Abstract

ZHAO, YAQING. Solution Paths for the Generalized Lasso with Applications to Spatially Varying Coefficients Regression. (Under the direction of Howard Bondell.)

Penalized regression can improve prediction accuracy and reduce dimension. The generalized lasso problem is used in many applications in various fields. The generalized lasso penalizes a linear transformation of the coefficients rather than the coefficients themselves. Here we propose an algorithm to solve the generalized lasso problem and provide the full solution path. A confidence set can then be constructed on the generalized lasso parameters based on the modified residual bootstrap lasso. We demonstrate our approach using spatially varying coefficients regression. We show that our algorithm is both accurate and efficient compared to previous work.

© Copyright 2018 by Yaqing Zhao

All Rights Reserved

Solution Paths for the Generalized Lasso with Applications to Spatially Varying
Coefficients Regression

by
Yaqing Zhao

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Statistics

Raleigh, North Carolina

2018

APPROVED BY:

Wenbin Lu

Brian Reich

Daowen Zhang

Howard Bondell
Chair of Advisory Committee

Dedication

This work is dedicated to my parents Qingjin Zhao and Xingmei Wang, and my husband Tao Huang. Thank you for your love, support and encouragement.

Biography

Yaqing Zhao was born in Xuzhou, Jiangsu province, China. She graduated from Xuzhou No.1 Middle school with honor in July 2007 and attended Wuhan University to pursue her undergraduate degree in chemistry. In 2011, she got the Bachelors' degree in science and then was admitted to Department of Chemistry in North Carolina State University for the graduate study in analytical chemistry. In Dr Gufeng Wang's lab, she developed 3D super localization imaging technique via pattern recognition analysis. She got the Master of Science degree in December 2013 and inspired by the data analysis in imaging, she pursued her graduate study in Statistics since January, 2014. Since the spring of 2015, she joined Dr Howard Bondell's research group and started to work on the algorithm to solve the generalized lasso problem and its applications in spatially varying coefficients regression. Meanwhile, she also worked as a teaching assistant in the Department of Statistics in the spring and summer of 2015, and then worked as a graduate industrial trainee at GlaxoSmithKline since August 2015.

Acknowledgements

I would like to gratefully and sincerely give my deepest thank to my advisor, Dr. Howard Bondell, for his guidance during my graduate study in the Department of Statistics. He has inspired me to be an independent researcher and set an excellent example as a brilliant statistician. I would like to express the same gratitude to my committee members, Dr. Wenbin Lu, Dr. Brian Reich and Dr. Daowen Zhang for their kind support and critical comments. In particular, I must give my special thank to Dr. Brain Reich for the data that is used in this thesis.

I'm truly grateful to be a member of the department of statistics. It has been an amazing experience to study and research here. I want to thank all professors that have taught me during the many years for their excellent teaching and the precious knowledge. I also would like to express my appreciation to the department for the opportunity to work as a Graduate Industrial Trainee, which inspires me to solve complex real world problems using my statistical knowledge. I must thank my manager Katja Remlinger at GlaxoSmithKline for her mentoring to improve my skills and confidence as a statistician.

Last, but not least, I must acknowledge my beloved family. To my parents, thank you for always being so supportive. Special thank to my best friend, soul-mate and husband, Tao, for his unwavering love, continuing support and encouragement through this entire process. There are no words that can express my appreciation for your company and faith in me.

Table of Contents

List of Tables	vii
List of Figures	viii
Chapter 1 Introduction	1
1.1 Lasso and Ridge Regression	1
1.2 Generalized lasso and Generalized Ridge	2
Chapter 2 Applications of Generalized Lasso	5
2.1 Fused Lasso	5
2.1.1 1D Fused Lasso	6
2.1.2 Multi-dimensional Fused Lasso	7
2.2 Trend Filtering	8
2.3 Other Applications	9
Chapter 3 The Generalized Lasso and Related Work	11
3.1 Asymptotic properties	11
3.2 Existing Algorithms	13
3.2.1 Generalized Lasso To Lasso	14
3.2.2 The Lagrange Dual Problem	16
3.2.3 ADMM and Augmented ADMM	19
3.2.4 Connection To Constrained Lasso	22
3.3 Implementation and Computation Performance	24
3.4 Lasso Inference	26
3.4.1 The Conditional Sampling Distribution Approach	27
3.4.2 Resampling-Based Approach	31
3.4.3 Bias Correction Approach	33
3.4.4 Stochastic Variational Inequality	34
3.4.5 Bayesian Lasso	37
3.4.6 Lasso inference to generalized lasso inference	38
Chapter 4 Proposed Algorithm And Confidence Intervals For The Generalized Lasso	40
4.1 Proposed Algorithm	40
4.1.1 D is full rank	40
4.1.2 D has full column rank	41
4.1.3 A general matrix D	43
4.2 Generalized Lasso Inference	44
4.2.1 Modified Residual Bootstrap Lasso	45

4.2.2	Confidence Intervals For The Original Parameters	48
Chapter 5	Simulation and Real Data Analysis	50
5.1	Simulation Study	50
5.2	Comparison to the genlasso package	55
5.3	Real Data Analysis	57
Chapter 6	Discussion and Conclusion	67
References		70

List of Tables

Table 3.1	Computational complexity based on the QR decomposition.	25
Table 3.2	Computational complexity of ADMM and augADMM.	25
Table 5.1	The table shows median MSPE and standard error of four different approaches for different structures for 100 simulated datasets. . . .	54
Table 5.2	Comparison of mean MSPE and standard error for four different models.	60
Table 5.3	Comparison of mean MSPE between four different methods with or without spatially varying the slope.	62

List of Figures

Figure 5.1	Runtime of both methods in seconds with increasing number of locations on the 2D grid.	55
Figure 5.2	MSE of both methods with increasing steps in the solution path. The dashed blue line indicate the optimal step chosen by cross validation.	56
Figure 5.3	Distribution of richness of fungi across the US.	58
Figure 5.4	Using voronoi tessellation, we determine the neighborhood structure of the pre-clustered groups.	59
Figure 5.5	Plot of the intercept when both the intercept and slope are spatially varying.	61
Figure 5.6	Plot of the coefficient of number of bedrooms when both the intercept and slope are spatially varying.	61
Figure 5.7	Intercept for different locations when the slope is fixed.	63
Figure 5.8	The overall OLS estimator, generalized lasso estimators and confidence intervals for the intercept.	64
Figure 5.9	Intercept estimated by the generalized lasso and whether the overall OLS is covered by the confidence interval at each location.	64
Figure 5.10	Clustering of locations by the generalized lasso.	66
Figure 5.11	Clustering of locations by the confidence intervals of the generalized lasso.	66

Chapter 1

Introduction

1.1 Lasso and Ridge Regression

Regularization methods have attracted much attention and a great deal of progress has been made in these approaches. Linear regression including a penalty of the l_2 norm or l_1 norm of the coefficients are well-known as ridge regression (Hoerl and Kennard, 1970) and lasso (least absolute shrinkage and selection operator) regression (Tibshirani, 1996, Chen et al., 2001). Let $\mathbf{y} \in \mathbb{R}^n$ be a vector of responses, $X \in \mathbb{R}^{n \times p}$ be a matrix of predictors and $\boldsymbol{\beta} \in \mathbb{R}^p$ be a vector of coefficients. Then ridge and lasso regression can be expressed as (1.1) and (3.39), respectively:

$$\hat{\boldsymbol{\beta}}^{Ridge} = \arg \min_{\boldsymbol{\beta}} \frac{1}{2} \|\mathbf{y} - X\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_2^2, \quad (1.1)$$

$$\hat{\boldsymbol{\beta}}^{Lasso} = \arg \min_{\boldsymbol{\beta}} \frac{1}{2} \|\mathbf{y} - X\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1, \quad (1.2)$$

where $\lambda \in [0, \infty)$ is a tuning parameter specifying how much the coefficients are penalized. Ridge regression achieves an appropriate trade-off between model bias and variance in

linear regression, which improves prediction accuracy, especially when p is large. It shrinks the estimated coefficient vector to be smaller and has a closed form solution:

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

Along with achieving a variance-bias balance, lasso also produces solutions of sparse structures by shrinking some parameters to exactly zero, which provides better interpretable models in many cases. In data analysis, lasso achieves variable selection by selecting the important predictors and discarding the unimportant ones. There exists a number of lasso solvers, for example, the most widely used R packages *lars* (Hastie and Efron, 2013) and *glmnet* (Friedman et al., 2010) are based on LARS (Least Angle Regression) algorithm (Efron et al., 2004) and pathwise coordinate descent (Friedman et al., 2010), respectively. Among these algorithms, the *lars* algorithm provides the complete solution path for the lasso problem by taking advantage of the fact that solutions of lasso are piecewise linear with respect to λ , and builds a connection between lasso and the forward stepwise selection. Additionally, homotopy (Osborne et al., 2000, Malioutov et al., 2005) is also a path tracking algorithm which can be considered as an extension of *lars*.

1.2 Generalized lasso and Generalized Ridge

There are a variety of lasso-type extensions, such as the fused lasso (Tibshirani et al., 2005) and adaptive lasso (Zou, 2006) among others. Tibshirani and Taylor (2011) discussed a generalized lasso problem, which is defined as:

$$\min_{\boldsymbol{\beta}} \frac{1}{2} \|\mathbf{y} - X\boldsymbol{\beta}\|_2^2 + \lambda \|D\boldsymbol{\beta}\|_1, \tag{1.3}$$

where $D \in \mathbb{R}^{m \times p}$ is a matrix that allows us to specify the desired structural/geometric property of a problem. The choice of D leads to different forms of lasso type problems and have broad applications, for example, fused lasso (Tibshirani et al., 2005), trend filtering (Kim et al., 2009), network lasso (Hallac et al., 2015), and others.

Similarly, the generalized ridge problem can be defined as:

$$\min_{\beta} \frac{1}{2} \|y - X\beta\|_2^2 + \lambda \|D\beta\|_2^2. \quad (1.4)$$

Unlike lasso, the generalized ridge problem does not encourage sparse structures in the solution. However, it has a nice closed form solution: $\hat{\beta} = (X^T X + \lambda D^T D)^{-1} X^T y$, and the degrees of freedom at each λ value can be computed as: $df(\lambda) = \text{tr}(X(X^T X + \lambda D^T D)^{-1} X^T)$. In this paper, we mainly focus on the generalized lasso problem, although we will discuss the generalized ridge as well. We will focus on the generalized lasso problem in the case where $m > p$ for the penalty matrix D , such as the 2D fused lasso. We propose a new algorithm to solve these types of generalized lasso problems in such a way that we allow for the use of existing lasso solvers, like lars, to get the full solution path. We also construct confidence intervals for the generalized lasso based on the modified bootstrap lasso approach (Chatterjee and Lahiri, 2011).

The thesis is organized as follows: in Chapter 2, we introduce some applications of the generalized lasso. It is shown that by using different penalty matrix D , the generalized lasso can take the form of many problems that are of interest. In Chapter 3, we discuss the asymptotic properties of the generalized lasso. We then introduce and compare existing algorithms to solve the problem. Since we build confidence intervals of the generalized lasso by reducing it to a regular lasso, we also discuss current methods in lasso inference in section (3.4). In Chapter 4, we propose our new algorithm to build solution path

for the generalized lasso when the penalty matrix $D_{m \times p}, m \geq p$ takes any form. And in section (4.2), we construct confidence intervals for the generalized lasso estimators based on the modified bootstrap approach. Section (5.1) shows the simulation study that demonstrates our proposed algorithm via spatially varying coefficients regression. In the simulation, we compare the generalized lasso, generalized ridge with two other models under different coefficient structures. We also compare our proposed algorithm to the existing generalized lasso path solution, and shows that our method is both efficient and accurate. Finally, in section (5.3) we show the use of our algorithm and confidence intervals in a real data example.

Chapter 2

Applications of Generalized Lasso

One important motivation to study the generalized lasso problem is because of its wide applications. Different structures of D result in different types of problems and have a variety of applications. In this section, we will take a look at the broad applications of the generalized lasso. We will focus on some popular problems in this family, however, its wide applications are not limited by what we list here.

2.1 Fused Lasso

The fused lasso problem was first proposed by Tibshirani et al. (2005). It is defined as:

$$\min \sum_i (y_i - \sum_j x_{ij} \beta_j)^2 \text{ subject to } \sum_{i=j} |\beta_j| \leq s_1 \text{ and } \sum_{j=2} |\beta_j - \beta_{j-1}| \leq s_2. \quad (2.1)$$

In most literatures, (2.1) is usually referred as the "sparse fused lasso" problem. In a sparse fused lasso problem, the second penalty shrinks the coefficients in the neighborhood towards each other. And a "fused lasso" problem only penalizes on the differences between neighboring parameters. Just like lasso encourages sparsity in β , the sparse

fused lasso and fused lasso problem also encourage adjacent coefficients to be exactly the same.

