\tilde{L}_n(\beta_j) = \frac{1}{2n} \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2 + \lambda |\beta_j| + \tilde{S}_n(\beta_j),
\]

where

\[
\tilde{S}_n(\beta_j) = \frac{\eta}{2(d - 1)} \sum_{s=j-d+1}^{j} \sum_{k \in W_s, k \neq j} \zeta_{k,j}(|\beta_k| - |\beta_j|)^2.
\]
It is straightforward to verify that

\[ \tilde{L}_n(\beta_j) = P_j \beta_j^2 + Q_j \beta_j + R_j |\beta_j| + C, \]

where \( C \) represents a term free of \( \beta_j \) and

\[
\begin{align*}
P_j &= \frac{1}{2} \left( \frac{1}{n} \sum_{i=1}^{n} X_{ij}^2 + \frac{\eta}{d-1} \sum_{s=j-d+1}^{j} \sum_{k \in W_s, k \neq j} \zeta_{k,j} \right), \\
Q_j &= \frac{-1}{n} \sum_{i=1}^{n} X_{ij} r_i, \\
R_j &= \lambda - \frac{\eta}{d-1} \sum_{s=j-d+1}^{j} \sum_{k \in W_s, k \neq j} \zeta_{k,j} |\beta_i|,
\end{align*}
\]

with \( r_i = y_i - \sum_{k \neq j} X_{ik} \beta_k \).

To minimize \( \tilde{L}_n(\beta_j) \) is equivalent to minimize \( P_j \beta_j^2 + Q_j \beta_j + R_j |\beta_j| \) over \( \beta_j \).

The minimizer is

\[
\hat{\beta}_j = -\text{sgn}(Q_j) \cdot \frac{(|Q_j| - R_j)_{+}}{2P_j}.
\]

We note that \( P_j \) is free of \( \beta \), which can be computed in advance. The coordinate descent algorithm proceeds as Algorithm 3.1.

As shown in Section 2.2, the objective function can be written in the form of \( f_0(\beta_1, ..., \beta_p) + f_1(\beta_1, ..., \beta_p) \). Here \( f_0 \) is the loss function added by the moving-window penalty, while \( f_1 = \lambda \sum_{j=1}^{p} |\beta_j| \). Since \( f_0 \) is a regular function and \( f_1 \) is separable, the coordinate descent solution converges to a coordinatewise minimum of \( f \), which is also a stationary point of \( f \) (30).

### 3.1.2.2 Binary Outcome

For binary outcome, negative log likelihood is used as the loss function. The coordinate descent algorithm to minimize the negative log likelihood depends on
Algorithm 3.1 Coordinate Descent Method for Continuous Outcome

1. Compute $P_j$, $j = 1, \ldots, p$ for $t = 0$
2. Input the initial values $(\hat{\beta}_1^{(0)}, \ldots, \hat{\beta}_p^{(0)})$
3. Initialize $r = y - \sum_{i=1}^p X_i \hat{\beta}_i^{(0)}$
4. repeat
   5. for $j = 1, \ldots, p$ do
      6. Fix $\hat{\beta}_k^{(t)}$, $k \neq j$
      7. Compute $r = r + X_j \hat{\beta}_j^{(t)}$
      8. Compute $Q_j$, $R_j$
      9. Update $\hat{\beta}_j^{(t)}$
     10. Update $r = r - X_j \hat{\beta}_j^{(t)}$
   end for
8. $t \leftarrow t + 1$
9. until $\hat{\beta}$ converges

iteratively reweighted least squares. Unlike linear model, the constant coefficient $\beta_0$
cannot be centered as zero in advance. Given the current values are $\beta_j$, $j = 0, 1, \cdots, p$,
we form $\tilde{L}_n(\beta)$ as a quadratic approximation to the objective function:

$$
\tilde{L}_n(\beta) = \frac{1}{2n} \sum_{i=1}^{n} w_i (z_i - \beta_0 - \sum_{j=1}^{p} X_{ij} \beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \\
+ \frac{\eta}{2(d-1)} \sum_{s=1}^{j-d+1} \sum_{k,j \in W_s, k \neq j} \zeta_{k,j} (|\beta_k| - |\beta_j|)^2,
$$

where

$$
\begin{align*}
z_i &= \beta_0 + \sum_{j=1}^{p} X_{ij} \beta_j + \frac{y_i - p_i}{p_i(1 - p_i)} \\
w_i &= p_i(1 - p_i).
\end{align*}
$$

Here $p_i = 1/(1 + \exp(-\beta_0 - \sum_{j=1}^{p} X_{ij} \beta_j))$. For $j = 1, \ldots, p$, the loss function can be
rewritten as:

$$
\tilde{L}_n(\beta_j) = P_j \beta_j^2 + Q_j \beta_j + R_j |\beta_j| + C,
$$
where $C$ is free of $\beta_j$, and

$$P_j = \frac{1}{2} \left( \frac{1}{n} \sum_{i=1}^{n} w_i X^2_{ij} + \frac{\eta}{d-1} \sum_{s=j-d+1}^{j} \sum_{k,j \in W_s, k \neq j} \zeta_{k,j} \right),$$

$$Q_j = -\frac{1}{n} \sum_{i=1}^{n} w_i X_{ij} r_i,$$

$$R_j = \lambda - \frac{\eta}{d-1} \sum_{s=j-d+1}^{j} \sum_{k,j \in W_s, k \neq j} \zeta_{k,j} |\beta_k|,$$

where $r_i = z_i - \beta_0 - \sum_{k \neq j}^p X_{ik} \beta_k$. Like in continuous outcome, we are able to get such explicit update as:

$$\hat{\beta}_j = -\text{sgn}(Q_j) \cdot \left( \frac{|Q_j| - R_j}{2P_j} \right).$$

We initialize $\hat{\beta}_0 = \ln \left( \frac{\Pr(y = 1)}{[1 - \Pr(y = 1)]} \right)$. We need to update $\hat{\beta}_0$: fix $\beta_j$, $j = 1, \ldots, p$,

$$\hat{\beta}_0 = \frac{1}{\sum_{i=1}^{n} w_i} \left( \sum_{i=1}^{n} w_i (z_i - \sum_{j=1}^{p} X_{ij} \beta_j) \right)$$

The coordinate descent algorithm proceeds as Algorithm 3.2.

### 3.1.3 Asymptotic Properties

We derived the asymptotic properties for the moving-window method which are analogous to those for the LASSO (18) and fused LASSO (29). The penalized criterion is:

$$\sum_{i=1}^{n} (y_i - x_i^T \beta)^2 + \lambda_n \sum_{j=1}^{p} |\beta_j| + \frac{\eta_n}{2(d-1)} \sum_{s=1}^{p-d+1} \sum_{k,j \in W_s, k < j} \zeta_{k,j} (|\beta_k| - |\beta_j|)^2,$$  \hspace{1cm} (4)

with $\beta = (\beta_1, \beta_2, \ldots, \beta_p)^T$ and $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T$. The Lagrangian multiplier $\lambda_n$ and $\eta_n$ are functions of sample size $n$. For simplicity, we assume that $p$ is fixed while
Algorithm 3.2 Coordinate Descent Method for Binary Outcome

1. For $t = 0$, Input the initial values $(\hat{\beta}_1^{(0)}, \ldots, \hat{\beta}_p^{(0)})$
2. Initialize $\hat{\beta}_0^{(0)}$
3. Compute $z_i, w_i, i = 1, \ldots, n$ by using $(\hat{\beta}_0^{(0)}, \hat{\beta}_1^{(0)}, \ldots, \hat{\beta}_p^{(0)})$
4. Compute $P_j, j = 1, \ldots, p$
5. Initialize $r = z - \hat{\beta}_0^{(0)} - \sum_{j=1}^p X_j \hat{\beta}_j^{(0)}$
6. repeat
7. repeat
8. Update $\hat{\beta}_0^{(t)}$
9. for $j = 1, \ldots, p$ do
10. Fix $\hat{\beta}_k^{(t)}, k \neq j$
11. Compute $r = r + X_j \hat{\beta}_j^{(t)}$
12. Compute $Q_j, R_j$
13. Update $\hat{\beta}_j^{(t)}$
14. Update $r = r - X_j \hat{\beta}_j^{(t)}$
15. end for
16. until $\hat{\beta}$ converges
17. Update $z_i$ and $w_i$ by using the current estimates $\hat{\beta}_j^{(t)}$
18. Update $P_j, j = 1, \ldots, p$
19. $t \leftarrow t + 1$
20. until $\hat{\beta}$ converges
$n \to \infty$. A result when $p \to \infty$ as $n \to \infty$ might be attainable but is out of the content of this paper.

**Lemma 3.1:** Assume $\lambda_n / \sqrt{n} \to \lambda_0 \neq 0$, $\eta_n / \sqrt{n} \to \eta_0 \neq 0$ and $C = \lim_{n \to \infty} (\frac{1}{n} \sum_{i=1}^{n} x_i x_i^T)$. Then there is $V_n(u) \overset{d}{\to} V(u)$, where

$$V_n(u) = \sum_{i=1}^{n} [(\epsilon_i - u^T x_i / \sqrt{n})^2 - \epsilon_i] + \lambda_n \sum_{j=1}^{p} (|\beta_j + u_j / \sqrt{n}| - |\beta_j|)$$

$$+ \frac{\eta_n}{2(d-1)} \sum_{s=1}^{p-d+1} \sum_{k,j \in W_s, k<j} \zeta_{k,j} [(|\beta_k + u_k / \sqrt{n}| - |\beta_j + u_j / \sqrt{n}|)^2$$

$$- (|\beta_k| - |\beta_j|)^2]$$,

with $u = (u_1, \ldots, u_p)^T$ and

$$V(u) = -2u^T w + u^T C u + \lambda_0 \sum_{j=1}^{p} [u_j \text{sgn}(\beta_j) I(\beta_j \neq 0) + |u_j| I(\beta_j = 0)]$$

$$+ \frac{\eta_0}{d-1} \sum_{s=1}^{p-d+1} \sum_{k,j \in W_s, k<j} \frac{|C_{kj}|}{\sqrt{C_{kk} C_{jj}}} \{(|\beta_k| - |\beta_j|)$$

$$* [(u_k \text{sgn}(\beta_k) I(\beta_k \neq 0) + |u_k| I(\beta_k = 0))$$

$$- (u_j \text{sgn}(\beta_j) I(\beta_j \neq 0) + |u_j| I(\beta_j = 0))\}$$,

where $w \sim N(0, \sigma^2 C)$.

**Theorem 3.1:** Let

$$J = \{j = 1, \ldots, p | \beta_j = 0, \sum_{s=j-d+1}^{j} \sum_{k \in W_s, k \neq j} \frac{|C_{kj}|}{\sqrt{C_{kk} C_{jj}}}|\beta_k| > 0\},$$

define

$$\eta_0 = \min_{j \in J} \frac{(d-1)\lambda_0}{\sum_{s=j-d+1}^{j} \sum_{k \in W_s, k \neq j} \frac{|C_{kj}|}{\sqrt{C_{kk} C_{jj}}}|\beta_k|},$$

then $\hat{\beta}^{(n)} \overset{d}{\to} \arg\min V(u)$ while $0 < \eta_0 \leq \eta_0.$
Corollary 3.1: If \( \max \{ \lambda, \eta \} = o(\sqrt{n}) \), then \( \hat{\beta}^{(n)} \) is a \( \sqrt{n} \)-consistent estimator of \( \beta \).

The proof of Lamma 3.1, Theorem 3.1 and Corollary 3.1 are discussed in Appendix.

3.2 A Two-stage Method: MW-Ridge

3.2.1 Motivation

Smoothing the coefficients of neighboring features where exist strong correlations, the moving-window method is supposed to select a strong covariate and its neighboring covariates which are highly correlated with it. We apply the moving-window regression on a simulated dataset in order to illustrate this effect. There are 20 samples and 50 predictors. The feature matrix \( X \) is randomly generated, with \( \text{Corr}(X_k, X_l) = 0.4^{|k-l|} \). The coefficients \( \beta \) are all zero, except for \((\beta_{11}, \ldots, \beta_{15}) = (3, 2, 3, -3, 2)\). We apply three methods on such dataset: LASSO, moving-window regression with \( d = 2 \) (MW-2), and moving-window regression with \( d = 4 \) (MW-4). The coefficient paths for the three methods are shown in Fig 3.1. In this experiment, the tuning parameter \( \eta \) is as large as \( \lambda \). Moving-window regression detected more true positives than LASSO. For instance, with \( \lambda \) as three, there are two true positives with LASSO (in red and green), and four true positives with MW-2. We also find the moving-window regression might include more false positives, especially when \( \lambda \) is small, as a price of the increment in true positive findings. In the next subsection, we propose moving-window ridge (MW-Ridge) in order to reduce the number of noisy
findings.

3.2.2 Method

With the smoothing penalty, a regression model can result in a larger model size so it is more likely to detect true positives. However, as a price of more positive finding, the model might include more false positives. Two-stage variable selection methods have been developed to remove the false positive noises. Relaxed LASSO (24) is a widely used two-stage method. It can outperform LASSO by reducing the number of false positives and providing higher predictive accuracy. In the first stage, LASSO is applied to perform variable selection. In the second stage, LASSO is performed again on the predictors selected in the first stage. Since LASSO is applied twice, there provides a simpler model. Other two-stage methods include relaxed elastic net (25) which applies elastic net twice, and generalized Dantzig selector (17) which applies Dantzig selector twice.

In this subsection, a two-stage method is discussed. Similar as relaxed LASSO, in the first stage, a candidate set of predictors is chosen out by applying moving-window regression. In the second stage, ridge regression is applied on the candidate set for prediction and coefficient estimation.

Definition 3.1: The two-stage method is defined as: find $\hat{\beta}^{\lambda,\eta,\phi}$ which minimizes:

$$
\frac{1}{2n} \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} X_{ij}\beta \cdot 1_{\Omega_{\lambda,\eta}})^2 + \frac{\phi}{2} \sum_{k=1}^{p} \beta_k^2.
$$

Here $\Omega_{\lambda,\eta} \subseteq \{1, 2, \ldots, p\}$ is the set of the index of $\beta$ which has nonzero coefficient
Figure 3.1. The coefficients paths for three methods. From top to bottom: 1) LASSO; 2) MW-2; 3) MW-4. The paths for the five true positives are shown in different colors. The paths in pink are for the coefficients of the false positives.
estimate under the moving window regression with tuning parameters $\lambda$ and $\eta$. $1_{\Omega_{\lambda,\eta}}$ is the indicator function on $\Omega_{\lambda,\eta}$ such that for all $j \in \{1, 2, \ldots, p\}$,

$$\{\beta \cdot 1_{\Omega_{\lambda,\eta}}\}_j = \begin{cases} 0, & j \notin \Omega_{\lambda,\eta} \\ \beta_j, & j \in \Omega_{\lambda,\eta} \end{cases}$$

Only the detected variables in $\Omega_{\lambda,\eta}$ are covered by the ridge regression. The tuning parameters $\lambda$ and $\eta$ control the feature selection part, while $\phi$ controls the estimation of coefficients. Ridge regression might alter the values of coefficients without forcing any coefficient to be zero. Thus, all the variables in $\Omega_{\lambda,\eta}$ will be remained in the final model. Such method can be proposed as follows:

**Step 1.** Select three tuning parameters, $\lambda$, $\eta$ and $\phi$.

**Step 2.** Fit the moving window regression model to response $y$, using all the predictors $X_1, X_2, \ldots, X_p$, by using the tuning parameters $\lambda$ and $\eta$.

**Step 3.** Identify the $q$ variables with nonzero coefficients as $X^*_1, \ldots, X^*_q$.

**Step 4.** Fit a ridge regression to $y$, by using predictors $X^*_1, \ldots, X^*_q$ and tuning parameter $\phi$.

We call such method 'Moving-Window Ridge' (MW-Ridge).

### 3.3 Simulation

#### 3.3.1 Simulation 0: Experiment on GAW 16 RA Data

First, the simulation experiment is conducted on GAW 16 RA data. 2000 genotypes are randomly selected from chromosome 6. The 2000 genotypes are consecutive in order to maintain the sequential structure. 200 subjects are randomly selected for training dataset and another 200 subjects are randomly selected for test
Table 3.1. Median value of true positive, median value of model size, mean(standard deviation) of PPV and TPR, over 100 replicates.

<table>
<thead>
<tr>
<th></th>
<th>True Positive</th>
<th>Model Size</th>
<th>PPV</th>
<th>TPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>LASSO</td>
<td>10</td>
<td>48</td>
<td>0.212(0.069)</td>
<td>0.364(0.055)</td>
</tr>
<tr>
<td>net</td>
<td>12</td>
<td>55</td>
<td>0.218(0.074)</td>
<td>0.454(0.075)</td>
</tr>
<tr>
<td>relaxo</td>
<td>8</td>
<td>18</td>
<td>0.536(0.266)</td>
<td>0.319(0.061)</td>
</tr>
<tr>
<td>SL</td>
<td>22</td>
<td>90.5</td>
<td>0.218(0.056)</td>
<td>0.771(0.167)</td>
</tr>
<tr>
<td>MW-R2</td>
<td>22</td>
<td>23</td>
<td>0.927(0.049)</td>
<td>0.818(0.073)</td>
</tr>
</tbody>
</table>

The genotypes are standardized such that $\sum x_{ij} = 0$ and $\sum x_{ij}^2 = 200$. 

The trait $y$ is generated from the linear model:

$$y_i = \mathbf{x}_i^T \mathbf{\beta} + \epsilon_i, i = 1, \ldots, 200.$$ 

(8)

where $\mathbf{x}_i \in \mathbb{R}^{2000}$ is a vector of SNP scores of subject $i$, $\mathbf{\beta}$ is the vector of genetic effects for these SNPs. $\epsilon_i$ is the random residual sampled from a normal distribution with mean 0 and variance 1. The entries of $\mathbf{\beta}$ are all 0, except that $(\beta_{1506}, \ldots, \beta_{1512}) = (0.7,-0.5,0.1,-0.5,0.3,-0.6,0.2)$ and $(\beta_{1514}, \ldots, \beta_{1532}) = (0.35,-0.4,0.2,-0.1,0.25,0.3,0.4,-0.4,-0.15,0.3,-0.4,0.4,-0.5,0.2,-0.3,0.16,0.36,-0.2,0.1)$. The number of truly non-zero $\beta$s is 26.

As discussed in Chapter 1, fused LASSO is not appropriate for genome-wide association studies. Thus, these five methods are compared: LASSO, elastic net (net), relaxed LASSO (relaxo), SLasso (SL) and MW-Ridge with $d = 2$ (MW-R2). The simulation results based on 100 replicates are shown in Table 3.1.

First of all, there is a performance improvement in the value of TP after using
SLasso. The median value of TP is increased by 120%, compared to LASSO (from 10 to 22) and by 83%, compared to elastic net (from 12 to 22). The TPR increases by 112% and 70%, respectively, compared to LASSO (from 0.364 to 0.771) and elastic net (from 0.454 to 0.771).

Secondly, SLasso results in larger model size. In spite of the improvement in TP findings, the PPV has little improvement, compared to LASSO and elastic net. The PPV of SLasso is the same as the PPV of elastic net (0.218).

MW-Ridge performs best among all the five methods. The TP value of MW-R2 is the same as the TP value of SLasso. The TPR of MW-R2 is even a little higher than the TPR of SLasso (from 0.771 to 0.818). Furthermore, MW-R2 results in simpler models with model size as only 23. The PPV increased by 337% and 73%, respectively, compared to LASSO (from 0.212 to 0.927) and relaxed LASSO (from 0.536 to 0.927).

Next, the simulation study is conducted in four different settings: continuous outcome with uncorrelated predictors, continuous outcome with correlated predictors, binary outcome with uncorrelated predictors, and binary outcome with correlated predictors. SLasso and MW-Ridge are implemented on both linear and logistic models. The purpose is to explore how these two methods behave in different aspects and to compare them with other methods.
3.3.2 Simulation 1: Continuous Outcome and Uncorrelated Predictors

The first simulation study focuses on continuous outcome and uncorrelated predictors. The simulated training dataset and test dataset both contain $n = 100$ observations, with different number of predictors: $p = 50$, $p = 100$ or $p = 200$. There are 100 different simulation experiments generated. The methods are fitted using the training data, while the tuning parameters are chosen so as to minimize the deviance on the test datasets. The response $y$ is generated as:

$$y_i = x_i^T \beta + \epsilon_i,$$

with $x_i$ being the $i$th row of $X$ and $\epsilon_i \sim N(0, \sigma^2)$ for $i \in \{1, \ldots, n\}$. The standard deviation of the error term, $\sigma$, is varied. We test the performances of different methods under $\sigma = 1, 4, 7, 10$. The larger value of $\sigma$ indicates a lower signal-to-noise ratio. $X_j$ for $j \in \{1, \ldots, p\}$ refers to the $j$th column of $X$, is standardized such that: $\sum_{i=1}^{n} X_{ij} = 0$ and $\sum_{i=1}^{n} X_{ij}^2 = n$. The true coefficients of $\beta$ are all zero, except that $(\beta_{11}, \ldots, \beta_{17}) = (0.6, 1.8, 2.5, 1.6, 0.8, -0.5, -0.5)$, $(\beta_{23}, \ldots, \beta_{27}) = (-0.4, 0.5, 1.2, 2, 0.75)$, and $(\beta_{41}, \ldots, \beta_{46}) = (-0.3, 0.8, 2, 1.2, 0.5, -1)$. There are totally 18 true positives. The methods which are compared are as follows:

- **LASSO**: classical LASSO implemented using 'glmnet' in R package 'glmnet'
- **elastic net**: elastic net regression implemented using 'glmnet'
- **fused LASSO**: fused LASSO implemented by 'fusedLASSO1d' function in R package 'genLASSO'
- **SLasso**: moving-window regression with window size $d = 2$
- relaxed LASSO: relaxed LASSO implemented using ‘relaxo’
- MW-Ridge2 (MW-R2): moving-window ridge method with window size \(d = 2\)
- MW-Ridge4 (MW-R4): moving-window ridge method with window size \(d = 4\)

A grid search is applied in the tuning parameter selection for SLasso, MW-R2 and MW-R4. The maximum value of \(\lambda\) is set as \(\lambda_{max} = \max_j |\sum_{i=1}^{n} X_{ij}y_i|/n\), and the grid of \(\lambda\) is set as \((0.01\lambda_{max}, 0.02\lambda_{max}, \ldots, \lambda_{max})\). A new parameter \(\alpha\) is introduced as the proportion of \(\lambda\) in the sum of tuning parameters: \(\alpha = \lambda/(\lambda + \eta)\). The grid of \(\alpha\) is \((0.1, 0.2, \ldots, 0.9)\). The tuning parameter \(\eta\) can be derived from \(\alpha\) and \(\lambda\). For SLasso, the performance of every combination of \(\lambda\) and \(\eta\) is evaluated, and the pair of parameters which can minimize the predictive mean squared error on the test dataset is chosen out. For MW-R2 and MW-R4, there is a third parameter \(\phi\), for the ridge regression. In each pair of \(\lambda\) and \(\eta\), ‘glmnet’ is use to choose \(\phi\). After evaluating the performances of all combinations of tuning parameters, we choose the parameters which work best on the test dataset.

Table 3.2 shows the predictive mean squared error over \(\sigma^2\) (PMSE/\(\sigma^2\)), true positive rate (TPR) and false detective rate (FDR) for these methods with different \(\sigma\) values. Here TPR and FDR are defined as:

\[
TPR = TP/(TP + FN),
\]

\[
FDR = FP/(TP + FP).
\]

Here TP stands for true positive, FN for false negative and FP for false positive. We see PMSE/\(\sigma^2\) increases as \(p\) increases. The performances of MW-R2 and MW-R4
are close. They have close values in PMSE/\sigma^2, TPR and FDR, for different \( p \) and \( \sigma \) values. Relaxed LASSO results in lower PMSE than LASSO, and similarly, MW-R2 results in lower PMSE than SLasso. Fused LASSO obtains lower PMSE than SLasso. Using smoothing penalty, fused LASSO and SLasso both attain apparently higher TPR and FDR than other methods, which indicates that these two methods might select more features including both true positives and false positives. MW-R2, MW-R4 and relaxed LASSO result in significantly lower FDR, compared with other methods. As a price, these three methods result in lower TPR. With a high value of \( \sigma \) (low signal-noise ratio), such price might be eliminated. As \( \sigma = 10 \) and \( p = 200 \), there are very strong noises. MW-R2 and MW-R4 result in higher TPR than all other methods except fused LASSO. Besides, there are 17 out of 100 experiments which select zero features for LASSO and 16 out of 100 for relaxed LASSO, while MW-R2 always select one or more features.

Fig 3.2 shows the median number of false positives (FP) over 100 simulation experiments, from top to bottom: 1) for \( p = 50 \), 2) for \( p = 100 \) and 3) for \( p = 200 \). MW-R2, MW-R4 and relaxed LASSO all result in much fewer FP than other methods. There is an apparent discrepancy between the number of FP for these three methods and the number of FP for other methods. Such discrepancy is larger at lower values of \( \sigma \) (higher signal-noise ratio). Fused LASSO and SLasso result in much more FP than other methods. As \( \sigma \) increases, the gaps between these two methods and other methods becomes wider, which indicates that these two methods are still able to provide findings even there exist strong noises. Such behavior should be driven by
Table 3.2. Mean of predictive mean squared error over $\sigma^2$ (PMSE/$\sigma^2$), true positive rate (TPR), and false detective rate (FDR), for seven different methods with different $\sigma$ values, using 100 simulations each with $p = 50$, $p = 100$, and $p = 200$. Continuous outcome, uncorrelated predictors.