2.1.1 1D Fused Lasso

When the problem is of a one-dimensional structure and the coefficients are penalized between its neighbors in one-dimension, for example, 1d signals, we consider the fused lasso as a 1d fused lasso problem. Considering the 1d fused lasso as a special case of the generalized lasso, the penalty matrix D can be written as:

$$D \in \mathbb{R}^{(p-1) \times p} = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -1 & \cdots & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 0 & 1 & -1 \end{bmatrix} \quad (2.2)$$

where each row of D represents difference of a pair of adjacent parameters. When $X = I$, this type of problems are called as fused-lasso signal approximator (FLSA). Comparative genomic hybridization (CGH) analysis is one of the many popular applications of fused lasso (Tibshirani and Wang, 2008, Nowak et al., 2011). CGH can be used to measure the DNA copy number variations (CNVs) which records events of loss or gain of genes on the genome. It has been studied and shown that unusual amplification or deletion of genes from the chromosome is related to cancer or other diseases (Feuk et al., 2006). The response $\mathbf{y}$ in CGH is a log ratio of DNA numbers of a gene in the tumor cells to that in the control cells. It is believed that adjacent sample locations are more likely to have similar CNVs, and CGH returns signals that are sparse in certain areas. As a result, fused lasso is a good way to smooth the CGH signal. Actually, the fused lasso

problem can be applied to many gene-related or spectroscopy-related problems. Tibshirani and Taylor (2011) showed the applications of fused lasso on mass spectroscopy and micro-array data, which are related to prostate cancer and leukaemia, respectively. Yang et al. (2016) discussed the applications of fused lasso on genome-wide associated problems. For example, when analyzing the relationship between genotype and phenotype using the Genome-Wide Association (GWA) study, it benefits from incorporating linkage disequilibrium (LD) (Reich et al., 2001) information between adjacent single-nucleotide polymorphisms (SNP) (Zhu et al., 2013, Liu et al., 2011).

2.1.2 Multi-dimensional Fused Lasso

In many cases, the problem is beyond one-dimensional structure, for example, on a surface, and the fused lasso problem penalizes on adjacent coefficients in higher dimensions. Consider a graph $\mathcal{G}$ with nodes $\mathcal{V}$ and edges $\mathcal{E}$, we can use fused lasso to penalize differences between nodes that are joined by edges. We can write the problem as:

$$\min_{\boldsymbol{\beta}} \|\mathbf{y} - X\boldsymbol{\beta}\|_2^2 + \lambda \sum_{j,k \in \mathcal{E}} |\beta_j - \beta_k| \quad (2.3)$$

Let D_i denote the i^{th} row of the matrix and it can be represented as:

$$D_i = (0, \dots, 1, \dots, -1, \dots, 0)$$

where 1 and -1 are the j^{th} and k^{th} element of this row, β_j and β_k are connected by edges. An example using two-dimensional fused lasso is to do image noise reduction by penalizing differences of signals between two vertically and horizontally adjacent pixels.

Another application of 2d fused lasso is to solve the spatially varying coefficients

regression. In spatially varying coefficients problems, the interest is to build a regression over a certain area while the coefficients vary at different locations. Hallac et al. (2015) showed the application of network lasso on a house price problem, which can be regarded as an adaptive fused lasso problem (the problem and algorithm are discussed in detail in section 3.2.3). The real data contains estate transactions for a short period of time in the Greater Sacramento area. The house locations are provided using latitudes and longitudes. Predictors are number of bedrooms, number of bathrooms, size of the house, and the response is house price. It is reasonable to assume that the house size and number of rooms affect the sales price differently at different locations. To solve such a spatially varying coefficients problem, Hallac et al. (2015) computed the nearest 5 houses for each location and thus built edges in the problem and used the adaptive fused lasso to achieve clustering and estimation.

The fused lasso problem was also used in human brain image applications using MRI. The MRI brain image can be considered as a 3D grid. The image is expected to be sparse with signals clustering at certain regions indicating early damage. Thus one can apply fused lasso to analyze such a problem. For example, Xin et al. (2014) used fused lasso in the diagnosis of Alzheimers disease (AD).

2.2 Trend Filtering

An l_1 linear trend filtering problem (Kim et al., 2009) is:

$$\min \|\mathbf{y} - \boldsymbol{\beta}\|_2^2 + \lambda \sum_{i=2} |\beta_{i-1} - 2\beta_i + \beta_{i+1}|. \quad (2.4)$$

When it is written as a generalized lasso problem, we have $X = I$ and the penalty matrix D is:

$$D_T \in \mathbb{R}^{(n-2) \times n} = \begin{bmatrix} 1 & -2 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -2 & \cdots & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 1 & -2 & 1 \end{bmatrix}$$

This so-called l_1 trend filtering has many applications in broad areas like finance (Tsay, 2005), biology (Greenland and Longnecker, 1992), social science (Levitt, 2004), etc. There are many approaches (Osborn, 1995, Leser, 1961) to solve the l_1 trend filtering problem and the generalized lasso is one of them. For the k^{th} order polynomial trend filtering, D_k can be directly defined in terms of a composition of individual filters (Tibshirani and Taylor, 2011):

$$D_k = DD_{k-1}$$

where D has the form in (2.2). Trend filtering on graphs (two-dimensional) has also been discussed and applied (Wang et al., 2015).

2.3 Other Applications

In addition to the fused lasso and trend filtering, the generalized lasso can be applied to many other problems. Tibshirani and Taylor (2011) discussed how generalized lasso can be applied to a most common wavelet smoothing problem, which solves a lasso problem:

$$\min_{\boldsymbol{\theta}} \frac{1}{2} \|\mathbf{y} - W\boldsymbol{\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}\|. \quad (2.5)$$

where $W \in \mathbb{R}^{n \times n}$ has orthogonal basis in columns. This can be transformed to a generalized lasso problem by changing variable $\beta = W\theta$, thus $D = W^T$ and $X = I$. Besides, an outlier detection approach was also discussed and can be transformed to a generalized lasso problem (Tibshirani and Taylor, 2011).

Chapter 3

The Generalized Lasso and Related Work

3.1 Asymptotic properties

Tibshirani et al. (2005) discussed the asymptotic properties of fused lasso. The fused lasso problem in (2.1) is written as:

$$\min \sum_i (y_i - \sum_j x_{ij} \beta_j)^2 + \lambda_N^{(1)} \sum_{j=1} |\beta_j| + \lambda_N^{(2)} \sum_{j=2} |\beta_j - \beta_{j-1}|. \quad (3.1)$$

where N represents sample size, and $\lambda_N^{(1)}$ and $\lambda_N^{(2)}$ are functions of N . A classical setting $\mathcal{C}$ is first defined: assuming p is fixed with $N \rightarrow \infty$, $\Sigma \triangleq X^T X/n \rightarrow C$ is positive definite with probability 1. Tibshirani and Taylor (2011) showed that if $\lambda_N^{(l)}/\sqrt{N} \rightarrow \lambda_0^{(l)} \geq 0$ ($l = 1, 2$) and $\mathcal{C}$ is satisfied, then

$$\sqrt{N}(\hat{\beta}_N - \beta) \xrightarrow{d} \arg \min(V) \quad (3.2)$$

where

$$\begin{aligned}
V(\mathbf{u}) = & -2\mathbf{u}^T \mathbf{W} + \mathbf{u}^T C \mathbf{u} + \lambda_0^{(1)} \sum_{j=1}^p u_j \text{sgn}(\beta_j) I(\beta_j \neq 0) + |u_j| I(\beta_j \neq 0) \\
& + \lambda_0^{(l)} \sum_{j=1}^p (u_j - u_{j-1}) \text{sgn}(\beta_j - \beta_{j-1}) I(\beta_j \neq \beta_{j-1}) + |u_j - u_{j-1}| I(\beta_j \neq \beta_{j-1})
\end{aligned}$$

where $\mathbf{W}$ follows a normal distribution with mean $\mathbf{0}$ and variance $\sigma^2 C$.

She (2008) provided some consistency results for the generalized lasso. They proved by KKT condition that $\hat{\beta}$ is optimal if and only if when:

$$X^T(X\beta - \mathbf{y}) + \lambda D^T \text{sgn}(D\beta) = \mathbf{0}.$$

for some $\text{sgn}(D\beta)$. They showed that under condition $\mathcal{C}$, if $\lambda = o(n)$, then $\hat{\beta} \xrightarrow{p} \beta$, and thus $D\hat{\beta} \xrightarrow{p} D\beta$. To achieve consistency, there is no more restriction on D and Σ , one just needs to properly choose λ .

Aside from this, they also study two other consistencies: sign consistency and zero s-consistency. $\hat{\beta}$ is defined to be sign consistent if $P(\text{sgn}(\hat{\beta}) = \text{sgn}(\beta)) \rightarrow 1$. If β satisfies $A\beta = \mathbf{0}$ for some A , $\hat{\beta}$ is defined to be zero s-consistent with respect to A if $P(A\hat{\beta} = \mathbf{0}) \rightarrow 1$. Zero sign consistency is important in terms of variable selection. It is worth noting that consistency implies nonzero sign consistency. And zero s-consistency can be used to characterize sparsity and implies clustering the true groups asymptotically. However, if $\lambda = O(n)$, then $\hat{\beta}$ is not zero s-consistent with respect to D . Under condition $\mathcal{C}$, let $D = \begin{bmatrix} D_1 \\ D_2 \end{bmatrix}$, where $D_1\beta = \mathbf{0}$, $\frac{\lambda}{n} \rightarrow 0$, $\frac{\lambda}{\sqrt{n}} \rightarrow \infty$. Let A^+ denote the Moore-Penrose inverse of A . Then a necessary condition for $\hat{\beta}$ to be zero s-consistent with respect to D_1

is:

$$\begin{aligned} \exists \quad \tilde{\text{sgn}}(D_2\boldsymbol{\beta}) \quad s.t. \quad & \| (D_1C^{-1}D_1^T)^+ (D_1C^{-1}D_2^T) \cdot \tilde{\text{sgn}}(D_2\boldsymbol{\beta}) \|_\infty \\ & \leq \| (D_1C^{-1}D_1^T)^+ (D_1C^{-1}D_2^T) \|_\infty. \end{aligned}$$

And a sufficient condition is that:

$$\| (D_1C^{-1}D_1^T)^+ (D_1C^{-1}D_2^T) \cdot \tilde{\text{sgn}}(D_2\boldsymbol{\beta}) \|_\infty < 1 \quad \forall \tilde{\text{sgn}}(D_2\boldsymbol{\beta}).$$

That is to say, unlike consistency where one just needs to properly choose $\lambda's$, to discover the true sparsity, there are more constraints requirements and we also need to properly control $(D_1C^{-1}D_1^T)^+ (D_1C^{-1}D_2^T)$, which is hard.

3.2 Existing Algorithms

There are a number of authors that have discussed algorithms to solve the lasso-type problems. Tibshirani et al. (2005) first proposed the fused lasso problem. Regarding this as a quadratic optimization problem, they used the two-phase active set algorithm in package SQOPT (Gill et al., 1997) to solve it for each pair of the tuning parameters. Friedman et al. (2007) proposed a pathwise coordinate descent algorithm to solve lasso, elastic net and the fused-lasso signal approximation (FLSA) where $X = I$. Their algorithm also works for two-dimensional fused lasso, however, it does not guarantee the exact solution for a general design matrix X . In this section, we will discuss several algorithms that consider the lasso type problems in its general form (1.3).