<table>
<thead>
<tr>
<th>Method</th>
<th>$\sigma = 1$</th>
<th></th>
<th></th>
<th></th>
<th>$\sigma = 4$</th>
<th></th>
<th></th>
<th></th>
<th>$\sigma = 7$</th>
<th></th>
<th></th>
<th></th>
<th>$\sigma = 10$</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>p=50 PMSE</td>
<td>TPR</td>
<td>FDR</td>
<td>p=100 PMSE</td>
<td>TPR</td>
<td>FDR</td>
<td>p=200 PMSE</td>
<td>TPR</td>
<td>FDR</td>
<td>p=50 PMSE</td>
<td>TPR</td>
<td>FDR</td>
<td>p=100 PMSE</td>
<td>TPR</td>
<td>FDR</td>
<td>p=200 PMSE</td>
</tr>
<tr>
<td>LASSO</td>
<td>1.37</td>
<td>0.999</td>
<td>0.484</td>
<td>1.73</td>
<td>0.99</td>
<td>0.620</td>
<td>2.40</td>
<td>0.973</td>
<td>0.699</td>
<td>1.20</td>
<td>0.631</td>
<td>0.448</td>
<td>1.28</td>
<td>0.471</td>
<td>0.611</td>
<td>1.36</td>
</tr>
<tr>
<td>net</td>
<td>1.37</td>
<td>0.999</td>
<td>0.492</td>
<td>1.74</td>
<td>0.989</td>
<td>0.622</td>
<td>2.45</td>
<td>0.974</td>
<td>0.704</td>
<td>1.18</td>
<td>0.706</td>
<td>0.497</td>
<td>1.27</td>
<td>0.555</td>
<td>0.661</td>
<td>1.35</td>
</tr>
<tr>
<td>FusedL</td>
<td>1.32</td>
<td>0.999</td>
<td>0.510</td>
<td>1.55</td>
<td>0.993</td>
<td>0.654</td>
<td>1.81</td>
<td>0.980</td>
<td>0.738</td>
<td>1.12</td>
<td>0.803</td>
<td>0.502</td>
<td>1.19</td>
<td>0.741</td>
<td>0.666</td>
<td>1.23</td>
</tr>
<tr>
<td>SL</td>
<td>1.36</td>
<td>0.999</td>
<td>0.490</td>
<td>1.70</td>
<td>0.992</td>
<td>0.632</td>
<td>2.25</td>
<td>0.984</td>
<td>0.714</td>
<td>1.17</td>
<td>0.811</td>
<td>0.503</td>
<td>1.26</td>
<td>0.684</td>
<td>0.681</td>
<td>1.30</td>
</tr>
<tr>
<td>relaxo</td>
<td>1.24</td>
<td>0.991</td>
<td>0.202</td>
<td>1.40</td>
<td>0.961</td>
<td>0.309</td>
<td>1.92</td>
<td>0.930</td>
<td>0.435</td>
<td>1.16</td>
<td>0.481</td>
<td>0.283</td>
<td>1.27</td>
<td>0.354</td>
<td>0.397</td>
<td>1.32</td>
</tr>
<tr>
<td>MW-R2</td>
<td>1.35</td>
<td>0.979</td>
<td>0.146</td>
<td>1.59</td>
<td>0.956</td>
<td>0.250</td>
<td>2.11</td>
<td>0.917</td>
<td>0.349</td>
<td>1.12</td>
<td>0.534</td>
<td>0.269</td>
<td>1.23</td>
<td>0.456</td>
<td>0.412</td>
<td>1.29</td>
</tr>
<tr>
<td>MW-R4</td>
<td>1.35</td>
<td>0.987</td>
<td>0.202</td>
<td>1.56</td>
<td>0.964</td>
<td>0.294</td>
<td>2.12</td>
<td>0.925</td>
<td>0.408</td>
<td>1.12</td>
<td>0.557</td>
<td>0.279</td>
<td>1.21</td>
<td>0.488</td>
<td>0.427</td>
<td>1.28</td>
</tr>
</tbody>
</table>

1 net, elastic net; FusedL, fused LASSO; SL, SLasso; relaxo, relaxed LASSO; MW-R2, MW-Ridge2; MW-R4, MW-Ridge4.
the smoothing penalties in these two methods.

Fig 3.3 shows the median number of true positives (TP), from top to bottom: 1) for $p = 50$, 2) for $p = 100$ and 3) for $p = 200$. As $\sigma = 1$, all methods attain high TP which are close to 18. There are totally 18 true positives, so all methods are able to choose out all or almost all true signals. The numbers of TP decrease as $\sigma$ increases, for all methods. Fused LASSO and SLasso always result in more TP than other methods, at a price of obtaining more FP as shown in fig 2. Elastic net always results in more TP and more FP than LASSO. As $\sigma = 10$, there are very strong noises and we can see that under different values of $p$, relaxed LASSO obtains less TP than LASSO while MW-R2 and MW-R4 always obtain more TP.

In the context of continuous outcome with uncorrelated predictors, MW-R2 behaves quite similar with MW-R4. Compared with classical LASSO, SLasso tends to select more features, which results in higher TPR and higher FDR. MW-Ridge method can maintain the advantage of SLasso as keeping true signals and eliminating the side-effect by removing noises. Compared with relaxed LASSO, MW-Ridge tends to detect more TP at the price of more FP. Such price might be eliminated in the context of continuous outcome with correlated predictors, as shown in the next subsection.

3.3.3 Simulation 2: Continuous Outcome and Correlated Predictors

Simulation 2 uses a continuous outcome and correlated predictors. Like in Simulation 1, the training dataset and test dataset both contain $n = 100$ observations, with different numbers of predictors: $p = 50$, $p = 100$ or $p = 200$. There are 100
Figure 3.2. Median number of false positives, continuous outcome and uncorrelated predictors, with different values of $\sigma$. From top to bottom: 1) $p = 50$, relaxo covered by MW-R4 before $\sigma = 7$, MW-R2 covered by MW-R4 after $\sigma = 7$; 2) $p = 100$; 3) $p = 200$. 
Figure 3.3. Median number of true positives, continuous outcome and uncorrelated predictors, with different values of $\sigma$. From top to bottom: 1) $p = 50$; 2) $p = 100$; 3) $p = 200$, MW-R2 covered by MW-R4 after $\sigma = 7$, relaxo covered by MW-R2 before $\sigma = 4$. 
different experiments randomly generated. The outcome $y$ is generated as:

$$y_i = x_i^T \beta + \epsilon_i,$$

with $x_i$ being the $i$th row of $X$ and $\epsilon_i \sim N(0, \sigma^2)$ for $i \in \{1, \ldots, n\}$. The standard deviation of the error term, $\sigma$, is varied as $\sigma = 1, 4, 7, 10$. $X_j$ refers to the $j$th column of $X$, are standardized such that: $\sum_{i=1}^{n} X_{ij} = 0$ and $\sum_{i=1}^{n} X_{ij}^2 = n$. The true coefficients $\beta$ is the same as that in Simulation 1. The predictors are correlated as $\text{Corr}(X_k, X_l) = \rho^{|k-l|}$, with $\rho = 0.4$. Like in Simulation 1, the performances of seven methods are explored: LASSO, elastic net, fused LASSO, SLasso, relaxed LASSO, MW-Ridge2 and MW-Ridge4. Table 3.3 shows the simulation results over 100 replicates.

We can see $\text{PMSE}/\sigma^2$ increases as $p$ increases. Like in simulation 1, the performances of MW-R2 and MW-R4 are quite close. Relaxed LASSO results in lower predictive error than LASSO and MW-R2 results in lower predictive error than SLasso. MW-R2 and MW-R4 always obtain lower PMSE than relaxed LASSO except for $p = 50$ and $\sigma = 1$. Fused LASSO obtains lower PMSE than SLasso. TPR decreases as $\sigma$ increases and $p$ increases for all methods. Compared with relaxed LASSO, as $\sigma = 1, 4, 7$, MW-R2 results in lower FDR and higher TPR; as $\sigma = 10$, MW-R2 results in higher TPR with a slight increment in FDR. The TPR of relaxed LASSO is apparently lower than LASSO, while that of MW-R2 is close to LASSO or even higher. As $\sigma = 10$ when there exist strong noises, MW-R2 always performs higher TPR than LASSO. Fused LASSO and SLasso always result in the highest values in TPR and FDR among all methods, indicating that they tend to select more true signals as well.
Table 3.3. Mean of predictive mean squared error over $\sigma^2$ (PMSE/$\sigma^2$), true positive rate (TPR), and false detective rate (FDR), for seven different methods with different $\sigma$ values, using 100 simulations each with $p = 50$, $p = 100$, and $p = 200$. Continuous outcome, correlated predictors.

<table>
<thead>
<tr>
<th>Method</th>
<th>$\sigma = 1$</th>
<th>$\sigma = 4$</th>
<th>$\sigma = 7$</th>
<th>$\sigma = 10$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>p=50 PMSE $\sigma^2$</td>
<td>p=100 PMSE $\sigma^2$</td>
<td>p=200 PMSE $\sigma^2$</td>
<td></td>
</tr>
<tr>
<td>LASSO</td>
<td>1.37 0.992 0.464</td>
<td>1.66 0.974 0.650</td>
<td>2.10 0.942 0.672</td>
<td></td>
</tr>
<tr>
<td>net</td>
<td>1.36 0.992 0.471</td>
<td>1.66 0.975 0.596</td>
<td>2.11 0.942 0.676</td>
<td></td>
</tr>
<tr>
<td>FusedL</td>
<td>1.32 0.996 0.504</td>
<td>1.53 0.986 0.596</td>
<td>1.81 0.967 0.723</td>
<td></td>
</tr>
<tr>
<td>SL</td>
<td>1.33 0.995 0.461</td>
<td>1.59 0.986 0.596</td>
<td>1.93 0.971 0.682</td>
<td></td>
</tr>
<tr>
<td>relaxo</td>
<td>1.27 0.968 0.237</td>
<td>1.46 0.919 0.294</td>
<td>1.80 0.879 0.368</td>
<td></td>
</tr>
<tr>
<td>MW-R2</td>
<td>1.29 0.981 0.189</td>
<td>1.44 0.957 0.217</td>
<td>1.64 0.948 0.286</td>
<td></td>
</tr>
<tr>
<td>MW-R4</td>
<td>1.34 0.983 0.195</td>
<td>1.47 0.967 0.243</td>
<td>1.64 0.940 0.306</td>
<td></td>
</tr>
<tr>
<td>LASSO</td>
<td>1.22 0.791 0.435</td>
<td>1.35 0.669 0.550</td>
<td>1.47 0.612 0.622</td>
<td></td>
</tr>
<tr>
<td>net</td>
<td>1.22 0.817 0.459</td>
<td>1.34 0.696 0.582</td>
<td>1.46 0.636 0.644</td>
<td></td>
</tr>
<tr>
<td>FusedL</td>
<td>1.16 0.876 0.491</td>
<td>1.24 0.794 0.641</td>
<td>1.31 0.759 0.709</td>
<td></td>
</tr>
<tr>
<td>SL</td>
<td>1.17 0.901 0.492</td>
<td>1.27 0.868 0.641</td>
<td>1.39 0.810 0.714</td>
<td></td>
</tr>
<tr>
<td>relaxo</td>
<td>1.16 0.603 0.178</td>
<td>1.23 0.526 0.206</td>
<td>1.30 0.431 0.125</td>
<td></td>
</tr>
<tr>
<td>MW-R2</td>
<td>1.10 0.756 0.159</td>
<td>1.14 0.677 0.150</td>
<td>1.19 0.593 0.106</td>
<td></td>
</tr>
<tr>
<td>MW-R4</td>
<td>1.16 0.767 0.175</td>
<td>1.14 0.694 0.172</td>
<td>1.20 0.614 0.137</td>
<td></td>
</tr>
<tr>
<td>LASSO</td>
<td>1.16 0.630 0.418</td>
<td>1.23 0.518 0.569</td>
<td>1.31 0.457 0.631</td>
<td></td>
</tr>
<tr>
<td>net</td>
<td>1.15 0.714 0.464</td>
<td>1.22 0.576 0.598</td>
<td>1.30 0.508 0.669</td>
<td></td>
</tr>
<tr>
<td>FusedL</td>
<td>1.09 0.808 0.482</td>
<td>1.14 0.734 0.642</td>
<td>1.19 0.686 0.705</td>
<td></td>
</tr>
<tr>
<td>SL</td>
<td>1.11 0.853 0.495</td>
<td>1.17 0.797 0.668</td>
<td>1.25 0.742 0.740</td>
<td></td>
</tr>
<tr>
<td>relaxo</td>
<td>1.11 0.462 0.169</td>
<td>1.17 0.387 0.241</td>
<td>1.21 0.328 0.217</td>
<td></td>
</tr>
<tr>
<td>MW-R2</td>
<td>1.05 0.587 0.148</td>
<td>1.10 0.536 0.197</td>
<td>1.15 0.432 0.183</td>
<td></td>
</tr>
<tr>
<td>MW-R4</td>
<td>1.05 0.618 0.172</td>
<td>1.10 0.556 0.210</td>
<td>1.15 0.455 0.208</td>
<td></td>
</tr>
<tr>
<td>LASSO</td>
<td>1.11 0.497 0.415</td>
<td>1.17 0.409 0.588</td>
<td>1.23 0.329 0.627</td>
<td></td>
</tr>
<tr>
<td>net</td>
<td>1.10 0.606 0.460</td>
<td>1.15 0.517 0.632</td>
<td>1.21 0.413 0.675</td>
<td></td>
</tr>
<tr>
<td>FusedL</td>
<td>1.06 0.743 0.485</td>
<td>1.10 0.663 0.637</td>
<td>1.14 0.647 0.714</td>
<td></td>
</tr>
<tr>
<td>SL</td>
<td>1.08 0.788 0.496</td>
<td>1.12 0.727 0.681</td>
<td>1.18 0.691 0.771</td>
<td></td>
</tr>
<tr>
<td>relaxo</td>
<td>1.08 0.358 0.201</td>
<td>1.13 0.278 0.280</td>
<td>1.18 0.224 0.290</td>
<td></td>
</tr>
<tr>
<td>MW-R2</td>
<td>1.04 0.510 0.187</td>
<td>1.08 0.467 0.317</td>
<td>1.13 0.389 0.314</td>
<td></td>
</tr>
<tr>
<td>MW-R4</td>
<td>1.04 0.517 0.203</td>
<td>1.08 0.486 0.313</td>
<td>1.13 0.412 0.318</td>
<td></td>
</tr>
</tbody>
</table>