3.2.1 Generalized Lasso To Lasso

A natural approach is if the generalized lasso problem can be reduced to a regular lasso, then standard routine can be applied and various lasso solvers are available according to different needs. It is simple and intuitive to do the transformation when D is a $p \times p$ full rank matrix. Let X be a general design matrix, then problem (1.3) can be reparameterized as

$$\min_{\boldsymbol{\theta}} \frac{1}{2} \|\mathbf{y} - XD^{-1}\boldsymbol{\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_1 \quad (3.3)$$

where $\boldsymbol{\theta} = D\boldsymbol{\beta}$ and estimation for $\boldsymbol{\beta}$ can be easily achieved by $\hat{\boldsymbol{\beta}} = D^{-1}\hat{\boldsymbol{\theta}}$. Tibshirani and Taylor (2011) showed under the condition that D is a $m \times p$ matrix and $\text{rank}(D) = m$, the generalized lasso can also be transformed to the lasso problem. They first found a new matrix A whose rows are orthogonal to those of D . Matrix $\tilde{D} \in \mathbb{R}^{p \times p} = \begin{bmatrix} D \\ A \end{bmatrix}$ is then formed and is an invertible full rank matrix. Let $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)^T = \tilde{D}\boldsymbol{\beta}$, the generalized lasso problem can be rewritten as:

$$\min_{\boldsymbol{\theta}} \frac{1}{2} \|\mathbf{y} - X\tilde{D}^{-1}\boldsymbol{\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}_1\|_1. \quad (3.4)$$

By defining $X\tilde{D}^{-1}\boldsymbol{\theta} = X_1\boldsymbol{\theta}_1 + X_2\boldsymbol{\theta}_2$, they further reduced (4.1) to:

$$\min_{\boldsymbol{\theta}_1} \frac{1}{2} \|(I - P)\mathbf{y} - (I - P)X_1\boldsymbol{\theta}_1\|_2^2 + \lambda \|\boldsymbol{\theta}_1\|_1 \quad (3.5)$$

where $P = X_2^T(X_2^T X_2)^{-1}X_2^T$. This is already a regular lasso problem and standard lasso algorithms can be directly applied to it to get $\hat{\boldsymbol{\theta}}_1$. $\boldsymbol{\theta}_2$ can also be easily estimated by $\hat{\boldsymbol{\theta}}_2 = (X_2^T X_2)^{-1}X_2^T(\mathbf{y} - X_1\hat{\boldsymbol{\theta}}_1)$, thus $\boldsymbol{\beta}$ coefficients can be estimated by doing back transformation $\hat{\boldsymbol{\beta}} = \tilde{D}^{-1}(\hat{\boldsymbol{\theta}}_1, \hat{\boldsymbol{\theta}}_2)^T$.

Duan et al. (2016) further discussed the connection between generalized lasso and regular lasso recently. By assuming a full row rank matrix D ($\text{rank}(D) = m, m \leq p$), they also showed that the generalized lasso can be transformed to a basic lasso. They claimed that given such a D and a full column rank matrix $\Phi = \begin{bmatrix} A \\ D \end{bmatrix}$, the generalized lasso problem can be transformed as:

$$\min_{\boldsymbol{\theta}} \frac{1}{2} \|\mathbf{z} - H\boldsymbol{\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_1 \quad (3.6)$$

where $H = Q_1 Q_2^\dagger$, $\mathbf{z} = (I - Q_1(I - Q_2^\dagger Q_2)Q_1^T)y$, $Q_2^\dagger = Q_2^T(Q_2 Q_2^T)^{-1}$. Q_1 and Q_2 come from QR decomposition: $\Phi = QR = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} R$. Duan et al. (2016) compared their method to the previous transformation and got the same results. However, they mentioned that improper choice of A in Tibshirani and Taylor (2011)'s transformation may result in numerical errors.

By reducing the generalized lasso to regular lasso problems, we can achieve the complete solution path and degree of freedom at each step easily by applying algorithms like LARS.

Both Tibshirani and Taylor (2011) and Duan et al. (2016)'s reduction approaches work only when D is a full row rank matrix. In the 1d fused lasso problem, the penalty matrix D which is in the form of 2.2 has full row rank. However, in many real applications, D does not satisfy the condition. One example is the 2d fused lasso problem described in section 2.1.2. The penalty matrix describes difference between neighbors in 2 dimensions, thus $m > p$. In this case, the transformations do not apply.

3.2.2 The Lagrange Dual Problem

Inspired by LARS, Tibshirani and Taylor (2011) proposed an algorithm to solve the generalized lasso problem by solving its dual. When $X = I$, they reformulated 1.3 as:

$$\min_{\beta, z} \frac{1}{2} \|\mathbf{y} - \beta\|_2^2 + \lambda \|\mathbf{z}\|_1 \quad \text{subject to } \mathbf{z} = D\beta \quad (3.7)$$

and then derived the dual problem of (1.3) given any fixed λ :

$$\min_{\mathbf{u} \in \mathbb{R}^m} \frac{1}{2} \|\mathbf{y} - D^T \mathbf{u}\|_2^2 \quad \text{subject to } \|\mathbf{u}\|_\infty \leq \lambda \quad (3.8)$$

Similarly, when X is a general matrix, the dual problem is:

$$\min_{\mathbf{u} \in \mathbb{R}^m} \frac{1}{2} \|XX^+ \mathbf{y} - (X^+)^T D^T \mathbf{u}\|_2^2 \quad \text{subject to } \|\mathbf{u}\|_\infty \leq \lambda, D^T \mathbf{u} \in \text{row}(X) \quad (3.9)$$

where X^+ is the Moore-Penrose inverse of X . The primal solution is $\hat{\beta} = \mathbf{y} - D^T \hat{\mathbf{u}}$ for $X = I$ and $X\hat{\beta} = XX^+ \mathbf{y} - (X^+)^T D^T \hat{\mathbf{u}}$ for any general X . Not only giving the dual solution and thus getting the primal solution $\hat{\beta}(\lambda)$ of the generalized lasso by back transforming at a fixed λ value, they also computed a full solution path for the problem. This algorithm starts a path from $\lambda = \infty$ to $\lambda = 0$. They referred the coordinates as "boundary coordinates" where $\hat{u}(\lambda) = \pm\lambda$, and others as "interior coordinates". Let $\mathcal{B}$ denote the set that contains the boundary coordinates and s denote the set that contains their signs. $D_{\mathcal{B}}$ represents rows of D that are in $\mathcal{B}$, while $D_{-\mathcal{B}}$ are rows in D that are not in $\mathcal{B}$. Considering the case where $X = I$ and D is a general matrix, the dual solution at

k^{th} iteration for the boundary coordinates is:

$$\hat{u}_{\lambda, \mathcal{B}} = \lambda s \quad \text{for all } \lambda \in [0, \lambda_k] \quad (3.10)$$

and for the interior coordinates, the solution is:

$$\hat{u}_{\lambda_k, \mathcal{B}} = (D_{-\mathcal{B}}(D_{-\mathcal{B}})^T)^{-1} D_{-\mathcal{B}}(y - \lambda_k(D_{\mathcal{B}}^T s)). \quad (3.11)$$

When $\lambda \leq \lambda_k$, the solution $\hat{u}_{\lambda, -\mathcal{B}}$ changes linearly with λ : $\hat{u}_{\lambda, -\mathcal{B}} = a - \lambda b$ until the coordinates hits the boundary. The "hitting time" t_i is obtained by solving $a_i - \lambda b_i = \pm \lambda$, where $\hat{a}$ and $\hat{b}$ are least square solutions to:

$$\min_a \|y - D_{-\mathcal{B}}^T a\|_2^2 \quad \text{and} \quad \min_b \|D_{\mathcal{B}}^T s - D_{-\mathcal{B}}^T b\|_2^2 \quad (3.12)$$

$$t_i^{hit} = \frac{\hat{a}_i}{\hat{b}_i \pm 1} = \frac{[(D_{-\mathcal{B}}(D_{-\mathcal{B}})^T)^{-1} D_{-\mathcal{B}} y]_i}{[(D_{-\mathcal{B}}(D_{-\mathcal{B}})^T)^{-1} D_{-\mathcal{B}}(D_{\mathcal{B}})^T s]_i \pm 1} \quad (3.13)$$

the next hitting is:

$$h_{k+1} = \max_i t_i^{hit} \quad (3.14)$$

and the leaving time when the coordinate leaves the boundary is:

$$t_i^{leave} = \begin{cases} c_i/d_i & \text{if } c_i < 0 \text{ and } d_i < 0 \\ 0 & \text{otherwise} \end{cases} \quad (3.15)$$

where

$$\begin{aligned} c_i &= s_i [D_{\mathcal{B}} [I - (D_{-\mathcal{B}})^T (D_{-\mathcal{B}}(D_{-\mathcal{B}})^T)^+ D_{-\mathcal{B}}] y]_i \\ d_i &= s_i [D_{\mathcal{B}} [I - (D_{-\mathcal{B}})^T (D_{-\mathcal{B}}(D_{-\mathcal{B}})^T)^+ D_{-\mathcal{B}}] (D_{\mathcal{B}})^T s]_i \end{aligned} \quad (3.16)$$

thus the next leaving time is:

$$l_{k+1} = \max_i t_i^{leave} \quad (3.17)$$

Building the solution path when $X = I$ can be described in algorithm 1.

Algorithm 1 Solution Path of the Generalized Lasso when $X = I$.

1. Initialize: $k = 0$, $\lambda = \infty$, $\mathcal{B} = \emptyset$, $s = \emptyset$
 2. While $\lambda_k > 0$:
 - a. Compute $\hat{u}_{\lambda_k, -\mathcal{B}}$ according to (3.11).
 - b. Compute $\hat{a}, \hat{b}$ and the hitting time according to (3.13) and (3.14).
 - b. Compute the leaving time according to (3.15), (3.16) and (3.18).
 - d. Let $\lambda_{k+1} = \max(h_{k+1}, l_{k+1})$. If the hitting time is larger, then the hitting coordinate is added to $\mathcal{B}$ and its sign to s ; if on the other hand, leaving time is larger, remove the leaving coordinate from $\mathcal{B}$ and its sign from s .
 - e. The solution at $\lambda \in [\lambda_k, \lambda_{k+1}]$ is $\hat{u} = \hat{a} - \lambda \hat{b}$. Update $k = k + 1$.
-

For a general design matrix X , we can see that although (3.9) looks more complicated than (3.7), they share exactly the same form just with different responses and predictors for the least square problem. Rewrite $\tilde{\mathbf{y}} = XX^+\mathbf{y}$ and $\tilde{D} = DX^+$, algorithm 1 can be applied directly to solve such a problem. Even when X does not have full column rank, the solution path can still be achieved but in a more complex way (Tibshirani and Taylor, 2011).

In this work, they also discussed the connection between their algorithm and LARS. By assuming $D = I$, the generalized lasso takes the same form as the regular lasso. Further assuming that $\text{rank}(X) = p$ and applying algorithm 1 to solve this problem but ignoring step 2(c) (set $l_{k+1} = 0$), the primal path $\hat{\beta}_\lambda = (X^T X)^{-1}(X^T \mathbf{y} - \hat{\mathbf{u}}_\lambda)$ is exactly

the same as the LARS path.

Further relying on the Stein's formula for df (Stein, 1981) and assuming $y \sim N(0, \sigma^2 I)$, the authors derived an unbiased estimator of the degrees of freedom for the generalized lasso problem given by $df(X\hat{\beta}_\lambda) = E[\text{nullity}(D_{-\mathcal{B}(y)})]$. From this, they gave several examples of degrees of freedom in some lasso-types problems. For example, in fused lasso df is the number of fused groups in $\hat{\beta}_\lambda(y)$; for the k^{th} order polynomial trend filtering problem, the degree of freedom is the number of knots $k + 1$. These estimates are intuitive, interpretable and can be easily computed in many cases. Having unbiased estimated df and the complete solution path, one can further perform model selection using criteria like AIC or BIC to choose the value of λ .

3.2.3 ADMM and Augmented ADMM

Alternating direction methods of multipliers (ADMM) algorithm (Boyd et al., 2011) has become popular in convex optimization problems. It was also applied to solve the lasso type problems due to it does not require the loss function to be differentiable. Hallac et al. (2015) considered a problem on the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents nodes and $\mathcal{E}$ are edges on the graph. They defined the network lasso problem as:

$$\sum_{i \in \mathcal{V}} f_i(\beta_i) + \lambda \sum_{j, k \in \mathcal{E}} \omega_{jk} \|\beta_j - \beta_k\|_1 \quad (3.18)$$

The problem penalizes differences of adjacent parameters. When $f(\cdot)$ takes the linear least square form, the network lasso becomes an adaptive fused lasso problem and the authors (Hallac et al., 2015) solved this problem using ADMM approach.

Zhu (2015) described solving the generalized lasso problem by ADMM. Write the

problem as:

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^p} f(\boldsymbol{\beta}) + g(D\boldsymbol{\beta}) \quad (3.19)$$

where $f(\boldsymbol{\beta}) = \|\mathbf{y} - X\boldsymbol{\beta}\|_2^2$ and $g(D\boldsymbol{\beta}) = \lambda\|D\boldsymbol{\beta}\|_1$ as a generalized lasso setting.