1 net, elastic net; FusedL, fused LASSO; SL, SLasso; relaxo, relaxed LASSO; MW-R2, MW-Ridge2; MW-R4, MW-Ridge4.
as more noises.

Fig 3.4 shows the median number of FP, from top to bottom: 1) for $p = 50$, 2) for $p = 100$ and 3) for $p = 200$. Relaxed LASSO, MW-R2 and MW-R4 detect apparently less FP than all other methods. Fused LASSO and SLasso detect more FP than others. With a lower value of $\sigma$ (higher signal-noise ratio), fused LASSO results in more FP than SLasso. With a higher value of $\sigma$ (lower signal-noise ratio), fused LASSO results in less FP than SLasso.

Fig 3.5 shows the median number of TP, from top to bottom: 1) for $p = 50$, 2) for $p = 100$ and 3) for $p = 200$. For all methods, the number of TP decreases as $\sigma$ increases. MW-R2 and MW-R4 over perform relaxed LASSO by detecting more TP. The graph of MW-R2, MW-R4 and LASSO are very close, indicating that the numbers of TP discovered by MW-Ridge are close to that by LASSO. This is different with Simulation 1 in which MW-Ridge methods detect less TP than LASSO when $\sigma$ has lower values. Like in Simulation 1, elastic net always detects more TP and more FP than LASSO. SLasso results in highest values in TP while fused LASSO results in the second highest.

In the context of continuous outcome with correlated predictors, MW-Ridge can over perform LASSO and relaxed LASSO. MW-R2 and MW-R4 result in close performances again. Compared with relaxed LASSO, MW-R2 results in less PMSE and detects more TP with little increment in FP. Compared with LASSO, MW-R2 achieves close numbers in TP, and significantly reduces the numbers in FP.
Figure 3.4. Median number of false positives, continuous outcome and correlated predictors, with different values of $\sigma$. From top to bottom: 1) $p = 50$, MW-R2 covered by MW-R4 before $\sigma = 7$, relaxo covered by MW-R2 after $\sigma = 7$; 2) $p = 100$; 3) $p = 200$. 
Figure 3.5. Median number of true positives, continuous outcome and correlated predictors, with different values of $\sigma$. From top to bottom: 1) $p = 50$, MW-R2 fully covered by MW-R4; 2) $p = 100$, LASSO covered by MW-R2 before $\sigma = 7$, net covered by MW-R4 after $\sigma = 7$; 3) $p = 200$, LASSO covered by MW-R2 before $\sigma = 7$. 
3.3.4 Simulation 3: Binary Outcome and Uncorrelated Predictors

Simulation 3 focuses on binary outcome with uncorrelated predictors. The feature matrix $X$ is generated from independent standard normal distribution. Sample size $n$ is fixed at 100 and predictor number $p$ is varied and takes the following values: 50, 100, 200. The coefficient $\beta$ is the same as that in Simulation 1. The number of true positives is 18. The response vector $y$ is generated as

$$y_i \sim \text{Bernoulli}(cp_i),$$

where $p_i$ refers to the logit:

$$p_i = \frac{1}{1 + e^{-x_i^T \beta}}$$

The parameter $c$ controls the signal-noise ratio in the logistic regression context. Here $c$ is varied and takes the following values: 0.25, 0.5, 1, 2. The higher value of $c$ indicates a higher signal-noise ratio. Feature matrix $X$ is standardized such that: $\sum_{i=1}^{n} X_{ij} = 0$ and $\sum_{i=1}^{n} X_{ij}^2 = n$, for $j = 1, \ldots, p$. The performances are evaluated based on a separate test dataset with $n = 100$. These methods are compared:

- LASSO: classical LASSO implemented using 'glmnet' in R package 'glmnet'
- elastic net: elastic net regression implemented using 'glmnet'
- svm: support vector machine implemented using R package 'rminer'
- random forest: binary outcome random forests implemented using R package 'randomForest'
- SLasso: moving-window regression with window size $d$ as two
- MW-Ridge2 (MW-R2): moving-window ridge method with window size $d = 2$
• MW-Ridge4 (MW-R4): moving-window ridge method with window size $d = 4$

As mentioned in Simulation 1, grid search is applied to select tuning parameters for SLasso, MW-R2 and MW-R4. The R package ‘rminer’ provides the ‘Importance’ function, which performs as a variable selection approach for support vector machine. For random forest, there is no variable selection method available.

Table 3.4 shows the mean predictive error rate (PER), TPR and FDR for the above methods. For all methods, PER increases as $p$ increases and $c$ decreases. Random forest is significantly over-performed by other methods in prediction. There is an apparent gap between the PER of random forest with other methods. MW-R2 and MW-R4 result in the lowest predictive error rates among all methods. Like in the context of continuous outcome, MW-R2 and MW-R4 perform closely in both prediction and variable selection. The FDR for MW-R2 is apparently lower than that for LASSO. The gap in FDR between these two methods is enlarged as higher $c$ value (stronger signals) and eliminated as lower $c$ value (weaker signals). For MW-R2, there is a slight decrement in TPR as a price of lower FDR, compared with LASSO.

Fig 3.6 shows the median number of FP, from top to bottom: 1) for $p = 50$, 2) for $p = 100$ and 3) for $p = 200$. As $p$ increases, SLasso results in more FP than other methods. The performances of MW-R2, MW-R4 and svm are similar and their graphs cross over each other’s. These three methods result in much lower numbers of FP than other methods.

Fig 3.7 shows the median number of true positives, from top to bottom: 1) for $p = 50$, 2) for $p = 100$ and 3) for $p = 200$. The number of TP decreases as $c$ decreases
Table 3.4. Mean of predictive error rate (PER), true positive rate (TPR), and false
detective rate (FDR), for eight different methods and for different $c$ values, using 100
simulations each with $p = 50$, $p = 100$, and $p = 200$. Binary outcome, uncorrelated
predictors.

<table>
<thead>
<tr>
<th>Method</th>
<th>$p=50$ PER</th>
<th>$p=50$ TPR</th>
<th>$p=50$ FDR</th>
<th>$p=100$ PER</th>
<th>$p=100$ TPR</th>
<th>$p=100$ FDR</th>
<th>$p=200$ PER</th>
<th>$p=200$ TPR</th>
<th>$p=200$ FDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>LASSO</td>
<td>0.136 0.799 0.458</td>
<td>0.166 0.689 0.566</td>
<td>0.190 0.561 0.639</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>net</td>
<td>0.129 0.801 0.439</td>
<td>0.160 0.668 0.542</td>
<td>0.187 0.531 0.584</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>svm</td>
<td>0.166 0.644 0.352</td>
<td>0.196 0.544 0.409</td>
<td>0.217 0.406 0.381</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>forest</td>
<td>0.296 - -</td>
<td>0.337 - -</td>
<td>0.375 - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SL</td>
<td>0.121 0.861 0.475</td>
<td>0.144 0.753 0.578</td>
<td>0.166 0.681 0.686</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MW-R2</td>
<td>0.119 0.707 0.289</td>
<td>0.145 0.582 0.346</td>
<td>0.168 0.443 0.363</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MW-R4</td>
<td>0.120 0.705 0.292</td>
<td>0.144 0.574 0.330</td>
<td>0.168 0.460 0.381</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$c = 2$

<table>
<thead>
<tr>
<th>Method</th>
<th>$p=50$ PER</th>
<th>$p=50$ TPR</th>
<th>$p=50$ FDR</th>
<th>$p=100$ PER</th>
<th>$p=100$ TPR</th>
<th>$p=100$ FDR</th>
<th>$p=200$ PER</th>
<th>$p=200$ TPR</th>
<th>$p=200$ FDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>LASSO</td>
<td>0.175 0.814 0.470</td>
<td>0.197 0.648 0.595</td>
<td>0.220 0.550 0.669</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>net</td>
<td>0.165 0.81 0.470</td>
<td>0.190 0.637 0.579</td>
<td>0.215 0.563 0.674</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>svm</td>
<td>0.193 0.659 0.376</td>
<td>0.216 0.456 0.388</td>
<td>0.241 0.447 0.486</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>forest</td>
<td>0.311 - -</td>
<td>0.345 - -</td>
<td>0.379 - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SL</td>
<td>0.155 0.859 0.453</td>
<td>0.177 0.709 0.596</td>
<td>0.196 0.648 0.712</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MW-R2</td>
<td>0.153 0.715 0.314</td>
<td>0.172 0.537 0.385</td>
<td>0.193 0.494 0.439</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MW-R4</td>
<td>0.152 0.749 0.339</td>
<td>0.172 0.541 0.381</td>
<td>0.193 0.502 0.453</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$c = 1$

<table>
<thead>
<tr>
<th>Method</th>
<th>$p=50$ PER</th>
<th>$p=50$ TPR</th>
<th>$p=50$ FDR</th>
<th>$p=100$ PER</th>
<th>$p=100$ TPR</th>
<th>$p=100$ FDR</th>
<th>$p=200$ PER</th>
<th>$p=200$ TPR</th>
<th>$p=200$ FDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>LASSO</td>
<td>0.244 0.725 0.481</td>
<td>0.270 0.553 0.595</td>
<td>0.284 0.452 0.642</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>net</td>
<td>0.235 0.696 0.455</td>
<td>0.256 0.589 0.598</td>
<td>0.274 0.471 0.655</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>svm</td>
<td>0.255 0.603 0.388</td>
<td>0.272 0.521 0.489</td>
<td>0.290 0.452 0.599</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>forest</td>
<td>0.350 - -</td>
<td>0.380 - -</td>
<td>0.418 - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SL</td>
<td>0.231 0.728 0.461</td>
<td>0.249 0.626 0.605</td>
<td>0.264 0.545 0.688</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MW-R2</td>
<td>0.219 0.661 0.386</td>
<td>0.237 0.536 0.479</td>
<td>0.265 0.424 0.495</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MW-R4</td>
<td>0.218 0.667 0.385</td>
<td>0.234 0.541 0.491</td>
<td>0.253 0.423 0.502</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$c = 0.5$

<table>
<thead>
<tr>
<th>Method</th>
<th>$p=50$ PER</th>
<th>$p=50$ TPR</th>
<th>$p=50$ FDR</th>
<th>$p=100$ PER</th>
<th>$p=100$ TPR</th>
<th>$p=100$ FDR</th>
<th>$p=200$ PER</th>
<th>$p=200$ TPR</th>
<th>$p=200$ FDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>LASSO</td>
<td>0.344 0.671 0.550</td>
<td>0.361 0.468 0.679</td>
<td>0.380 0.321 0.758</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>net</td>
<td>0.333 0.648 0.513</td>
<td>0.347 0.525 0.666</td>
<td>0.367 0.383 0.763</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>svm</td>
<td>0.339 0.528 0.479</td>
<td>0.350 0.503 0.638</td>
<td>0.364 0.374 0.754</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>forest</td>
<td>0.421 - -</td>
<td>0.447 - -</td>
<td>0.456 - -</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SL</td>
<td>0.333 0.666 0.530</td>
<td>0.346 0.534 0.690</td>
<td>0.365 0.401 0.785</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MW-R2</td>
<td>0.311 0.612 0.485</td>
<td>0.321 0.451 0.595</td>
<td>0.336 0.360 0.731</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MW-R4</td>
<td>0.308 0.592 0.474</td>
<td>0.319 0.466 0.599</td>
<td>0.336 0.368 0.727</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1 net, elastic net; svm, support vector machine; forest, random forest; SL, SLasso; MW-R2, MW-Ridge2; MW-R4, MW-Ridge4.
Figure 3.6. Median number of false positives, binary outcome and uncorrelated predictors, with different values of $c$. From top to bottom: 1) $p = 50$, MW-R2 covered by MW-R4 after $c = 0.5$; 2) $p = 100$; 3) $p = 200$. 
and $p$ increases. Similar as the context of continuous outcome, SLasso always results in a highest number of TP. The numbers of TP for MW-R2 and MW-R4 are close, higher than that for svm. As a price of detecting fewer noises, MW-R2 detects less true signals than LASSO and elastic net, especially with higher $c$ values (higher signal-noise ratio). With $c = 0.25$, there are strong noises and MW-R2 achieves a close number of TP and lower FP, compared with LASSO.