The problem is equivalent to:

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^p} f(\boldsymbol{\beta}) + g(\mathbf{z}) \quad \text{subject to} \quad \mathbf{z} = D\boldsymbol{\beta} \quad (3.20)$$

Define ρ as a positive penalty parameter, $\mathbf{u}$ as the dual variable and k as the number of iterations, a standard ADMM approach solves this in the following steps (which is also shown to be equivalent to the *proximal point algorithm* (Eckstein and Bertsekas, 1992)):

$$\boldsymbol{\beta}^{k+1} = \arg \min_{\boldsymbol{\beta}} (f(\boldsymbol{\beta}) + \frac{\rho}{2} \|D\boldsymbol{\beta} - \mathbf{z}^k + \rho^{-1}\mathbf{u}^k\|_2^2) \quad (3.21a)$$

$$\mathbf{z}^{k+1} = \arg \min_{\mathbf{z}} (g(\mathbf{z}) + \frac{\rho}{2} \|D\boldsymbol{\beta}^{k+1} - \mathbf{z} + \rho^{-1}\mathbf{u}^k\|_2^2) \quad (3.21b)$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \rho(D\boldsymbol{\beta}^{k+1} - \mathbf{z}^{k+1}) \quad (3.21c)$$

ADMM breaks large problems down to small pieces thus it is more efficient and can handle large scale problems. Moreover, it is suitably designed for parallel computing. However, it is worth noting that when D is not a diagonal matrix, it is complex and time-consuming to update $\boldsymbol{\beta}$ in (3.21a). As a result, they proposed a modified ADMM algorithm which is called "augmented ADMM" by introducing an augmented variable $(\mathbf{z}, \tilde{\mathbf{z}})$ where $\tilde{\mathbf{z}} = (H - D^T D)^{\frac{1}{2}} \boldsymbol{\beta}$ with $D \succeq D^T D$. The augmented ADMM updates are as follows:

$$\boldsymbol{\beta}^{k+1} = \arg \min_{\boldsymbol{\beta}} (f(\boldsymbol{\beta}) + (2\mathbf{u}^k - \mathbf{u}^{k-1})^T D\boldsymbol{\beta} + \frac{\rho}{2} (\boldsymbol{\beta} - \boldsymbol{\beta}^k)^T H (\boldsymbol{\beta} - \boldsymbol{\beta}^k)) \quad (3.22a)$$

$$\mathbf{z}^{k+1} = \arg \min_{\mathbf{z}} (g(\mathbf{z}) + \frac{\rho}{2} \|D\boldsymbol{\beta}^{k+1} - \mathbf{z} + \rho^{-1}\mathbf{u}^k\|_2^2) \quad (3.22b)$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \rho(D\boldsymbol{\beta}^{k+1} - \mathbf{z}^{k+1}) \quad (3.22c)$$

The augmented ADMM algorithm simplifies $\boldsymbol{\beta}$ updates when structure of D is not simple. This improves computation efficiency compared to the standard ADMM approach.

Hallac et al. (2015) further discussed the regularization path in the network lasso setting. Different from LARS or the dual problem of the generalized lasso (Tibshirani and Taylor, 2011), they started the regularization path from small λ which is close to zero to a stopping large λ value. The λ values increase in a way of $\lambda_{k+1} = \alpha\lambda_k$, where $\alpha > 1$ is a constant. Let $\lambda_{critical}$ denote the stopping value, it is reached when all $\boldsymbol{\beta}'s$ are a constant value and won't change in the next iteration. Initial values of $\boldsymbol{\beta}$, $\mathbf{u}$ and $\mathbf{z}$ at each iteration is the solution of the last step. It only remains to determine the first $\lambda > 0$ which is close to zero. To get this value, they first randomly picked a pair of adjacent $\boldsymbol{\beta}_i, \boldsymbol{\beta}_j$ and estimated their values by $\hat{\boldsymbol{\beta}}_i, \hat{\boldsymbol{\beta}}_j$ when $\lambda = 0$. Next they calculated the gradients of $f_i(\boldsymbol{\beta})$ and $f_j(\boldsymbol{\beta})$ at $\boldsymbol{\beta} = (\hat{\boldsymbol{\beta}}_i + \hat{\boldsymbol{\beta}}_j)/2$. Then the initial value of λ is set to be:

$$\lambda = 0.01 \left(\frac{\|\nabla f_i(\boldsymbol{\beta})\|_2^2 + \|\nabla f_j(\boldsymbol{\beta})\|_2^2}{2\omega_{ij}} \right) \quad (3.23)$$

They repeated this procedure several times and picked a smallest λ and set it to be the initial value in the algorithm. However, they didn't mention how α was selected which would also influence the regularization path. Though they determined the beginning value of λ and the stopping criteria, it is not guaranteed that the complete solution path can be achieved by this sequence of λ .

It is clear that the sparsity of the solution $\boldsymbol{\beta}$ increases when λ is increased. This

is in line with the ADMM procedure, which produces sparser solutions as the iteration goes. Based on this observation, Hu et al. (2015) proposed to iteratively update ADMM algorithm and uses the solution at each iteration to approximate the regularization path. This saves the runtime dramatically, since now all one need to do is to perform a slightly modified version of ADMM algorithm. This method is good for the clustering purpose, however it is not accurate in estimation.

3.2.4 Connection To Constrained Lasso

The constrained lasso problem (James et al., 2013) is defined as:

$$\begin{aligned} \min_{\boldsymbol{\beta}} \quad & \frac{1}{2} \|y - X\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1 \\ \text{subject to} \quad & A\boldsymbol{\beta} = b \text{ and } C\boldsymbol{\beta} \leq d \end{aligned} \tag{3.24}$$

James et al. (2013) showed in their work that when D in (1.3) has full row rank, the generalized lasso can be transformed to a constrained lasso problem. Gaines and Zhou (2016) showed that for any arbitrary matrix D , it can be transformed to a constrained lasso problem. Assuming that $\text{rank}(D) = r$, they first conducted a single value decomposition (SVD):

$$D = U\Sigma V^T = (U_1, U_2) \begin{pmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \end{pmatrix} = U_1 \Sigma_1 V_1^T \tag{3.25}$$

Define an augmented $\tilde{D}$ matrix as:

$$\tilde{D} = \begin{pmatrix} U_1 \Sigma_1 V_1^T \\ V_2^T \end{pmatrix} \begin{pmatrix} U_1 \Sigma_1 & 0 \\ 0 & I_{p-r} \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix} = \begin{pmatrix} U_1 \Sigma_1 & 0 \\ 0 & I_{p-r} \end{pmatrix} V^T \tag{3.26}$$

and

$$\tilde{D}\boldsymbol{\beta} = \begin{pmatrix} U_1 \Sigma_1 V_1^T \\ V_2^T \end{pmatrix} \boldsymbol{\beta} = \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\gamma} \end{pmatrix} \boldsymbol{\beta} \quad (3.27)$$

They then showed that the generalized problem can be rewritten as:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} & \|\mathbf{y} - XD^+ \boldsymbol{\alpha} - XV_2 \boldsymbol{\gamma}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1 \\ \text{subject to} & \quad U_2^T \boldsymbol{\alpha} = \mathbf{0} \end{aligned} \quad (3.28)$$

which is a constrained lasso problem. The authors then introduced their algorithms to solve the constrained lasso problem. They first used the quadratic programming by decomposing the coefficients $\boldsymbol{\beta}$ into the positive part $\boldsymbol{\beta}_+$ and negative part $\boldsymbol{\beta}_-$. By adding nonnegative constraints on $\boldsymbol{\beta}_+$ and $\boldsymbol{\beta}_-$, the problem can be transformed to a quadratic programming problem and can be solved using standard quadratic program solvers. They also applied ADMM algorithm to solve (3.28), which is similar to what is shown in section 3.2.3.

Additionally, they derived a new solution path algorithm for the constrained lasso. In their path algorithm, they tracked events which lead to "kinks" along the path. The events are "an active coefficients hits 0", "an inactive coefficient turns active", "a strictly inequality constraint hits the boundary", "an inequality constraint leaves the boundary", and the corresponding monitors are described in Gaines and Zhou (2016). They also studied the degrees of freedom of such a problem. Similarly as in (Tibshirani and Taylor, 2011), based on the Stein's formula and assuming y follows a normal distribution, they gave the df as:

$$df(X\hat{\boldsymbol{\beta}}(\lambda)) = E[|\mathcal{A}| - (q + |\mathcal{Z}_I|)] \quad (3.29)$$

where $|\mathcal{A}|$ is the number of active predictors, q is the number of equality constraints and $|\mathcal{Z}_I|$ is the number of binding equality constraints.

3.3 Implementation and Computation Performance

Arnold and Tibshirani (2016) studied the efficient implementation of algorithm proposed by Tibshirani and Taylor (2011). They used the QR decomposition to solve the least square problem at each iteration and update the solution path. They proposed two strategies which correspond to a full row rank matrix D ($m \leq p$) and a general D matrix, respectively, and gave the two strategies names as "wide strategy" and "tall strategy". When D satisfies the "wide strategy", they calculate $\hat{u}$ that minimizes $\|y - D^T u\|$ using QR decomposition $D^T = RQ$, where $Q \in \mathbb{R}^{p \times p}$ is orthogonal and $R \in \mathbb{R}^{m \times p}$ takes the form $\begin{pmatrix} R1 \\ 0 \end{pmatrix}$. Similarly, by QR decomposition of $D_{-\mathcal{B}}^T$, one can get $\hat{a}$ and $\hat{b}$ that solve (3.12). Note that $D_{-\mathcal{B}^{k+1}}$ and $D_{\mathcal{B}^k}$ only differ by one row as we either remove or add one row at each iteration, thus one can update the QR decomposition of the new from the old. When D does not have full row rank or $m > p$, the "tall strategy" is applied. Again, they can first estimate $\hat{u}$ by doing QR decomposition $DPG = QR$, where $P \in \mathbb{R}^{p \times p}$ is a permutation matrix, G, Q are orthogonal and R takes the form $\begin{bmatrix} 0 & R1 \\ 0 & 0 \end{bmatrix}$. Similarly, $\hat{a}$ and $\hat{b}$ can be achieved by updating the QR decomposition from the old one.

Let N_{dual} denote the number of iterations in the generalized lasso dual path algorithm, the computational complexity of this QR decomposition method for the full solution path is given in Table 3.1.

Duan et al. (2016) also discussed the complexity of standard ADMM and augmented ADMM of the sparse fused lasso problem. Let l denote the length of the augmented variable, N_{admm} denote the number of iterations in the ADMM and augmented ADMM algorithm mentioned in section 3.2.3 and N_{chol} denote the number of Cholesky factorizations in ADMM, the computation complexity of the sparse fused lasso problem for a

Table 3.1: Computational complexity based on the QR decomposition.

Conditions	Complexity
$m \leq p, \text{rank}(D) = m$	$\mathcal{O}(m^2p + N_{dual}mp)$
$m \leq p, \text{rank}(D) < m$	$\mathcal{O}(m^2p + N_{dual}p^2)$
$m > p$	$\mathcal{O}(mp^2 + N_{dual}m^2)$

fixed λ value is given in Table 3.2. In this case, D is a $m < p$ full row rank matrix and n is the total number of observations.

Table 3.2: Computational complexity of ADMM and augADMM.

	$p \leq n$	$p > n$
standADMM	$\mathcal{O}(N_{chol}p^2n + N_{admm}p^2)$	$\mathcal{O}(N_{chol}p^3 + N_{admm}p^2)$
augADMM	$\mathcal{O}(N_{chol}p^2n + N_{admm}p^2)$	$\mathcal{O}(N_{chol}n^2p + N_{admm}max(pn, l))$

Note that the dual path algorithm provides computation complexity for the full solution path while the ADMM procedure gives complexity for a given λ value. Generally, N_{chol} is smaller than 10. Assuming the dual path algorithm and ADMM for a sequence of λ 's take the same number of iterations, it can be seen from Table 3.1 and Table 3.2 that the former method based on the dual algorithm is more efficient when $p \leq n$ for a sparse fused lasso problem. When $p > n$, dual algorithm is still less complex compared to the standard ADMM algorithm. To compare the complexity between the dual and augmented ADMM, we need to know more about n and m .

It is worth noting that Arnold and Tibshirani (2016) not only implemented the algorithm for a general D matrix, but also provided fast implementation for special cases

like fused lasso, sparse fused lasso and trend filtering problem. All the implementations are available in R package **genlasso**.

Gaines and Zhou (2016) also compared computational performance of solving the constrained lasso problem among quadratic programming, ADMM and their path following algorithm. From their numeric experiment, ADMM is more efficient than QP. It is interesting to see that it takes the path algorithm less time to calculate the full regularization path than ADMM at a fixed λ value.

3.4 Lasso Inference

Because of the sparsity property of lasso, it achieves variable selection by shrinking unimportant variables to zeros. Similarly, in the generalized lasso, unimportant structures of the coefficients, for example: differences, are shrunk to zeros. However, it is likely that in some cases lasso may produce results with low variable selection prediction (He and Lin, 2010, Devlin et al., 2003) where some of the “important” variables included in the model are false positive. Thus, when facing this kind of situations or when our interest in the data analysis is beyond prediction, it is crucial to perform statistical inference after estimation. However, statistical inferences like hypothesis tests, p-values, confidence intervals are not immediately available for lasso estimators due to the complexity of its limiting distribution. When solving the lasso problem, common algorithms do not account for the uncertainty of lasso estimators at the same time. Knight and Fu (2000) study the asymptotic properties of lasso estimators and show that:

$$\sqrt{n}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \xrightarrow{d} \arg \min_{\boldsymbol{u}} V(\boldsymbol{u}) \quad \text{as } n \rightarrow \infty, \quad (3.30)$$

where $V(\cdot)$ is a random process. Thus there is no available closed form for the limiting distribution of lasso estimators. Besides, conventional inference methods often lead to invalid results for post-selection approaches due to failure to take the stochastic property into consideration.