In the context of binary outcome and uncorrelated predictors, MW-R2 and MW-R4 result in close performances. MW-Ridge can generally overperform support vector machine in prediction and feature selection. MW-Ridge might detect less FP than LASSO, at the price of detecting less TP. Such price might be eliminated in the context of binary outcome and correlated predictors.

### 3.3.5 Simulation 4: Binary Outcome and Correlated Predictors

Simulation 4 studies binary outcome with correlated predictors. The feature matrix $X$ is correlated as $\text{Corr}(X_k, X_l) = \rho^{|k-l|}$, with $\rho = 0.4$. Sample size $n$ is fixed at 100 and predictor number $p$ is varied as 50, 100, 200. The coefficients $\beta$ is the same as that in Simulation 1. The response vector $y$ is generated as $y_i \sim \text{Bernoulli}(cp_i)$. Parameter $c$ is varied and takes these values: 0.25, 0.5, 1, 2. The performances of LASSO, elastic net, support vector machine, random forest, SLasso, MW-R2 and MW-R4 are explored and compared.

Table 3.5 shows the mean of PER, TPR and FDR of the above methods. Similar to Simulation 3, random forest results in highest predictive error. MW-
Figure 3.7. Median number of true positives, binary outcome and uncorrelated predictors, with different values of $c$. From top to bottom: 1) $p = 50$, LASSO fully covered by net, net covered by SL after $c = 0.5$. MW-R2 covered by MW-R4 after $c = 0.5$; 2) $p = 100$, net covered by SL after $c = 0.5$; 3) $p = 200$, LASSO covered by net before $c = 1$, MW-R2 fully covered by MW-R4.
R2 and MW-R4 result in close PER, which are lower than other methods. SLasso produces higher TPR and higher FDR. MW-Ridge can keep TPR as high as that of LASSO, and apparently decrease FDR at the same time. Compared with support vector machine, MW-R2 and MW-R4 result in better predictive performances, higher TPR and lower FDR.

Fig 3.8 shows the median number of FP, from top to bottom: 1) for $p = 50$, 2) for $p = 100$ and 3) for $p = 200$. SLasso detects highest number of FP among all methods. Generally speaking, svm detects less FP than LASSO and elastic net. However, as $c = 0.25$ (low signal-noise ratio), svm detects no less FP than LASSO. The graphs of MW-R2 and MW-R4 are almost overlapping. There is a clear gap between the graph of MW-R2 and svm. This is different from Simulation 3 in which the graphs of MW-R2 and svm are quite close. Such phenomenon indicates that MW-Ridge can detect apparently less false signals than svm and all the other methods, in the context of correlated predictors.

Fig 3.9 shows the median number of TP, from top to bottom: 1) for $p = 50$, 2) for $p = 100$ and 3) for $p = 200$. SLasso always detects highest number of TP. The graphs of MW-R2 and MW-R4 are almost overlapping. As $p = 50$, MW-R2 behaves similar with svm. As $p = 100$ or 200, MW-R2 detects more TP than svm. Under the context of uncorrelated predictors, as shown in Fig 3.7, LASSO detects more TP than MW-R2 when $c$ has higher values (stronger signals). For correlated predictors, it is different and MW-R2 results in similar numbers of TP with LASSO, under both higher and lower values of $c$. 
Table 3.5. Mean of predictive error rate (PER), true positive rate (TPR), and false
detective rate (FDR), for eight different methods and for different $c$ values, using 100
simulations each with $p = 50$, $p = 100$, and $p = 200$. Binary outcome, correlated
predictors.

<table>
<thead>
<tr>
<th>Method</th>
<th>$c = 2$</th>
<th>$c = 1$</th>
<th>$c = 0.5$</th>
<th>$c = 0.25$</th>
</tr>
</thead>
<tbody>
<tr>
<td>LASSO</td>
<td>0.096</td>
<td>0.133</td>
<td>0.192</td>
<td>0.282</td>
</tr>
<tr>
<td>net</td>
<td>0.085</td>
<td>0.122</td>
<td>0.193</td>
<td>0.269</td>
</tr>
<tr>
<td>svm</td>
<td>0.117</td>
<td>0.149</td>
<td>0.261</td>
<td>0.282</td>
</tr>
<tr>
<td>forest</td>
<td>0.197</td>
<td>0.221</td>
<td>0.261</td>
<td>-</td>
</tr>
<tr>
<td>SL</td>
<td>0.074</td>
<td>0.113</td>
<td>0.171</td>
<td>0.263</td>
</tr>
<tr>
<td>MW-R2</td>
<td>0.069</td>
<td>0.101</td>
<td>0.155</td>
<td>0.247</td>
</tr>
<tr>
<td>MW-R4</td>
<td>0.070</td>
<td>0.103</td>
<td>0.155</td>
<td>0.245</td>
</tr>
</tbody>
</table>

1 net, elastic net; svm, support vector machine; forest, random forest; SL, SLasso; MW-R2, MW-Ridge2; MW-R4, MW-Ridge4.
Figure 3.8. Median number of false positives, binary outcome and correlated predictors, with different values of $c$. From top to bottom: 1) $p = 50$, MW-R2 covered by MW-R4 before $c = 1$; 2) $p = 100$, MW-R2 fully covered by MW-R4; 3) $p = 200$, MW-R2 covered by MW-R4 between $c = 1$ and $c = 0.5$. 
In the context of binary outcome with correlated predictors, MW-Ridge can over performs LASSO and svm. First, MW-Ridge results in lower PER. Secondly, MW-Ridge performs better in denoising (detect less FP) and feature detection (detect more TP). In the aspect of denoising, svm might remove noises if there is a high value of $c$. However, as noises become stronger, such denoising effect will be eliminated. In comparison, MW-Ridge always achieves small FP numbers, in both high and low signal-noise ratio settings. In the aspect of feature detection, svm discovers less TP than LASSO, which is a side-effect of removing noises. Without suffering such side-effect, MW-Ridge detects close numbers of TP as LASSO. In a sum, by applying SLasso in the first step, MW-Ridge enjoys more positive findings. By using the two-stage setting, MW-Ridge is able to remove unrelated noises and remain true features.

### 3.4 Application

We examined the GAW 16 RA data in order to compare the performances of SLasso and MW-Ridge with other methods. The predictors are the genotypes on chromosome 6. There are 34385 SNPs. The response variable $y$ is binary, where $y = 0$ for controls and $y = 1$ for cases. The data consisted of 2062 samples. We randomly chose 1031 samples as the training set and the other 1031 samples as the validation set.

Considering MW-R2 and MW-R4 behaved very similarly in the simulation study, we didn’t apply MW-R4 and the methods applied to this real dataset were LASSO, elastic net, SLasso and MW-R2. A grid search was applied to search for
Figure 3.9. Median number of true positives, binary outcome and correlated predictors, with different values of $c$. From top to bottom: 1) $p = 50$, svm covered by MW-R2 after $c = 1$, net covered by MW-R4 after $c = 0.5$; 2) $p = 100$, LASSO and MW-R2 both fully covered by MW-R4; 3) $p = 200$, MW-R2 fully covered by MW-R4, LASSO covered by MW-R4 before $c = 0.5$, net covered by MW-R4 before $c = 1$. 
Table 3.6. Results of GAW 16 RA data.

<table>
<thead>
<tr>
<th>method</th>
<th># misclassified</th>
<th>model size</th>
</tr>
</thead>
<tbody>
<tr>
<td>LASSO</td>
<td>226</td>
<td>247</td>
</tr>
<tr>
<td>elastic net</td>
<td>211</td>
<td>1903</td>
</tr>
<tr>
<td>SLasso</td>
<td>226</td>
<td>399</td>
</tr>
<tr>
<td>MW-R2</td>
<td>222</td>
<td>177</td>
</tr>
</tbody>
</table>

tuning parameters. The grid for $\lambda$ was $(0.01\lambda_{max}, 0.02\lambda_{max}, \ldots, \lambda_{max})$, while the grid for $\alpha$ was $(0.05, 0.1, \ldots, 0.95)$. The results are shown in Table 3.6.

As discussed in Chapter 2, there existed considerable correlation effects in the GAW 16 data. Therefore it was reasonable to take such correlation effects into consideration by applying the moving-window penalty. Elastic net resulted in the least number of misclassification with a model size of 1903 predictors. LASSO and SLasso both resulted in 226 misclassified number. MW-R2 over performed LASSO and SLasso by resulting in less number of misclassification and smaller model size. MW-R2 resulted in 222 number of misclassification, with only 177 features.

There may still be some advantages of MW-Ridge even in such a problem with a small sample size and relatively much larger feature size. In the aspect of feature selection, LASSO might find predictors that are highly correlated with the true factors but unable to distinguish the causative factors under the existence of collinearity. Elastic net might achieve high classification accuracy by finding both the true features and highly correlated factors. MW-R2 achieved its high accuracy as well as using much fewer predictors than elastic net, with the potential of finding more true factors than LASSO.
3.5 Appendix

Proof for Lemma 1:

Consider each term in $V_n$. Since $p$ is fixed and $n \to \infty$, by the central limit theorem (CLT) and the assumption that $\{\epsilon_i\}$ are independent and $\lim_{n \to \infty} (\frac{1}{n} \sum_{i=1}^{n} x_i x_i^T) = C$, for the first term,

$$\sum_{i=1}^{n} [(\epsilon_i - u^T x_i / \sqrt{n})^2 - \epsilon_i^2] = -2u^T \sum_{i=1}^{n} \epsilon_i x_i / \sqrt{n} + u^T (\sum_{i=1}^{n} x_i x_i^T / n) u$$

$$\xrightarrow{d} -2u^T w + u^T Cu,$$

where $w \sim N(0, \sigma^2 C)$. For the second term, it is obvious that

$$\lambda_n \sum_{j=1}^{p} (|\beta_j + u_j / \sqrt{n}| - |\beta_j|) \to \lambda_0 \sum_{j=1}^{p} [u_j \text{sgn}(\beta_j) I(\beta_j \neq 0) + |u_j| I(\beta_j = 0)].$$

For the third term, one can obtain that

$$\eta_n \frac{p-d+1}{2(d-1)} \sum_{s=1}^{p-d+1} \sum_{k,j \in W_s, k < j} \zeta_{k,j} [(|\beta_k + u_k / \sqrt{n}| - |\beta_j + u_j / \sqrt{n}|)^2 - (|\beta_k| - |\beta_j|)^2]$$

$$\to \frac{\eta_0}{d-1} \sum_{s=1}^{p-d+1} \sum_{k,j \in W_s, k < j} \frac{|C_{kj}|}{\sqrt{C_{kk}C_{jj}}} \{(|\beta_k| - |\beta_j|)$$

$$*[(u_k \text{sgn}(\beta_k) I(\beta_k \neq 0) + |u_k| I(\beta_k = 0))$$

$$- (u_j \text{sgn}(\beta_j) I(\beta_j \neq 0) + |u_j| I(\beta_j = 0))]\}.$$

Thus, there is $V_n(u) \xrightarrow{d} V(u)$.