One common and well accepted approach to do post-inference is associated with data splitting. The entire dataset is spitted into tow subsets. The first half is used for variable selection and the second half is used for inference. The problem is that fitted models may be different if we randomly split the data into halves. In recent years, there is a growing number of literatures that discuss and propose algorithms to provide valid post-selection inferences. There are different frequentist approaches to solve the lasso inference problem. Lee et al. (2016), Lockhart et al. (2014) constructed confidence intervals or conducted significant tests by characterizing the distribution of lasso estimators conditioned on the selected model. Minnier et al. (2011), Chatterjee and Lahiri (2011) used resampling- based methods, such as permutation resampling and bootstrapping to approximate distribution of lasso estimators. Zhang and Zhang (2014) achieved valid post-inference by adjusting the bias introduced by the l_1 term in lasso. Lu et al. (2017) applied stochastic variational inequality theory in optimization in lasso inference. Lasso and its inference are also studied in a Bayesian way. In this section, we are going to further introduce and discuss some of them.

3.4.1 The Conditional Sampling Distribution Approach

Lee et al. (2016) constructed confidence intervals and significance tests for lasso estimators by studying the conditional distributions of the coefficients. Suppose $M \subset (1, 2, \dots, p)$ is a subset model which minimizes the expected error:

$$\hat{\boldsymbol{\beta}}^M = \arg \min_{\mathbf{b}^M} \mathbb{E} \|\mathbf{y} - (X_M^T X_M)^{-1} X_M^T \mathbf{b}^M\|^2, \quad (3.31)$$

and $\hat{M}$ is the model selected by lasso where all parameters $\hat{\beta}_j \neq 0$. The confidence interval $\hat{C}_j^M$ for $\hat{\beta}_j^M$ is defined as:

$$\mathbb{P}(\beta_j^M \in C_j^M | \hat{M} = M) \geq 1 - \alpha. \quad (3.32)$$

To study this distribution, they first characterized the model selection event $\{\hat{M} = M\}$ or $\{\hat{M} = M, \hat{\mathbf{s}}_M = \mathbf{s}_M\}$, where $\mathbf{s}$ represents the signs of the parameters. $\{\hat{M} = M, \hat{\mathbf{s}}_M = \mathbf{s}_M\}$ is actually a polyhedron and $\hat{\beta}_j$ can be represented as a linear transformation of the observation $\boldsymbol{\eta}^T \mathbf{y}$, where $\boldsymbol{\eta}^T = ((X_M^T X_M)^{-1} X_M^T)^T \mathbf{e}^j$, thus $\{\beta_j^M \in C_j^M | \hat{M} = M, \hat{\mathbf{s}}_M = \mathbf{s}_M\}$ can be written as $\{\boldsymbol{\eta}^T \mathbf{y} | A\mathbf{y} \leq \mathbf{b}\}$ with

$$A = \begin{pmatrix} A_0(M, \mathbf{s}) \\ A_1(M, \mathbf{s}) \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} \mathbf{b}_0(M, \mathbf{s}) \\ \mathbf{b}_1(M, \mathbf{s}) \end{pmatrix}$$

where

$$\begin{aligned} A_0(M, \mathbf{s}) &= \frac{1}{\lambda} \begin{pmatrix} X_{-M}^T (I - P_M) \\ -X_{-M}^T (I - P_M) \end{pmatrix}, \\ A_1(M, \mathbf{s}) &= -\text{diag}(X_M^T X_M)^{-1} X_M^T, \\ \mathbf{b}_0(M, \mathbf{s}) &= \begin{pmatrix} 1 - X_{-M}^T (X_M^T X_M)^{-1} X_M^T \mathbf{s} \\ 1 + X_{-M}^T (X_M^T X_M)^{-1} X_M^T \mathbf{s} \end{pmatrix}, \\ \mathbf{b}_1(M, \mathbf{s}) &= -\lambda \text{diag}(X_M^T X_M)^{-1} \mathbf{s}, \end{aligned}$$

P_M is projection matrix of X_M . To describe this conditional distribution, they further introduced $\mathbf{z} = (I_n - \mathbf{c}\boldsymbol{\eta}^T)\mathbf{y}$, which is independent of $\boldsymbol{\eta}^T\mathbf{y}$, and $\mathbf{c} = \Sigma\boldsymbol{\eta}(\boldsymbol{\eta}^T\Sigma\boldsymbol{\eta})$ assuming the observations $\mathbf{y} \sim N(\boldsymbol{\mu}, \sigma^2)$. Then, $A\mathbf{y} \leq \mathbf{b}$ is rewritten as:

$$\{A\mathbf{y} \leq \mathbf{b}\} = \mathcal{V}^-(\mathbf{z}) \leq \boldsymbol{\eta}^T\mathbf{y} \leq \mathcal{V}^+(\mathbf{z}), \quad \mathcal{V}^0(\mathbf{z}) \geq 0,$$

where $\mathcal{V}^-(\mathbf{z})$, $\mathcal{V}^+(\mathbf{z})$, $\mathcal{V}^0(\mathbf{z})$ can be written as function of $A, \mathbf{b}, \mathbf{c}$ and $\mathbf{z}$. From here, the authors proved that:

$$F_{\boldsymbol{\eta}^T\boldsymbol{\mu}, \boldsymbol{\eta}^T\Sigma\boldsymbol{\eta}}^{\mathcal{V}^-(\mathbf{z}), \mathcal{V}^+(\mathbf{z})}(\boldsymbol{\eta}^T\mathbf{y}) | \{A\mathbf{y} \leq \mathbf{b}\} \sim \text{Unif}(0, 1), \quad (3.33)$$

where $F_{\mu, \sigma^2}^{[a, b]}$ is the CDF of a truncated normal random variable with mean μ , variance σ^2 on $[a, b]$.

As $\{A\mathbf{y} \leq \mathbf{b}\}$ describes the event $\{\hat{M} = M, \hat{\mathbf{s}} = \mathbf{s}\}$ and the event of interest $\{\hat{M} = M\}$ is the union of all $\{A\mathbf{y} \leq \mathbf{b}\}$, the distribution conditioned on $\{\hat{M} = M\}$ is written as:

$$F_{\boldsymbol{\eta}^T\boldsymbol{\mu}, \boldsymbol{\eta}^T\Sigma\boldsymbol{\eta}}^{\cup_s [\mathcal{V}^-(\mathbf{z}), \mathcal{V}^+(\mathbf{z})]}(\boldsymbol{\eta}^T\mathbf{y}) | \cup_s [\{A\mathbf{y} \leq \mathbf{b}\}] \sim \text{Unif}(0, 1). \quad (3.34)$$

Finally, the confidence interval for each individual β_j^M can be obtained as:

$$C_j^M = \{\beta_j^M : \frac{\alpha}{2} \leq F_{\boldsymbol{\eta}^T\boldsymbol{\mu}, \boldsymbol{\eta}^T\Sigma\boldsymbol{\eta}}^{\cup_s [\mathcal{V}^-(\mathbf{z}), \mathcal{V}^+(\mathbf{z})]}(\boldsymbol{\eta}^T\mathbf{y}) \leq 1 - \frac{\alpha}{2}\}. \quad (3.35)$$

In this method, the authors studied the distribution of the parameters conditioned on the selection event. It provides a way to construct confidence intervals for non-zero estimators selected by lasso. However, coefficients which are estimated as zeros are not considered. In addition, the confidence intervals are built for individual β_j separately

and we cannot get a confidence region for the parameter β jointly.

Lockhart et al. (2014) proposed a test statistic “covariance test statistic” to test the significance of an additional variable entered into the lasso solutions. In multiple linear regression, it is to common to use an F statistic to test whether some component of parameters are significant by comparing the full model to the reduced model:

$$F = \frac{(SSE(R) - SSE(F))/(df_R - df_F)}{SSE(F)/df_F}, \quad (3.36)$$

where $SSE(R)$ and $SSE(F)$ are sum of squared errors of the reduced model and the full model, df_R and df_F are corresponding degrees of freedom. However, for post-selection method like lasso, $(SSE(R) - SSE(F))/(df_R - df_F)$ is not a chi-square distribution under the null. Thus the F statistic cannot be used for lasso inference and a new statistic needs to be proposed for significance tests of lasso. The “covariance test statistic” is based on the solution path of lasso obtained by LARS (Efron et al., 2004). The solution path starts with $\lambda = \infty$ and all coefficients are estimated to be zeros. When decreasing λ and at each step, a new predictor enters the path. The active set contains all $\hat{\beta}(\lambda)$'s that are non-zero. The proposed statistic can be used to perform significance test of the predictor that enters the current active set. Let $\mathcal{A}$ be the current active set before the predictor k enters at λ_k . Suppose that $\hat{\beta}(\lambda_{k+1})$ is the solution using the predictors in $\mathcal{A} \cup k$ and $\tilde{\beta}(\lambda_{k+1})$ is the solution using $\lambda = \lambda_{k+1}$ but only predictors in $\mathcal{A}$. Suppose the error variance σ^2 is known, then the test statistic is defined as:

$$T_k = (\langle \mathbf{y}, X\hat{\beta}(\lambda_{k+1}) \rangle - \langle \mathbf{y}, X_{\mathcal{A}}\tilde{\beta}(\lambda_{k+1}) \rangle) / \sigma^2. \quad (3.37)$$

They further proved that under the null hypothesis that current lasso model contains all nonzero coefficients that are significantly different from zero, the covariance test

statistic converges in distribution to an exponential random variable, that is $T \xrightarrow{d} Exp(1)$.

In the case where σ^2 is unknown, σ^2 is estimated by $\hat{\sigma}^2 = \|\mathbf{y} - X\hat{\boldsymbol{\beta}}^{LS}\|_2^2/(n-p)$. The test statistic becomes:

$$\begin{aligned} F_k &= \frac{T_k}{(\hat{\sigma}^2/\sigma^2)} \\ &= \frac{\left\langle \mathbf{y}, X\hat{\boldsymbol{\beta}}(\lambda_{k+1}) \right\rangle - \left\langle \mathbf{y}, X_{\mathcal{A}}\tilde{\boldsymbol{\beta}}(\lambda_{k+1}) \right\rangle}{\hat{\sigma}^2} \\ &\xrightarrow{d} F_{2,n-p}. \end{aligned} \tag{3.38}$$

This proposed statistic is useful in conducting significance tests for lasso predictors, but it does not provide confidence intervals for the coefficients either individually or jointly.

3.4.2 Resampling-Based Approach

Bootstrapping approach is a powerful tool in estimating statistical accuracy in terms of bias, variance, confidence intervals and so on when the sample size is small or the distribution is hard to be described. As the limiting distribution of lasso estimators are complex, it would be ideal if we could apply resampling methods like bootstrapping to get the standard errors and confidence intervals. Consider a regular lasso problem:

$$\hat{\boldsymbol{\beta}}^{Lasso} = \arg \min_{\boldsymbol{\beta}} \frac{1}{2} \|\mathbf{y} - X\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1, \tag{3.39}$$

the residuals of lasso is defined as $\mathbf{r} = \mathbf{y} - X\hat{\boldsymbol{\beta}}^{Lasso}$. In the residual bootstrapping approach, $\mathbf{r}^{boot}$ are sampled with replacement from centered $\mathbf{r}$. The new observations are

obtained from the bootstrapped residuals:

$$\mathbf{y}^{boot} = X\hat{\boldsymbol{\beta}} + \mathbf{r}^{boot}. \quad (3.40)$$

And the bootstrapped lasso estimator is calculated as:

$$\hat{\boldsymbol{\beta}}^{boot} = \arg \min_{\mathbf{u}} \|\mathbf{y}^{boot} - X^T \mathbf{u}\|_2^2 + \lambda \|\mathbf{u}\|_1. \quad (3.41)$$

However, Chatterjee and Lahiri (2010) studied the asymptotic properties of residual bootstrapped lasso estimators and proved that the residual bootstrap lasso approximation is inconsistent if the solution is sparse. And the bootstrap approach is only valid for some special cases. Chatterjee and Lahiri (2011) proposed a modified residual bootstrap lasso to construct confidence regions for the true parameters $\boldsymbol{\beta}^{True}$. In the modified residual bootstrap lasso, $\hat{\boldsymbol{\beta}}$ that are close to zero are forced to be zero. The modified lasso estimator is defined as:

$$\hat{\beta}_j^M = \hat{\beta}_j 1(|\hat{\beta}_j| \geq a_n) \quad \text{for } j = 1, 2, \dots, p, \quad (3.42)$$

where a_n satisfies:

$$a_n + (n^{-\frac{1}{2}} \log n) a_n^{-1} \rightarrow 0 \quad \text{as } n \rightarrow \infty, \quad (3.43)$$

and the modified lasso residual can be calculated as $\mathbf{r} = \mathbf{y} - X\hat{\boldsymbol{\beta}}^M$. Similarly as in the residual bootstrap lasso, $\mathbf{r}^*$ is sampled with replacement from the centered residuals. Let $\mathbf{y}^* = X\hat{\boldsymbol{\beta}}^M + \mathbf{r}^*$. Finally the modified bootstrap lasso estimator can be expressed as:

$$\hat{\boldsymbol{\beta}}^* = \arg \min_{\mathbf{u}} \|\mathbf{y}^* - X^T \mathbf{u}\|_2^2 + \lambda \|\mathbf{u}\|_1. \quad (3.44)$$

Chatterjee and Lahiri (2011) proved that $T_n^* = \sqrt{n}(\hat{\beta}^* - \hat{\beta}^*)$ is a valid approximation of $T_n = \sqrt{n}(\hat{\beta} - \beta)$. Thus confidence intervals can be built based on the distribution of T_n^* .

Camponovo (2015) further proposed a modified pair bootstrap lasso method. Let $z_j = (y_j, x_j^T)^T$, and random sample $(z_1^*, \dots, z_p^*)$ are sampled with replacement from $(z_1, \dots, z_p)$. The modified pair bootstrap lasso estimator is defined as:

$$\hat{\beta}_{pair}^* = \arg \min_{\mathbf{u}} \sum_{i=1}^p (y_i^* - u^T x_i^*)^2 + \lambda \sum_{i=1}^p |u_i - \hat{\beta}_i^{LS} 1(|\hat{\beta}_i^{LS}| \leq a_n)|, \quad (3.45)$$

where a_n satisfies 3.43 and $\hat{\beta}^{LS}$ is the least square solution. Similarly, $T_{pair}^* = \sqrt{n}(\hat{\beta}_{pair}^* - \hat{\beta}^{LS})$ is a valid approximation to T_n .

In Chapter 4, we apply the modified residual lasso to get the confidence intervals for the generalized lasso.

3.4.3 Bias Correction Approach

Zhang and Zhang (2014) proposed a method to obtain confidence intervals for low dimensional parameters for high dimensional problems. The idea is based on correcting the bias in lasso estimators $\hat{\beta}$. The algorithm starts with an initial parameter vector β^{init} . $\beta^{(init)}$ and a consistent estimate $\hat{\sigma}$ are obtained by solving the scaled lasso (Sun and Zhang, 2010):

$$\{\hat{\beta}^{(init)}, \hat{\sigma}\} = \arg \min_{\mathbf{b}, \sigma} \{\|\mathbf{y} - X\mathbf{b}\|_2^2 / 2\sigma n + \sigma/2 + \lambda_0 \|\mathbf{b}\|_1\}. \quad (3.46)$$

They use a vector $\mathbf{z}_j$ to approximate $x_j^\perp$, which is the projection of $\mathbf{x}_j$ to the orthogonal complement of X_{-j} , which is the column space of design matrix that does not contain

$\mathbf{x}_j$. $\mathbf{z}$ can be regarded as the residuals of lasso and it can be calculated as:

$$\mathbf{z}_j = \mathbf{x}_j - X_{-j}\hat{\boldsymbol{\gamma}}_j, \quad \hat{\boldsymbol{\gamma}}_j = \arg \min_{\mathbf{b}} \{\|\mathbf{x}_j - X_{-j}\mathbf{b}\|_2^2/2n + \lambda_j \|\mathbf{b}\|_1\}. \quad (3.47)$$

Then the low dimensional projection estimator (LDPE) for β_j is expressed as:

$$\hat{\beta}_j = \hat{\beta}_j^{(init)} + \mathbf{z}_j^T(\mathbf{y} - X\hat{\boldsymbol{\beta}}^{(init)})/\mathbf{z}_j^T\mathbf{x}_j. \quad (3.48)$$

Finally, the confidence interval for each $\hat{\beta}_j$ can be calculated as:

$$\hat{\beta}_j \pm \Phi^{-1}(1 - \frac{\alpha}{2})\hat{\sigma}\tau_j, \quad (3.49)$$

where $\tau_j = \|\mathbf{z}_j\|_2/|\mathbf{z}_j^T\mathbf{x}_j|$ is the noise factor.

The main contribution of this method is that it does not require the uniform signal strength condition. The uniform signal strength condition requires the non-zero estimators to be greater than a noise level defined as $\mathcal{C}\sigma\sqrt{(2/n)\log(p)}$ with $\mathcal{C} \geq 1/2$. This condition is important in terms of variable selection consistency. While in many real cases the condition cannot be satisfied, this low dimensional projection approach does not need to consider it. There are other related work (Bühlmann et al., 2013, Van de Geer et al., 2014, Javanmard and Montanari, 2014) that make efforts in conducting confidence intervals for high dimensional problems by adjusting for the bias and we are not going to discuss the details here.

3.4.4 Stochastic Variational Inequality

Lu et al. (2017) applied stochastic variational inequality to solve optimization problems to build confidence intervals for lasso estimators. The population lasso problem is defined

as:

$$\min_{\boldsymbol{\beta}} (E(\mathbf{y} - \sum_{j=1}^p \beta_j X_j)^2 + \lambda \sum_{j=1}^p |t_j|), \quad (3.50)$$

and the regular lasso we usually solve as in equation 3.39 is referred as the sample average approximation (SAA) of form 3.50. The SAA lasso estimator is $\hat{\boldsymbol{\beta}}$ and the estimators for the population lasso is $\tilde{\boldsymbol{\beta}}$. To study the asymptotic distribution of lasso estimators, they transform the population lasso and SAA lasso to a normal map. Define a convex set $\mathcal{S} = \{(\boldsymbol{\beta}, \mathbf{t}) : t_j - \beta_j \geq 0, t_j \beta_j \geq 0, j = 1, 2, \dots, p\}$. Define a function F :

$$F(\boldsymbol{\beta}, \mathbf{t}, X, \mathbf{y}) = \begin{pmatrix} -2(\mathbf{y} - \sum_{j=1}^p \beta_j X_j)X_1 \\ -2(\mathbf{y} - \sum_{j=1}^p \beta_j X_j)X_2 \\ \vdots \\ -2(\mathbf{y} - \sum_{j=1}^p \beta_j X_j)X_p \\ \lambda \mathbf{e}_p \end{pmatrix}, \quad (3.51)$$

then the population lasso can be written as the variational inequality:

$$-f_0(\boldsymbol{\beta}, \mathbf{t}) \in N_S(\boldsymbol{\beta}, \mathbf{t}), \quad (3.52)$$

where f_0 is the expectation of function F , and $N_S(\boldsymbol{\beta}, \mathbf{t})$ is the normal cone to S . Define the normal map as:

$$(f_0)_S(\mathbf{z}) = f_0\{\prod_S(\mathbf{z}) + \mathbf{z} - \prod_S(\mathbf{z})\} \quad \text{for a vector } \mathbf{z}, \quad (3.53)$$

and thus solving the population lasso is equivalent as solving:

$$(f_0)_S(\mathbf{z}) = 0. \quad (3.54)$$

The authors showed that $\mathbf{z}_0 = (\tilde{\boldsymbol{\beta}}, \tilde{\mathbf{t}}) - f_0(\tilde{\boldsymbol{\beta}}, \tilde{\mathbf{t}})$ is a solution to 3.54 and further studied its asymptotic properties. Let $\mathbf{z}_N = (\hat{\boldsymbol{\beta}}, \hat{\mathbf{t}}) - f_0(\hat{\boldsymbol{\beta}}, \hat{\mathbf{t}})$ be the sample version of $\mathbf{z}_0$, and then:

$$\sqrt{N}\Phi_N(\mathbf{z}_N)(\mathbf{z}_N - \mathbf{z}_0) \rightarrow N(0, \Sigma_0), \quad (3.55)$$

It is further proved that if Σ_N^1 is non-singular with Σ_N^1 satisfies $\Sigma_N = \begin{pmatrix} \Sigma_N^1 & 0 \\ 0 & 0 \end{pmatrix}$, the asymptotic distribution can be described as:

$$\sqrt{N} \begin{pmatrix} (\Sigma_N^1)^{-1/2} & 0 \\ 0 & I_p \end{pmatrix} \Phi_N(\mathbf{z}_N)(\mathbf{z}_N - \mathbf{z}_0) \rightarrow N(0, I_{p+1}) \times 0., \quad (3.56)$$

And if Σ_N^1 is singular, decompose Σ_N^1 as $\Sigma_N^1 = U_N^T \Delta U_N$, then:

$$N(\Phi_N(\mathbf{z}_N)(\mathbf{z}_N - \mathbf{z}_0))^T \begin{pmatrix} (U_N)_1^T D_N^{-1} (U_N)_1 & 0 \\ 0 & 0 \end{pmatrix} (\Phi_N(\mathbf{z}_N)(\mathbf{z}_N - \mathbf{z}_{[0]}))^T \rightarrow \chi_l^2, \quad (3.57)$$

and

$$N(\Phi_N(\mathbf{z}_N)(\mathbf{z}_N - \mathbf{z}_{[0]}))^T \begin{pmatrix} (U_N)_2^T (U_N)_2 & 0 \\ 0 & I_p \end{pmatrix} (\Phi_N(\mathbf{z}_N)(\mathbf{z}_N - \mathbf{z}_{[0]}))^T \rightarrow 0, \quad (3.58)$$

where D_N is the upper left of matrix Δ with diagonals greater than some value, l is the number of rows of D_N , $(U_N)_1$ is the first l rows of U_N and $(U_N)_2$ are the rest rows in U_N . Using this distribution, confidence intervals of $\mathbf{z}_0$ can be built and confidence intervals

of $\tilde{\boldsymbol{\beta}}$ can be calculated by doing back transformation for a fixed λ value using:

$$\tilde{\beta}_j = \begin{cases} (z_0)_{j+1} - \lambda & \text{if } (z_0)_{j+1} > \lambda \\ 0 & \text{if } -\lambda \leq (z_0)_{j+1} \leq \lambda \\ (z_0)_{j+1} + \lambda & \text{if } (z_0)_{j+1} < -\lambda \end{cases} \quad (3.59)$$

In this approach, the authors not only built confidence intervals for individual population lasso estimators, but also constructed a confidence region. By studying the relationship between the population lasso estimators and the true parameter values, they also built confidence intervals for $\boldsymbol{\beta}^{True}$. It is noteworthy that because of involving large matrix calculation and optimization with a large number of parameters, the computation of this approach can be very expensive.

3.4.5 Bayesian Lasso

In addition to frequentist approaches, efforts were made in Bayesian approaches to achieve sparse solutions and variable selection (Figueiredo, 2003, Bae and Mallick, 2004, Yuan and Lin, 2005). Park and Casella (2008) proposed a Bayesian lasso approach by using the Laplace prior. The Laplace prior of $\boldsymbol{\beta}$ conditional on σ^2 is:

$$\pi(\boldsymbol{\beta}|\sigma^2) = \prod_{j=1}^p \frac{\lambda}{2\sqrt{\sigma^2}} \exp\left(-\frac{\lambda|\beta_j|}{\sqrt{\sigma^2}}\right). \quad (3.60)$$

As the Laplace distribution can be expressed as a mixture of normal and exponential,

they represented the hierarchical Bayesian model as:

$$\begin{aligned}
\mathbf{y}|\mu, X, \boldsymbol{\beta}, \sigma^2 &\sim N_n(\mu \mathbf{1}_n + X\boldsymbol{\beta}, \sigma^2 \mathbf{I}_n) \\
\boldsymbol{\beta}|\sigma^2, \tau_1^2, \dots, \tau_p^2 &\sim N_p(\mathbf{0}_p, \sigma^2 D_{\boldsymbol{\tau}}) \\
D_{\boldsymbol{\tau}} &= \text{diag}(\tau_1^2, \dots, \tau_p^2) \\
\sigma^2, \tau_1^2, \dots, \tau_p^2 &\sim \pi(\sigma^2) d\sigma^2 \prod_{j=1}^p \frac{\lambda^2}{2} \exp\left(-\frac{\lambda^2 \tau_j^2}{2} d\tau_j^2\right), \\
\pi(\sigma^2) &= 1/\sigma^2 \\
\sigma^2, \tau_1^2, \dots, \tau_p^2 &> 0
\end{aligned} \tag{3.61}$$

and a flat prior can be used on μ . To choose the tuning parameter λ , the prior 3.62 can be used in the hierarchical model:

$$\pi(\lambda^2) = \frac{\delta^r}{\Gamma r} (\lambda^2)^{r-1} e^{-\delta \lambda^2}, \quad \lambda^2 > 0 (r > 0, \delta > 0). \tag{3.62}$$

The posterior mode serves as the lasso estimator. This Bayesian lasso is easy to implement using Gibbs sampling as the priors are conjugate. And as it gives the posterior distribution, we can easily achieve the confidence sets.