Proof for Theorem 1:

Consider function $V(u)$. Vector $u$ is in $\mathbb{R}^p$ and the hyperplanes $\{u \in \mathbb{R}^p | u_j = 0\}$ separate $\mathbb{R}^p$ into $2^p$ subspaces, while $V(u)$ is convex in each of these subspaces.
However, the convexity is not guaranteed on the boundaries \( \{ u \in R^p | u_j = 0, j = 1, \ldots, p \} \). Suppose \( A_k, A_l \) are two neighboring subspaces and \( A_k \cap A_l \subseteq \{ u \in R^p | u_j = 0 \} \). Since \( V(u) \) is a piece-wise quadratic function, thus \( V(u) \) is convex (2) in \( A_k \cup A_l \) if \( \frac{\partial V(u)}{\partial u} |_{A_k} = \frac{\partial V(u)}{\partial u} |_{A_l} \). Such result is obvious if \( \beta_j \neq 0 \). If \( \beta_j = 0 \), consider

\[
\frac{\partial V(u)}{\partial u} |_{U} = -2w_j + 2 \sum_{i=1}^{p} C_{ji} u_i - \lambda_0 + \frac{\eta_0}{d - 1} \sum_{j=s=d+1}^{j} \sum_{k \in W_s, k \neq j} \frac{|C_{kj}|}{\sqrt{C_{kk} C_{jj}}} |\beta_k|,
\]

\[
|\beta_k| = \frac{w_j + 2 \sum_{i=1}^{p} C_{ji} u_i + \lambda_0 - \frac{\eta_0}{d - 1} \sum_{j=s=d+1}^{j} \sum_{k \in W_s, k \neq j} \frac{|C_{kj}|}{\sqrt{C_{kk} C_{jj}}} |\beta_k|.
\]

There needs \( \frac{\partial V(u)}{\partial u} |_{U} = \frac{\partial V(u)}{\partial u} |_{V} \). One can obtain

\[
\eta_0 = \eta_j^*, \eta_j^* = \frac{(d - 1) \lambda_0}{\sum_{j=s=d+1}^{j} \sum_{k \in W_s, k \neq j} \frac{|C_{kj}|}{\sqrt{C_{kk} C_{jj}}} |\beta_k|}.
\]

The non-convex term in \( V(u) \) is controlled by \( \eta_0 \) and a smaller value of \( \eta_0 \) results in a smaller effect of the non-convexity. Thus \( V(u) \) is convex in \( A_k \cup A_l \) if \( \eta_0 \leq \eta_j^* \).

Consider all \( j = 1, \ldots, p \), one can conclude that \( V(u) \) is convex over \( R^p \) if

\[
\eta_0 \leq \eta_j^*, \eta_j^* = \frac{(d - 1) \lambda_0}{\sum_{j=s=d+1}^{j} \sum_{k \in W_s, k \neq j} \frac{|C_{kj}|}{\sqrt{C_{kk} C_{jj}}} |\beta_k|},
\]

for \( j \in J : \{ j = 1, \ldots, p | \beta_j = 0, \sum_{s=d+1}^{j} \sum_{k \in W_s, k \neq j} \frac{|C_{kj}|}{\sqrt{C_{kk} C_{jj}}} |\beta_k| > 0 \} \). Consider Lemma 1. Define \( W_n(u) = V(u) + 2u^T w - 2u^T \sum_{i=1}^{n} \epsilon_i x_i / \sqrt{n} \). Assume \( \eta_0 \leq \eta_\star \).

\( W_n(u) \) is convex and \( V(u) \) has a unique minimum, it follows (13) that \( \text{argmin} W_n(u) \xrightarrow{d} \text{argmin} V(u) \). Consider \( \phi_n(u) = V_n(u) - W_n(u) \). \( \phi_n(u) \) converges to zero pointwisely.

Thus \( \phi_n(u) \) uniformly converges to zero in any compact set. To any \( \epsilon > 0 \), there exists \( M > 0 \) and compact set \( D \in R^p \), s.t. for \( n > M \), \( \Pr(\text{argmin} W_n(u) \in D) > 1 - \epsilon \).

Since \( \phi_n(u) \) uniformly converges to zero in \( D \), there is \( \text{argmin} V_n(u) - \text{argmin} W_n(u) \to 0 \) uniformly.
0 in $D$, which results in $\Pr(\arg\min V_n(u) - \arg\min W_n(u) \to 0) > 1 - \epsilon$. Thus $\arg\min V_n(u) - \arg\min W_n(u) \xrightarrow{p} 0$ and there is $\arg\min V_n(u) \xrightarrow{d} \arg\min V(u)$. Let $\hat{\beta}^{(n)}$ minimize $f(\beta)$ where $f$ is the objective function defined in (4), it follows that $\sqrt{n}(\hat{\beta}^{(n)} - \beta)$ minimizes $V_n$, thus $\hat{\beta}^{(n)} \xrightarrow{d} \arg\min V(u)$.

**Proof for Corollary 1:**

Condition $\max \{\lambda, \eta\} = o(\sqrt{n})$ implies that $\lambda_0 = \eta_0 = 0$ and therefore

$$V(u) = -2u^T w + u^T Cu.$$  

Then it is obvious that $C^{-1}w$ minimizes $V(u)$. The proof is complete since $C^{-1}w \sim N(0, \sigma^2C^{-1})$. 
CHAPTER 4
MULTIVARIATE RESPONSE DATA ANALYSIS

This chapter focuses on analyzing data with multiple response variables. In classical statistics with a small number of predictors, data with multiple responses can be accommodated by using multivariate analysis of variance (MANOVA) and multivariate analysis of covariance (MANCOVA) (27). However, these methods cannot solve high dimensional problems. It is possible to accommodate high dimensional data by applying penalized regression and considering each response variable individually and then combining the feature selection results. However, such an approach ignores the correlation among response variables. We are interested in learning multiple correlated responses jointly by analyzing data from all of the response variables at the same time. When response variables are highly correlated, it is greatly advantageous to borrow the information in the data from other related responses to learn each response variable more effectively.

In this chapter, Section 4.1 discusses how to transform multi-response data into uni-response data following the same distribution (21). In Section 4.2, SLasso and MW-Ridge are compared with classical LASSO for simulated multivariate response data.

4.1 Method

Assume a sample of \( n \) subjects, each represented by a \( p \)-dimensional feature vector and a \( M \)-dimensional response vector. Denote \( Y = (y^1, \ldots, y^M) \in R^{n \times M} \)
as the response variables and $X$ as the $n \times p$ covariate matrix. For each of the $M$ responses, we assume a linear model:

$$y^m = X\beta^m + \epsilon^m, \quad m = 1, \ldots, M, \tag{1}$$

where $\beta^m = (\beta^m_1, \ldots, \beta^m_p)' \in \mathbb{R}^p$ is the regression coefficient corresponding to the $m$th response variable. The residue vectors $\epsilon^m$, $m = 1, \ldots, M$ are correlated and the covariance matrix is $\Sigma$. We center $y^m$'s and $X$ such that $\sum_{i=1}^n y^m_i = 0$ and $\sum_{i=1}^n X_{ij} = 0$, $j = 1, \ldots, p$, and consider the model without intercepts. Let $B = (\beta^1, \ldots, \beta^M) \in \mathbb{R}^{p \times M}$ denote the coefficients matrix of all $M$ response variables. By using LASSO, $\hat{B}^{\text{LASSO}}$ is obtained by solving the following optimization problem:

$$\hat{B}^{\text{LASSO}} = \arg\min_B \frac{1}{2n} \| Y - XB \|_F^2 + \lambda \| B \|_1, \tag{2}$$

where $\| \cdot \|_F$ denotes the matrix Frobenius norm, $\| \cdot \|_1$ denotes the entry-wise $L_1$ norm, and $\lambda$ is the tuning parameter which controls the sparsity level. The LASSO approach is actually considering each response variable individually, without offering any mechanism for a joint coefficients estimation for the multiple responses.

We propose the penalization method under a joint modeling framework. First, multi-response data is transformed to uni-response data (21). We use the symbols $Y^u$, $X^u$ and $B^u$ for the uni-response problem. Let $y_i$ be the length-$M$ vector of response variables for the $i$th subject, and $Y^u = (y'_1, \ldots, y'_n)'$. For the $i$th subject, the covariates are formed as $X_i = (x_{i1}I_M, \ldots, x_{ip}I_M)$. Then $X^u = (X'_1, \ldots, X'_n)'$. The regression coefficient vector is $B^u = (\beta'_1, \ldots, \beta'_p)$ where $\beta_j = (\beta^1_j, \ldots, \beta^M_j)'$. Now consider a toy example to better illustrate the transformation process. Consider a
dataset with $M = 2$ and assume only the first three predictors are related to the responses. The coefficients are $\beta^1 = (1, 2, 3, 0, \ldots, 0)'$ and $\beta^2 = (4, 5, 6, 0, \ldots, 0)'$. Then correspondently, the regression coefficient for the uni-response data is $B^u = (1, 4, 2, 5, 3, 6, 0, \ldots, 0)'$.

Now consider the distribution of the uni-response problem. Suppose each subject is independent, then the correlation among response variables are driven from the residuals. Let $E^u = (\epsilon_1', \ldots, \epsilon_n')'$ where $\epsilon_i$ is the length-$M$ vector of residuals for the $i$th subject. Thus the covariance matrix of $E^u$ is $\Sigma^u \in R^{Mn \times Mn}$. In $\Sigma^u$, the blocks in the diagonal are $\Sigma$’s, while all other entries are zero. Now we can propose the uni-response problem as:

$$Y^u = X^u B^u + E^u, E^u \sim N(0, \Sigma^u).$$

(3)

The least square loss function for the transformed data can be written as

$$(Y^u - X^u B^u)'(\Sigma^u)^{-1}(Y^u - X^u B^u).$$

In order to take the correlation effect into consideration, we apply smoothed LASSO (SLasso) by adding a moving-window penalty to the loss function. Then $\hat{B}^{SL}$ is obtained by solving the optimization problem:

$$\hat{B}^{SL} = \arg\min_{B^u} (Y^u - X^u B^u)'(\Sigma^u)^{-1}(Y^u - X^u B^u) + \lambda ||B^u||_1 + \frac{\eta}{2} \sum_{s=1}^{M_p-1} \sum_{k,j \in W_s, k < j} \zeta_{k,j} (|B^u_k| - |B^u_j|)^2.$$  (4)

Besides, we apply MW-Ridge with window size $d = 2$ to the uni-response data and set $\hat{B}^{MW-R2}$ as the coefficient estimate.
By applying the smoothing penalty on $B^a$, the following two pairs of coefficients: $\beta^1_j$ and $\beta^2_j$, $\beta^2_j$ and $\beta^1_{j+1}$, are smoothed. The correlation between response variables encourages $(|\beta^1_j| - |\beta^2_j|)^2$ to be smoothed towards zero. Furthermore, $(|\beta^2_j| - |\beta^1_{j+1}|)^2$ are encouraged to be zero, enabling $\beta^1$ and $\beta^2$ to be smoothed together.

4.2 Simulation

In the simulation, we compare the performances of these three approaches: LASSO, SLasso and MW-Ridge. In MW-Ridge, the window size is set as two. In both of the training and test dataset, there are 400 subjects and the number of predictors is set as 1000 or 4000. For each subject, we simulate two response variables. The two residual vectors are both under the standard normal distribution. The correlation between the two residual vectors is set as $\rho = 0.1, 0.5, 0.9$, representing weak, moderate or strong correlations. Thus the two response variables are correlated through the residuals. For each response variable, there are fifteen non-zero predictor variables, which can be grouped into three clusters. In each cluster, the correlation between two predictors $X_k$ and $X_j$ is $0.2^{|k-j|}$. The correlation among predictors not associated with responses is set in the same way. Response-associated and noisy predictors are independent. For the first response variable, the regression coefficients are all zero, except for $(\beta^1_{21}, \ldots, \beta^1_{26}) = (1, 2, 2, 1, 2, 1)$, $(\beta^1_{41}, \ldots, \beta^1_{44}) = (1, 1, 1, 1)$, and $(\beta^1_{61}, \ldots, \beta^1_{65}) = (1, 2, 2, 1, 1)$. For the second response variable, the coefficients are all zero except for $(\beta^2_{21}, \ldots, \beta^2_{26}) = (0.5, 0.5, 1, 1, 0.5, 0.5)$, $(\beta^2_{41}, \ldots, \beta^2_{44}) = (0.5, 0.5, 0.5, 0.5)$, and $(\beta^2_{61}, \ldots, \beta^2_{65}) = (1, 1, 0.5, 0.5, 1)$. The two re-
response variables depend on the same set of predictors. For the first response variable, the signal-noise ratio is higher, while for the second response variable the signal-noise ratio is lower. Grid search approaches are applied to choose tuning parameters. The parameters are those which are able to minimize the predictive squared error on the test dataset. There are 50 replicates when the number of predictors is 1000, and 20 replicates when the number is 4000.

Results of simulation studies are summarized in Table 4.1 and Table 4.2. Table 4.1 shows the results for the first response variable and Table 4.2 shows the results for the second one. SPC stands for specificity, while FDR stands for false detective rate. SPC and FDR are defined as follows:

\[ SPC = \frac{TN}{TN + FP}, \]
\[ FDR = \frac{FP}{TP + FP}. \]

Here TN, TP, and FP stand for true negative, true positive, and false positive, respectively. Since the noises are not too strong thus all of the three methods are able to detect all or almost all true positives. In the context of moderate correlations, SLasso tends to select more variables than classical LASSO. In the context of high correlation, SLasso is able to select fewer variables and remove some unrelated noises. MW-R2 always results in the best performances by significantly reducing false detective rates. Especially for the second response variable (weaker signals and stronger noises), the model sizes are very close to the number of true positives and the false detective rates are close to zero.
Table 4.1. Simulation results for the first response variable in mean(standard deviation) with all matched non-zero $\beta$s.

<table>
<thead>
<tr>
<th>method</th>
<th>$\rho$</th>
<th>True Positive</th>
<th>Model Size</th>
<th>SPC</th>
<th>FDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>p=1000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LASSO</td>
<td>0.1</td>
<td>15(0.0)</td>
<td>74.30(18.32)</td>
<td>0.940(0.019)</td>
<td>0.784(0.062)</td>
</tr>
<tr>
<td>SLasso</td>
<td>0.1</td>
<td>15(0.0)</td>
<td>89.24(44.80)</td>
<td>0.925(0.045)</td>
<td>0.773(0.126)</td>
</tr>
<tr>
<td>MW-R2</td>
<td>0.1</td>
<td>15(0.0)</td>
<td>15.02(0.14)</td>
<td>1.000(1e-4)</td>
<td>0.001(0.009)</td>
</tr>
<tr>
<td>LASSO</td>
<td>0.5</td>
<td>15(0.0)</td>
<td>72.74(19.42)</td>
<td>0.941(0.020)</td>
<td>0.779(0.059)</td>
</tr>
<tr>
<td>SLasso</td>
<td>0.5</td>
<td>15(0.0)</td>
<td>130.42(19.32)</td>
<td>0.883(0.020)</td>
<td>0.881(0.026)</td>
</tr>
<tr>
<td>MW-R2</td>
<td>0.5</td>
<td>15(0.0)</td>
<td>15.62(0.92)</td>
<td>1.000(9e-4)</td>
<td>0.037(0.052)</td>
</tr>
<tr>
<td>LASSO</td>
<td>0.9</td>
<td>15(0.0)</td>
<td>70.54(16.84)</td>
<td>0.944(0.017)</td>
<td>0.774(0.060)</td>
</tr>
<tr>
<td>SLasso</td>
<td>0.9</td>
<td>15(0.0)</td>
<td>57.44(9.86)</td>
<td>0.957(0.010)</td>
<td>0.731(0.047)</td>
</tr>
<tr>
<td>MW-R2</td>
<td>0.9</td>
<td>14.98(0.14)</td>
<td>23.12(8.32)</td>
<td>0.992(0.008)</td>
<td>0.291(0.184)</td>
</tr>
<tr>
<td>p=4000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LASSO</td>
<td>0.1</td>
<td>15(0.0)</td>
<td>96.0(21.80)</td>
<td>0.980(0.005)</td>
<td>0.837(0.035)</td>
</tr>
<tr>
<td>SLasso</td>
<td>0.1</td>
<td>15(0.0)</td>
<td>95.6(46.91)</td>
<td>0.980(0.012)</td>
<td>0.818(0.060)</td>
</tr>
<tr>
<td>MW-R2</td>
<td>0.1</td>
<td>15(0.0)</td>
<td>15.1(0.31)</td>
<td>1.000(8e-5)</td>
<td>0.006(0.019)</td>
</tr>
<tr>
<td>LASSO</td>
<td>0.5</td>
<td>15(0.0)</td>
<td>107.3(40.81)</td>
<td>0.977(0.010)</td>
<td>0.844(0.051)</td>
</tr>
<tr>
<td>SLasso</td>
<td>0.5</td>
<td>15(0.0)</td>
<td>199.3(15.24)</td>
<td>0.954(0.004)</td>
<td>0.924(0.006)</td>
</tr>
<tr>
<td>MW-R2</td>
<td>0.5</td>
<td>15(0.0)</td>
<td>17.3(2.20)</td>
<td>1.000(5e-4)</td>
<td>0.120(0.105)</td>
</tr>
<tr>
<td>LASSO</td>
<td>0.9</td>
<td>15(0.0)</td>
<td>95.6(28.39)</td>
<td>0.980(0.007)</td>
<td>0.833(0.039)</td>
</tr>
<tr>
<td>SLasso</td>
<td>0.9</td>
<td>15(0.0)</td>
<td>94.6(12.49)</td>
<td>0.980(0.003)</td>
<td>0.839(0.021)</td>
</tr>
<tr>
<td>MW-R2</td>
<td>0.9</td>
<td>14.8(0.41)</td>
<td>28.95(15.39)</td>
<td>0.996(0.004)</td>
<td>0.372(0.258)</td>
</tr>
</tbody>
</table>
Table 4.2. Simulation results for the second response variable in mean(standard deviation) with all matched non-zero $\beta$s.

<table>
<thead>
<tr>
<th>method</th>
<th>$\rho$</th>
<th>True Positive</th>
<th>Model Size</th>
<th>SPC</th>
<th>FDR</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>p=1000</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LASSO 0.1</td>
<td>15(0.0)</td>
<td>72.42(20.82)</td>
<td>0.942(0.02)</td>
<td>0.778(0.056)</td>
<td></td>
</tr>
<tr>
<td>SLasso 0.1</td>
<td>15(0.0)</td>
<td>84.42(44.72)</td>
<td>0.930(0.05)</td>
<td>0.752(0.140)</td>
<td></td>
</tr>
<tr>
<td>MW-R2 0.1</td>
<td>15(0.0)</td>
<td>15(0.0)</td>
<td>1(0.0)</td>
<td>0(0.0)</td>
<td></td>
</tr>
<tr>
<td>LASSO 0.5</td>
<td>15(0.0)</td>
<td>74.12(21.16)</td>
<td>0.940(0.021)</td>
<td>0.781(0.064)</td>
<td></td>
</tr>
<tr>
<td>SLasso 0.5</td>
<td>15(0.0)</td>
<td>125.14(19.01)</td>
<td>0.888(0.019)</td>
<td>0.876(0.030)</td>
<td></td>
</tr>
<tr>
<td>MW-R2 0.5</td>
<td>15(0.0)</td>
<td>15.08(0.27)</td>
<td>1.000(2e-4)</td>
<td>0.005(0.017)</td>
<td></td>
</tr>
<tr>
<td>LASSO 0.9</td>
<td>15(0.0)</td>
<td>72.92(19.60)</td>
<td>0.941(0.020)</td>
<td>0.780(0.060)</td>
<td></td>
</tr>
<tr>
<td>SLasso 0.9</td>
<td>14.72(0.614)</td>
<td>48.08(9.39)</td>
<td>0.956(0.004)</td>
<td>0.921(0.006)</td>
<td></td>
</tr>
<tr>
<td>MW-R2 0.9</td>
<td>14.92(0.342)</td>
<td>16.1(5.48)</td>
<td>0.999(0.006)</td>
<td>0.029(0.133)</td>
<td></td>
</tr>
<tr>
<td><strong>p=4000</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LASSO 0.1</td>
<td>15(0.0)</td>
<td>104.3(33.43)</td>
<td>0.978(0.008)</td>
<td>0.842(0.048)</td>
<td></td>
</tr>
<tr>
<td>SLasso 0.1</td>
<td>15(0.0)</td>
<td>91.85(45.80)</td>
<td>0.981(0.011)</td>
<td>0.809(0.065)</td>
<td></td>
</tr>
<tr>
<td>MW-R2 0.1</td>
<td>15(0.0)</td>
<td>15.05(0.22)</td>
<td>1.000(6e-5)</td>
<td>0.003(0.014)</td>
<td></td>
</tr>
<tr>
<td>LASSO 0.5</td>
<td>15(0.0)</td>
<td>101.1(26.60)</td>
<td>0.978(0.007)</td>
<td>0.841(0.044)</td>
<td></td>
</tr>
<tr>
<td>SLasso 0.5</td>
<td>15(0.0)</td>
<td>190.85(15.17)</td>
<td>0.956(0.004)</td>
<td>0.921(0.006)</td>
<td></td>
</tr>
<tr>
<td>MW-R2 0.5</td>
<td>14.95(0.224)</td>
<td>16.05(2.18)</td>
<td>0.999(0.001)</td>
<td>0.055(0.106)</td>
<td></td>
</tr>
<tr>
<td>LASSO 0.9</td>
<td>15(0.0)</td>
<td>98.95(21.16)</td>
<td>0.979(0.005)</td>
<td>0.843(0.030)</td>
<td></td>
</tr>
<tr>
<td>SLasso 0.9</td>
<td>14.6(0.598)</td>
<td>73.8(11.07)</td>
<td>0.985(0.003)</td>
<td>0.798(0.029)</td>
<td></td>
</tr>
<tr>
<td>MW-R2 0.9</td>
<td>14.6(0.503)</td>
<td>17.35(12.17)</td>
<td>0.999(0.003)</td>
<td>0.040(0.178)</td>
<td></td>
</tr>
</tbody>
</table>
Table 4.3. Values of $\alpha$’s for SLasso and MW-R2 in mean(standard deviation) with all matched non-zero $\beta$s.

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>p=1000</th>
<th>p=4000</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\alpha$</td>
<td>$\alpha$</td>
</tr>
<tr>
<td>0.1</td>
<td>0.644(0.298)</td>
<td>0.615(0.260)</td>
</tr>
<tr>
<td>0.5</td>
<td>0.786(0.154)</td>
<td>0.85(0.079)</td>
</tr>
<tr>
<td>0.9</td>
<td>0.85(0.079)</td>
<td>0.615(0.260)</td>
</tr>
</tbody>
</table>

Consider $\alpha = \lambda / (\lambda + \eta)$, the $\alpha$’s values are illustrated in Table 4.3. A lower value of $\alpha$ indicates a stronger effect of the moving-window penalty compared with the $L_1$ penalty. MW-R2 always results in a lower value of $\alpha$ than SLasso, which helps to provide clusters of non-zero coefficients and significantly remove noisy predictors.

With the proposed approaches, it is assumed that the multiple responses have the same set of important predictors. Such assumption is not always true in practice. In order to get a more comprehensive understanding of the proposed methods, we conducted a simulation in which the two sets of important predictors are partially matched. We consider the simulation setting where 40% of the important predictors are not matched. Let $\beta^1$ be the same value, while for the second response variable, the coefficients are all zero except for $(\beta^2_{19}, \ldots, \beta^2_{24}) = (0.5, 0.5, 1, 1, 0.5, 0.5)$, $(\beta^2_{43}, \ldots, \beta^2_{46}) = (0.5, 0.5, 0.5, 0.5)$, and $(\beta^2_{59}, \ldots, \beta^2_{63}) = (1, 1, 0.5, 0.5, 1)$. The simulation results are shown in Table 4.4 and Table 4.5.

Even when the regression coefficients are not matched, MW-R2 still results in the best performances among the three approaches. All of the three approaches are able to detect all true positives. However, LASSO and SLasso also detect many...
Table 4.4. Simulation results for the first response variable in mean(standard deviation). 40% of the regression coefficients are not matched.