3.4.6 Lasso inference to generalized lasso inference

When can we take advantage of the lasso inference methods and apply them to do hypothesis tests and obtain confidence intervals for the generalized lasso? First of all, the generalized lasso needs to be reduced to a regular lasso. As is mentioned earlier in this Chapter, both Tibshirani and Taylor (2011) and Duan et al. (2016)'s transformations only apply to problems where $D_{m \times p}$ is a full row rank matrix. In applications when

we have more constraints than the dimension of the parameters, we cannot transform the generalized lasso problem to lasso to solve it. In Chapter 4, we will discuss our proposed algorithm, which reduces any generalized lasso to a lasso problem when $m \geq p$ with $\text{rank}(D) \leq p$. Secondly, after we reduce the generalized lasso to a regular lasso problem, we need to get the confidence intervals of lasso estimators jointly or know about the covariance matrix of $\hat{\beta}$ in order to do back transformation to do inference on the generalized lasso estimates. However, many of current methods only construct confidence intervals on individual lasso estimators. In Chapter 4, we will introduce a method to get confidence intervals for the generalized lasso estimators by applying the modified bootstrap lasso approach.

Chapter 4

Proposed Algorithm And Confidence Intervals For The Generalized Lasso

4.1 Proposed Algorithm

To consider cases such as the 2D fused lasso and any other situation with $m > p$, we develop a new approach to compute the full solution path. Inspired by the idea that we can apply existing lasso solvers to solve the generalized problem if it can be transformed to lasso, we introduce our algorithm that can reduce any generalized lasso to a regular lasso problem, even in the case that we have a “long” penalty matrix D in problem (1.3).

4.1.1 D is full rank

We start from the simplest case where D is a $p \times p$ full rank square matrix and X is a general design matrix. Since D is invertible, problem (1.3) can be reparameterized as

$$\min_{\boldsymbol{\theta}} \frac{1}{2} \|\mathbf{y} - XD^{-1}\boldsymbol{\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_1, \quad (4.1)$$

where $\boldsymbol{\theta} = D\boldsymbol{\beta}$. Thus, standard lasso solvers, such as LARS, can be used to solve problem (4.1) and provide complete solution paths. The coefficients can then be directly obtained via back transformation: $\hat{\boldsymbol{\beta}} = D^{-1}\hat{\boldsymbol{\theta}}$.

4.1.2 D has full column rank

Now, suppose that D is an $m \times p$ matrix and $\text{rank}(D) = p$. We still would like to apply a similar transformation as above to get a lasso problem. To achieve this, we first construct an $m \times m$ matrix $D^* \in \mathbb{R}^{m \times m} = (D, D^c)$, with D^c being a basis of the orthogonal complement space, so that $\text{space}(\mathcal{C}(D^*)) = \mathbb{R}^m$, where $\mathcal{C}(D^*)$ denotes the column space of D^* . Let $\boldsymbol{\theta} \in \mathbb{R}^m = (D, D^c) \times \begin{bmatrix} \boldsymbol{\beta} \\ \boldsymbol{\gamma} \end{bmatrix} = D\boldsymbol{\beta} + D^c\boldsymbol{\gamma}$, where $\boldsymbol{\gamma}$ is a set of

nuisance parameters, thus $\begin{bmatrix} \boldsymbol{\beta} \\ \boldsymbol{\gamma} \end{bmatrix} = (D, D^c)^{-1}\boldsymbol{\theta}$. The problem is reformed as:

$$\min_{\boldsymbol{\beta}, \boldsymbol{\gamma}} \frac{1}{2} \|\mathbf{y} - (X, 0)(D, D^c)^{-1}\boldsymbol{\theta}\|_2^2 + \lambda \|D\boldsymbol{\beta}\|_1. \quad (4.2)$$

Note that

$$\begin{cases} P_D\boldsymbol{\theta} = D\boldsymbol{\beta} & \text{and} \\ P_{D^c}\boldsymbol{\theta} = D^c\boldsymbol{\gamma}, \end{cases}$$

where $P_D\boldsymbol{\theta}$ represents the projections onto $\mathcal{C}(D)$, and $P_{D^c}\boldsymbol{\theta}$ is the projection onto the orthogonal space. Thus, the problem above is the same as

$$\min_{\boldsymbol{\theta}} \frac{1}{2} \|\mathbf{y} - (X, 0)(D, D^c)^{-1}\boldsymbol{\theta}\|_2^2 + \lambda \|P_D\boldsymbol{\theta}\|_1$$

where $\boldsymbol{\theta} = D\boldsymbol{\beta} + D^c\boldsymbol{\gamma}$. The problem can be further re-parameterized as:

$$\min_{\boldsymbol{\theta}} \frac{1}{2} \|\mathbf{y} - (X, 0)(D, D^c)^{-1}P_D\boldsymbol{\theta}\|_2^2 + \lambda \|P_D\boldsymbol{\theta}\|_1 \quad (4.3)$$

since

$$\begin{cases} (X, 0)(D, D^c)^{-1}P_DD = (X, 0)(D, D^c)^{-1}D & \text{and} \\ (X, 0)(D, D^c)^{-1}P_DD^c = 0. \end{cases}$$

Note that if we force $\boldsymbol{\gamma} = 0$ in equation (4.3), then $\boldsymbol{\theta} = D\boldsymbol{\beta} = P_D\boldsymbol{\theta}$, and the problem would become:

$$\min_{\boldsymbol{\theta}} \frac{1}{2} \|\mathbf{y} - (X, 0)(D, D^c)^{-1}\boldsymbol{\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_1, \quad (4.4)$$

and problem (4.4) can be solved by standard lasso algorithms.

To force $\boldsymbol{\gamma} = 0$, we set an "infinite" penalty on $\|P_{D^c}\boldsymbol{\theta}\|_2^2$. We accomplish this by appending pseudo y values and pseudo observations to the responses and design matrix. Specifically we modify (4.4) to:

$$\min_{\boldsymbol{\theta}} \frac{1}{2} \|\mathbf{y}_{new} - X_{new}\boldsymbol{\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_1, \quad (4.5)$$

where

$$\mathbf{y}_{new} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix}, \quad X_{new} = \begin{bmatrix} (X, 0)(D, D^c)^{-1} \\ cP_{D^c} \end{bmatrix},$$

where c is a large constant relative to the value of λ to ensure the "infinite" penalty on $\|P_{D^c}\boldsymbol{\theta}\|_2^2$.

4.1.3 A general matrix D

In section (4.1.2), we provided an algorithm to transform the problem when $\text{rank}(D) = p$. More generally, let D be any $m \times p$ matrix, so that it follows that $\text{rank}(D) \leq p$. If $\text{rank}(D) < p$, we add rows to D and construct a full column rank matrix L . These rows can be chosen arbitrarily as any basis to complete the p -dimensional space. For example, in a 2D fused lasso, the matrix D constructs all pairwise differences in coefficients. In order to make $\text{rank}(D) = p$, we can add a row corresponding to β_1 , or any other β_j , for example. Suppose, we add m_2 rows to D to make it a full column rank matrix. Now we are in the case of $L \in \mathbb{R}^{m^* \times p}$ and $\text{rank}(L) = p$, where $m^* = m + m_2$. If $m^* = p$, then we are back to the case of full invertibility as in (4.1.1), and we proceed from there. Otherwise we have $m^* > p$, which is the case most often in practice. Let $\boldsymbol{\theta} = (L, L^c) \times \begin{bmatrix} \boldsymbol{\beta} \\ \boldsymbol{\gamma} \end{bmatrix}$, and

thus $\begin{bmatrix} \boldsymbol{\beta} \\ \boldsymbol{\gamma} \end{bmatrix} = (L, L^c)^{-1} \boldsymbol{\theta}$. Analogous to (4.1.2), we now form a generalized lasso problem as:

$$\min_{\boldsymbol{\beta}, \boldsymbol{\gamma}} \frac{1}{2} \|\mathbf{y} - (X, 0)(L, L^c)^{-1} \boldsymbol{\theta}\|_2^2 + \lambda \|L\boldsymbol{\beta}\|_1. \quad (4.6)$$

So as before, we solve:

$$\min_{\boldsymbol{\theta}} \frac{1}{2} \|\mathbf{y}_1 - X_1 \boldsymbol{\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_1, \quad (4.7)$$

where

$$\mathbf{y}_1 = \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix}, \quad X_1 = \begin{bmatrix} (X, 0)(L, L^c)^{-1} \\ cP_{L^c} \end{bmatrix}$$

However, now $\boldsymbol{\theta} \in \mathbb{R}^{m^*}$ and only the first m elements represent $D\boldsymbol{\beta} + D^c\boldsymbol{\gamma}$, while the others complete the space. Thus, we need to leave the final m_2 elements unpenalized.

To do this, we project out the added dimensions and solve:

$$\min_{\boldsymbol{\theta}} \frac{1}{2} \|\mathbf{y}_{new} - X_{new} \boldsymbol{\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_1, \quad (4.8)$$

where

$$\mathbf{y}_{new} = \begin{bmatrix} (I - P)\mathbf{y} \\ \mathbf{0} \end{bmatrix}, \quad X_{new} = \begin{bmatrix} (I - P)(X, 0)(L, L^c)^{-1} \\ cP_{L^c} \end{bmatrix},$$

and P is the projection matrix corresponding to the last m_2 columns of $(X, 0)(L, L^c)^{-1}$.

We can now get the entire solution path for $\boldsymbol{\theta}$. Let $\begin{bmatrix} \tilde{\boldsymbol{\beta}} \\ \tilde{\boldsymbol{\gamma}} \end{bmatrix} = (L, L^c)^{-1} \tilde{\boldsymbol{\theta}}$, where $\tilde{\boldsymbol{\theta}}$ is the solution for a chosen λ . However, $\tilde{\boldsymbol{\beta}}$ does not estimate $\boldsymbol{\beta}$, unless $\text{rank}(D) = p$ originally. Since we projected out certain terms to leave unpenalized, we now must estimate those coefficients. So let $\tilde{\mathbf{y}} = \mathbf{y} - X\tilde{\boldsymbol{\beta}}$ and perform least square regression using the last m_2 columns of $(X, 0)(L, L^c)^{-1}$. This gives the last $m^* - m$ elements of $\hat{\boldsymbol{\theta}}$. Then we get $\begin{bmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\boldsymbol{\gamma}} \end{bmatrix} = (L, L^c)^{-1} \hat{\boldsymbol{\theta}}$.

4.2 Generalized Lasso Inference

Statistical inference is a crucial step after estimation. Conventional inference methods often lead to invalid post-selection inference, and it remains challenging to construct confidence sets or perform hypothesis tests on lasso type estimators due to the complexity of their limiting distributions. In recent years, many efforts have been made to study the post-selection inference for lasso estimates (Lee et al., 2013, Lockhart et al., 2014, Zhang and Zhang, 2014, Minnier et al., 2011, Chatterjee and Lahiri, 2010, 2011, Camponovo, 2015, Lu et al., 2017) . Since we solve the generalized lasso problem by re-parameterizing

it to a regular lasso, we can take advantage of current inference methods on lasso estimators and adapt them for use in the generalized lasso problem. In particular, we apply the modified residual bootstrapped lasso approach to form confidence intervals for the parameters. However, we stress that other approaches are also possible due to the reparameterization approach.

4.2.1 Modified Residual Bootstrap Lasso

Considering a regular lasso problem in the form of (3.39), let $\hat{\boldsymbol{\beta}}$ denote the lasso estimator for a fixed λ value. Chatterjee and Lahiri (2010) discussed the asymptotic properties of the standard residual bootstrap lasso estimators and show that they fail to approximate the distribution of lasso estimators. Chatterjee and Lahiri (2011) later proposed a modified residual bootstrap approach which consistently approximates the distribution of $T = \sqrt{n}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})$ by forcing some components of $\hat{\boldsymbol{\beta}}$ to be exact zeros if they are close to zero. Then a confidence set can be built at a given α level.