<table>
<thead>
<tr>
<th>method</th>
<th>$\rho$</th>
<th>True Positive</th>
<th>Model Size</th>
<th>SPC</th>
<th>FDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>p=1000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LASSO</td>
<td>0.1</td>
<td>15(0.0)</td>
<td>69.3(17.17)</td>
<td>0.945(0.017)</td>
<td>0.770(0.056)</td>
</tr>
<tr>
<td>SLasso</td>
<td>0.1</td>
<td>15(0.0)</td>
<td>91.14(43.58)</td>
<td>0.923(0.044)</td>
<td>0.778(0.129)</td>
</tr>
<tr>
<td>MW-R2</td>
<td>0.1</td>
<td>15(0.0)</td>
<td>15.08(0.27)</td>
<td>1.000(2e-4)</td>
<td>0.005(0.017)</td>
</tr>
<tr>
<td>LASSO</td>
<td>0.5</td>
<td>15(0.0)</td>
<td>73.38(22.10)</td>
<td>0.941(0.022)</td>
<td>0.778(0.063)</td>
</tr>
<tr>
<td>SLasso</td>
<td>0.5</td>
<td>15(0.0)</td>
<td>119.7(17.82)</td>
<td>0.894(0.018)</td>
<td>0.871(0.028)</td>
</tr>
<tr>
<td>MW-R2</td>
<td>0.5</td>
<td>15(0.0)</td>
<td>16.06(1.42)</td>
<td>0.999(0.001)</td>
<td>0.060(0.073)</td>
</tr>
<tr>
<td>LASSO</td>
<td>0.9</td>
<td>15(0.0)</td>
<td>71.92(20.19)</td>
<td>0.942(0.021)</td>
<td>0.775(0.062)</td>
</tr>
<tr>
<td>SLasso</td>
<td>0.9</td>
<td>15(0.0)</td>
<td>43.08(7.99)</td>
<td>0.971(0.008)</td>
<td>0.639(0.073)</td>
</tr>
<tr>
<td>MW-R2</td>
<td>0.9</td>
<td>15(0.0)</td>
<td>23.58(3.60)</td>
<td>0.991(0.004)</td>
<td>0.352(0.078)</td>
</tr>
<tr>
<td>p=4000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LASSO</td>
<td>0.1</td>
<td>15(0.0)</td>
<td>81.0(23.57)</td>
<td>0.983(0.006)</td>
<td>0.800(0.057)</td>
</tr>
<tr>
<td>SLasso</td>
<td>0.1</td>
<td>15(0.0)</td>
<td>87.25(43.15)</td>
<td>0.982(0.011)</td>
<td>0.798(0.070)</td>
</tr>
<tr>
<td>MW-R2</td>
<td>0.1</td>
<td>15(0.0)</td>
<td>15.35(0.67)</td>
<td>1.000(2e-4)</td>
<td>0.021(0.040)</td>
</tr>
<tr>
<td>LASSO</td>
<td>0.5</td>
<td>15(0.0)</td>
<td>81.3(22.75)</td>
<td>0.983(0.006)</td>
<td>0.801(0.062)</td>
</tr>
<tr>
<td>SLasso</td>
<td>0.5</td>
<td>15(0.0)</td>
<td>170.75(40.87)</td>
<td>0.961(0.010)</td>
<td>0.903(0.043)</td>
</tr>
<tr>
<td>MW-R2</td>
<td>0.5</td>
<td>15(0.0)</td>
<td>16.5(1.64)</td>
<td>1.000(4e-4)</td>
<td>0.083(0.081)</td>
</tr>
<tr>
<td>LASSO</td>
<td>0.9</td>
<td>15(0.0)</td>
<td>85.65(20.68)</td>
<td>0.982(0.005)</td>
<td>0.814(0.488)</td>
</tr>
<tr>
<td>SLasso</td>
<td>0.9</td>
<td>15(0.0)</td>
<td>76.75(13.95)</td>
<td>0.984(0.003)</td>
<td>0.798(0.038)</td>
</tr>
<tr>
<td>MW-R2</td>
<td>0.9</td>
<td>15(0.0)</td>
<td>26.75(6.22)</td>
<td>0.997(0.002)</td>
<td>0.414(0.115)</td>
</tr>
</tbody>
</table>
Table 4.5. Simulation results for the second response variable in mean(standard deviation). 40% of the regression coefficients are not matched.

<table>
<thead>
<tr>
<th>method</th>
<th>$\rho$</th>
<th>True Positive</th>
<th>Model Size</th>
<th>SPC</th>
<th>FDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p=1000$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LASSO</td>
<td>0.1</td>
<td>15(0.0)</td>
<td>83.0(22.34)</td>
<td>0.931(0.023)</td>
<td>0.806(0.054)</td>
</tr>
<tr>
<td>SLasso</td>
<td>0.1</td>
<td>15(0.0)</td>
<td>90.46(44.12)</td>
<td>0.923(0.044)</td>
<td>0.772(0.138)</td>
</tr>
<tr>
<td>MW-R2</td>
<td>0.1</td>
<td>15(0.0)</td>
<td>15.18(0.44)</td>
<td>1.000(1e-4)</td>
<td>0.011(0.027)</td>
</tr>
<tr>
<td>LASSO</td>
<td>0.5</td>
<td>15(0.0)</td>
<td>78.52(21.58)</td>
<td>0.936(0.022)</td>
<td>0.795(0.054)</td>
</tr>
<tr>
<td>SLasso</td>
<td>0.5</td>
<td>15(0.0)</td>
<td>116.54(18.67)</td>
<td>0.897(0.019)</td>
<td>0.867(0.032)</td>
</tr>
<tr>
<td>MW-R2</td>
<td>0.5</td>
<td>15(0.0)</td>
<td>15.76(1.25)</td>
<td>0.999(4e-4)</td>
<td>0.043(0.066)</td>
</tr>
<tr>
<td>LASSO</td>
<td>0.9</td>
<td>15(0.0)</td>
<td>79.04(24.31)</td>
<td>0.935(0.025)</td>
<td>0.793(0.062)</td>
</tr>
<tr>
<td>SLasso</td>
<td>0.9</td>
<td>14.26(1.05)</td>
<td>38.86(9.81)</td>
<td>0.975(0.009)</td>
<td>0.614(0.080)</td>
</tr>
<tr>
<td>MW-R2</td>
<td>0.9</td>
<td>15(0.0)</td>
<td>21.46(0.54)</td>
<td>0.993(5e-4)</td>
<td>0.301(0.017)</td>
</tr>
<tr>
<td>$p=4000$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LASSO</td>
<td>0.1</td>
<td>15(0.0)</td>
<td>113.35(21.31)</td>
<td>0.975(0.005)</td>
<td>0.863(0.026)</td>
</tr>
<tr>
<td>SLasso</td>
<td>0.1</td>
<td>15(0.0)</td>
<td>86.6(35.94)</td>
<td>0.982(0.009)</td>
<td>0.807(0.055)</td>
</tr>
<tr>
<td>MW-R2</td>
<td>0.1</td>
<td>15(0.0)</td>
<td>15.25(0.44)</td>
<td>1.000(1e-4)</td>
<td>0.016(0.028)</td>
</tr>
<tr>
<td>LASSO</td>
<td>0.5</td>
<td>15(0.0)</td>
<td>113.3(26.96)</td>
<td>0.975(0.007)</td>
<td>0.860(0.036)</td>
</tr>
<tr>
<td>SLasso</td>
<td>0.5</td>
<td>15(0.0)</td>
<td>168.0(45.59)</td>
<td>0.962(0.011)</td>
<td>0.896(0.059)</td>
</tr>
<tr>
<td>MW-R2</td>
<td>0.5</td>
<td>14.95(0.22)</td>
<td>15.7(1.45)</td>
<td>1.000(3e-4)</td>
<td>0.042(0.073)</td>
</tr>
<tr>
<td>LASSO</td>
<td>0.9</td>
<td>15(0.0)</td>
<td>112.95(31.76)</td>
<td>0.975(0.008)</td>
<td>0.857(0.040)</td>
</tr>
<tr>
<td>SLasso</td>
<td>0.9</td>
<td>14.45(0.89)</td>
<td>62.15(11.40)</td>
<td>0.988(0.003)</td>
<td>0.761(0.038)</td>
</tr>
<tr>
<td>MW-R2</td>
<td>0.9</td>
<td>15(0.0)</td>
<td>21.5(0.51)</td>
<td>0.998(1e-4)</td>
<td>0.302(0.017)</td>
</tr>
</tbody>
</table>
Table 4.6. Values of $\alpha$’s for SLasso and MW-R2 in mean(standard deviation). 40% of the regression coefficients are not matched.

<table>
<thead>
<tr>
<th></th>
<th>p=1000</th>
<th></th>
<th></th>
<th>p=4000</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\rho = 0.1$</td>
<td>$\rho = 0.5$</td>
<td>$\rho = 0.9$</td>
<td>$\rho = 0.1$</td>
<td>$\rho = 0.5$</td>
<td>$\rho = 0.9$</td>
</tr>
<tr>
<td>SLasso</td>
<td>0.84(0.163)</td>
<td>0.886(0.053)</td>
<td>0.9(0.0)</td>
<td>0.9(0.0)</td>
<td>0.895(0.022)</td>
<td>0.9(0.0)</td>
</tr>
<tr>
<td>MW-R2</td>
<td>0.3(0.181)</td>
<td>0.712(0.151)</td>
<td>0.152(0.091)</td>
<td>0.28(0.188)</td>
<td>0.715(0.190)</td>
<td>0.17(0.122)</td>
</tr>
</tbody>
</table>

unrelated noises and these two methods have high FDR. MW-R2 is able to significantly reduce the noises. When there are slight or moderate correlations, MW-R2 is able to remove almost all unrelated noises and find out the exact set of non-zero coefficients. When there exist high correlations, the FDR of MW-R2 is 30%, which is still an acceptable value.

The $\alpha$ values are displayed in Table 4.6. MW-R2 always results in a lower value of $\alpha$ which indicates a stronger effect of the smoothing penalty.
CHAPTER 5
CONCLUSION AND FUTURE WORK

Penalized regression is a modern approach to handling the variable selection problem when \( p \gg n \). This thesis focuses on penalized regression with a moving-window smoothing penalty.

In Chapter 2, we have proposed a penalized moving-window regression method that incorporates adjacent LD information in genome-wide association studies. This method is an extension to the SMCP method in that the smoothing penalty considers more than 2 SNPs. By including more SNPs in a smoothing window, it is expected that valuable LD information among neighboring SNPs can be better utilized. For dense SNPs typically seen in nowadays association studies, LD information captures by 2 SNPs may be rather limited. Indeed, our simulation has demonstrated that including more than 2 SNPs in a moving-window does improve the PPV and TPR of association studies. The proposed moving-window regression also has a clustering effect in which SNPs in LD tend to be selected together. The simulation study also confirms the intuition that including too many SNPs has a negative effect on the performance of the proposed method as true signals tend to be smoothed out while false signals tend to be picked up.

We also described two coordinate descent algorithms for the proposed method. One for quadratic loss and the other for logistic loss. To enhance the computation speed, explicit expressions for updating parameter estimates are given for each step of the algorithm.
We have used a constant window size $d$ across the genome in order to achieve computation efficiency. In theory, it is possible to make $d$ adaptive to local features of the genetic structure such as the density of SNPs and the strength of LD at the cost of extra computation time.

We note that $\sum_{k,j \in W_s, k < j} (|\beta_k| - |\beta_j|)^2$ is proportional to the sample variance of the $|\beta|$s that are in window $W_s$. So the smoothing penalty $S(S_s; \eta) \propto \sum_{k,j \in W_s, k < j} \zeta_{k,j} (|\beta_k| - |\beta_j|)^2$ can be regarded as a measure of variation in $|\beta|$s that are in $W_s$ but with pair-wise weights $\{\zeta_{k,j}\}$.

In Chapter 3, we proposed two methods for generalized linear regressions: SLasso and MW-Ridge. In Chapter 2, there proposed a feature selection approach by using the marginal loss function under the context of the genome-wide association study. Here we applied the joint loss as opposed to the marginal loss, so the model can serve for both variable selection and prediction. Such approach has been introduced (20) before, but the predictive performances have never been studied. In this chapter, we studied the predictive performances and found that compared with LASSO, SLasso tends to select more true positives with a price as including more random noises. Two coordinate descent algorithms are developed to implement the SLasso method. One for quadratic loss and the other for logistic loss.

In order to eliminate the unrelated noises, we proposed the MW-Ridge approach. As a two-stage method, in the first stage, SLasso is applied in order to find the candidate set of the non-zero coefficients. In the second stage, ridge regression is applied for coefficient estimates. MW-Ridge is able to significantly reduce the ran-
dom noises and at the same time, maintain the true positives in the model. In the circumstance of highly correlated data, MW-Ridge is able to over perform relaxed LASSO by achieving higher TPR and lower FDR.

In the simulation, with different window size as two and four, MW-Ridge results in very similar performances. Consider the computational expense, it is reasonable to set two as the window size.

Using MCP instead of the $L_1$ penalty, SMCP is able to relax the biases from LASSO. However, SMCP doesn’t relax the smoothing penalty and the coefficients are shrunk towards zero (22). Compared with SMCP, the main advantage of MW-Ridge is the relaxation of both the $L_1$ penalty and the moving-window penalty. As shown in Chapter 2, the moving-window penalty is able to improve the selection accuracy, while at the same time, shrink the coefficient estimates towards zero. The relaxation of the moving-window penalty can release such biases and result in good performances in both selection and prediction.

In Chapter 4, the SLasso and MW-Ridge approaches are implemented on multivariate response data. The response variables are correlated through residuals. By using some algebra, the multivariate response data can be transformed to the univariate response data. Then SLasso and MW-Ridge are applied on the univariate response problem. In the simulation, we found classical LASSO can detect true causative factors but including many random noises and result in high FDR. In the circumstance of low or moderate correlations, MW-Ridge is able to detect the exact set of true positives and remove almost all random noises. In the circumstance of
high correlations, the FDR of MW-Ridge is around 30%, which is still an acceptable value. MW-Ridge performs well no matter the non-zero regression coefficients are all matched or 40% are unmatched. Compared with SLasso, the moving-window penalty is encouraged to be stronger in MW-Ridge.

The moving-window penalty encourages the smoothing effects between consecutive predictors. In some circumstance, the feature matrix can be transformed into a graph. Each feature can be considered as a vertex and if two features are highly correlated, there is an edge connecting the two corresponding vertices. Regularized regression can be applied to such graph-structured problem. If two vertices are connected by an edge, then the moving-window penalty can be proposed to smooth the corresponding regression coefficients.
REFERENCES