In the generalized lasso, as discussed in Section 4.1, we re-parameterize the problem to a regular lasso as (4.8):

$$\min_{\boldsymbol{\theta}} \frac{1}{2} \|\mathbf{y}_{new} - X_{new} \boldsymbol{\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_1,$$

where

$$\mathbf{y}_{new} = \begin{bmatrix} (I - P)\mathbf{y} \\ \mathbf{0} \end{bmatrix}, \quad X_{new} = \begin{bmatrix} (I - P)(X, 0)(L, L^c)^{-1} \\ cP_{L^c} \end{bmatrix}.$$

The key idea is that by applying the residual bootstrap method to the lasso problem in (4.8), we construct confidence intervals for $\boldsymbol{\theta}$ jointly, and thus construct confidence intervals for the parameters $\boldsymbol{\beta}$ via back transformation. Again, let $\tilde{\boldsymbol{\theta}}$ denote the lasso

estimator of $\boldsymbol{\theta}$ from equation (4.8) for a fixed λ value. The sample size of the new lasso problem is n_L and the size of parameter $\boldsymbol{\theta}$ is p_L . As described by Chatterjee and Lahiri (2011), a sequence a_n was introduced in the modified method such that

$$a_n + (n_L^{-\frac{1}{2}} \log n_L) a_n^{-1} \rightarrow 0 \quad \text{as } n_L \rightarrow \infty. \quad (4.9)$$

Let $\tilde{\boldsymbol{\theta}}^M$ denote the modified lasso estimator and each component of $\tilde{\boldsymbol{\theta}}^M$ satisfies:

$$\tilde{\theta}_j^M = \tilde{\theta}_j 1(|\tilde{\theta}_j| \geq a_n) \quad \text{for } j = 1, 2, \dots, p_L, \quad (4.10)$$

where a_n will take the form $a_n = cn_L^{-\delta}$, for $c \in (0, \infty)$ and $\delta \in (0, \frac{1}{2})$. This modification shrinks some of the lasso estimators to zero if they are not greater than the threshold a_n . Given the modified lasso estimators, the corresponding lasso residuals $\mathbf{r} = (r_1, r_2, \dots, r_{n_L})^T$ are calculated as:

$$\mathbf{r} = \mathbf{y}_{new} - \mathbf{x}_{new}^T \tilde{\boldsymbol{\theta}}^M. \quad (4.11)$$

Note that the regular lasso problem (4.8) we solve is equivalent to:

$$\min_{\boldsymbol{\theta}} \frac{1}{2} \|\mathbf{y}_{new,1} - X_{new,1} \boldsymbol{\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_1 + c \|\mathbf{0} - P_{L^c} \boldsymbol{\theta}\|_2^2, \quad (4.12)$$

where

$$\mathbf{y}_{new,1} = (I - P)\mathbf{y}, \quad X_{new,1} = (I - P)(X, 0)(L, L^c)^{-1}.$$

The l_2 norm in (4.12) is used to force the nuisance parameters $\boldsymbol{\gamma}$ to be zeros. And the $\mathbf{0}$ observations and extra rows in the design matrix are added to implement the constraint that $P_{L^c} = 0$. As we are actually solving the lasso problem $\min_{\boldsymbol{\theta}} \frac{1}{2} \|\mathbf{y}_{new,1} - X_{new,1} \boldsymbol{\theta}\|_2^2 +$

$\lambda\|\boldsymbol{\theta}\|_1$, the first n components of $\mathbf{r}$ are ‘real’ residuals from the lasso problem and rest are exactly zeros.

The bootstrap sample, t , starts with a residual set $\{e_1^{(t)}, \dots, e_{n_L}^{(t)}\}$. Its first elements are randomly selected with replacement from $\{r_i - n_L^{-1} \sum_1^n r_i, \dots, r_n - n^{-1} \sum_1^n r_i\}$, $i = 1, 2, \dots, n$, and, in our case, its last $n_L - n$ elements are zeros. From $\mathbf{e}^{(t)}$, we define:

$$\mathbf{y}_{new}^{(t)} = \begin{pmatrix} x_{new,1}^T \tilde{\boldsymbol{\theta}}^M \\ \mathbf{0} \end{pmatrix} + \mathbf{e}^{(t)}. \quad (4.13)$$

Then the modified bootstrap lasso estimators are given via:

$$\boldsymbol{\theta}^{(t)} = \arg \min_{\mathbf{u}} \|\mathbf{y}_{new}^{(t)} - x_{new}^T \mathbf{u}\|_2^2 + \lambda \|\mathbf{u}\|_1. \quad (4.14)$$

Thus, we obtain the bootstrap statistic:

$$\mathbf{T}^{(t)} = \sqrt{n_L}(\boldsymbol{\theta}^{(t)} - \tilde{\boldsymbol{\theta}}^M). \quad (4.15)$$

Now $\mathbf{T}^{(t)}$ is the bootstrap version of $\mathbf{T} = \sqrt{n_L}(\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta})$, and Chatterjee and Lahiri (2011) proved that its distribution provides a valid approximation to the distribution of $\mathbf{T}$. From $\mathbf{T}^{(t)}$, we are able to construct confidence intervals for each component in $\boldsymbol{\theta}$:

$$\{\tilde{\theta}_j + n_L^{-\frac{1}{2}} T_j^{(t)}(\frac{\alpha}{2}), \tilde{\theta}_j + n_L^{-\frac{1}{2}} T_j^{(t)}(1 - \frac{\alpha}{2})\} \quad \text{for } j = 1, 2, \dots, p_L \quad (4.16)$$

where $T_j^{(t)}(\frac{\alpha}{2})$ and $T_j^{(t)}(1 - \frac{\alpha}{2})$ are the $\frac{\alpha}{2}$ and $1 - \frac{\alpha}{2}$ quantiles of the bootstrap distribution of $T_j^{(t)}$ for $j = 1, \dots, p_L$. Note that $\boldsymbol{\theta}$ is $D\boldsymbol{\beta}$ as $\boldsymbol{\gamma}$'s are estimated to be $\mathbf{0}$. For example, in the setting of the 2D fused lasso, $\tilde{\boldsymbol{\theta}}$ represents estimates of the differences between coefficients in the neighborhood, and the confidence intervals can be further studied to

determine whether the differences are significant.

4.2.2 Confidence Intervals For The Original Parameters

In most instances, interest focuses on the original parameters, β , so we develop confidence intervals for these parameters. As is discussed in section 4.1.3, we get $\begin{pmatrix} \tilde{\beta} \\ \hat{\gamma} \end{pmatrix} = (L, L^c)^{-1} \tilde{\theta}$,

which are estimates of $\begin{pmatrix} \beta \\ \gamma \end{pmatrix}$ with bias in β because some components are projected out.

As discussed, we then solve a least square problem to complete the missing component, to get $\hat{\theta}$ and $\begin{pmatrix} \hat{\beta} \\ \hat{\gamma} \end{pmatrix} = (L, L^c)^{-1} \hat{\theta}$. Denote the last $m^* - m$ elements of θ as β_1 , and we estimate it by the least square estimator $\hat{\beta}_1$, where $\tilde{\beta}$ and $\hat{\beta}_1$ are independent. Meanwhile, we have $\hat{\beta} = \tilde{\beta} + \hat{\beta}_1$. Via back transformation and the modified bootstrap lasso, we can get

$$\mathbf{S}^{(t)} = (L, L^c)^{-1} \sqrt{n}(\theta^{(t)} - \tilde{\theta}), \quad (4.17)$$

whose distribution then approximates the distribution of $\sqrt{n}(\begin{pmatrix} \tilde{\beta} \\ \hat{\gamma} \end{pmatrix} - \begin{pmatrix} \beta^* \\ \gamma \end{pmatrix})$. Since γ is not of interest, we define $\mathbf{S}_1^{(t)}$ as the component of $\mathbf{S}^{(t)}$ that characterizes β only:

$$\mathbf{S}_1^{(t)} = \sqrt{n}(\beta^{(t)} - \tilde{\beta}). \quad (4.18)$$

We are interested in $\hat{\beta} = \tilde{\beta} + \hat{\beta}_1$, however, the bootstrapped statistic $\mathbf{S}_1^{(t)}$ does not account for the variability introduced by estimating $\hat{\beta}_1$. Our target distribution is $\sqrt{n}(\hat{\beta} -$

β), which we now write as

$$\sqrt{n}(\hat{\beta} - \beta) = \sqrt{n}(\tilde{\beta} - \beta^*) + \sqrt{n}(\hat{\beta}_1 - \beta_1), \quad (4.19)$$

and we can use this decomposition to construct this distribution. For each bootstrap draw, we randomly sample a $\beta_1^{(t)}$ from a normal distribution $N(\hat{\beta}_1, \widehat{Var}(\beta_1))$, where $\hat{\beta}_1$ and $\widehat{Var}(\beta_1)$ are directly obtained due to the linear regression. Let $\beta_1^{(t)}$ denote the sampling distribution of β_1 , and as $\hat{\beta}_1$ is orthogonal to $\tilde{\beta}$, we write

$$\mathbf{T}_1^{(t)} = \mathbf{S}_1^{(t)} + \sqrt{n}(\beta_1^{(t)} - \hat{\beta}_1), \quad (4.20)$$

hence $\mathbf{T}_1^{(t)}$ is a valid approximation of $\mathbf{T}_1 = \sqrt{n}(\hat{\beta} - \beta)$.

Finally, the confidence interval for each individual generalized lasso parameter is defined as:

$$\{\hat{\beta}_j + n^{-\frac{1}{2}}T_{1,j}^{(t)}(\frac{\alpha}{2}), \hat{\beta}_j + n^{-\frac{1}{2}}T_{1,j}^{(1)}(1 - \frac{\alpha}{2})\} \quad \text{for } j = 1, 2, \dots, p, \quad (4.21)$$

where $T_{1,j}^{(t)}(\frac{\alpha}{2})$ and $T_{1,j}^{(1)}(1 - \frac{\alpha}{2})$ are the $\frac{\alpha}{2}$ and $1 - \frac{\alpha}{2}$ quantiles of the bootstrap distribution of $T_{1,j}^{(t)}$ for $j = 1, \dots, p$.

Chapter 5

Simulation and Real Data Analysis

5.1 Simulation Study

Here we run a simulation study with a two-dimensional spatially varying coefficients regression (Gelfand et al., 2003), where the parameters, β , are expected to vary at different locations. In this simulation study, the coefficients are varied at different grids cells.

Assume that we have a two-dimensional grid in our problem: to smooth the 2D grid surface, we set constraints on the absolute value of differences of β 's that are adjacent. Thus the 2D spatially varying coefficients problem can be considered as a 2D fused lasso problem. In terms of convenience, we use notations in graphs when describing the problem. Let $\mathcal{G}$ denote the 2D grid, we can write the problem as:

$$\min_{\beta} \|\mathbf{y} - X\beta\|_2^2 + \lambda \sum_{i,j \in \mathcal{E}} |\beta_i - \beta_j|, \quad (5.1)$$

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ are nodes in the graph representing grid cells, and $\mathcal{E}$ are edges on

the grid. Each edge contains β_i and β_j that are adjacent on the grid. The D matrix in the generalized lasso corresponding to this problem is:

$$D_k = (0, \dots, 1, \dots, -1, \dots, 0), \quad (5.2)$$

where D_k is the k^{th} row of D , 1 and -1 are the i^{th} and j^{th} element of this row, β_i and β_j are neighbors connected by edges.

In this simulation study, we build a 7×7 grid. The least square problem has one intercept, four predictors and five observations at each location. In the design matrix and for each slope, predictors are independently generated from a normal distribution with mean 0 and variance 5. We have a spatially varying coefficients model with

$$Cov(\beta_{ij}, \beta_{ik}) = 0.8^{d_{jk}} \quad (5.3)$$

where $i = 0, 1, 2, 3, 4$ for an intercept and 4 predictors, d_{jk} is the distance between locations j and k . To compute the distance, we consider the surface as a 7×7 lattice and calculate the euclidean distance. After standardizing and centering X , the response variable is generated by $y \in \mathbb{R}^n = X\beta + \mathbf{e}$, $\mathbf{e} \stackrel{iid}{\sim} N(0, 1)$.

The matrix D is the adjacency matrix which constructs the differences between all neighboring sites for each of the 5 coefficients. The dimension of D is 420×245 and $\text{rank}(D) = 240 < 245$, as it only describes the contrasts between neighboring coefficients within the intercept or each slope without anchoring the parameter vector. To make it a full column rank matrix, we add a 5×245 matrix D_1 to D : each row of D_1 defines the intercept or one of the four predictors at the first location. For example, the first row of D_1 : $D_{11} = (1, 0, 0, \dots)^T$ corresponds to β_{01} which is the intercept at the first location

on the grid. Thus $L = \begin{bmatrix} D_1 \\ D \end{bmatrix}$ has full column rank. As is discussed earlier, the lasso problem that we solve is:

$$\min_{\boldsymbol{\beta}} \frac{1}{2} \|\mathbf{y}_{new} - X_{new} \boldsymbol{\theta}\|^2 + \lambda \|\boldsymbol{\theta}\|_1,$$

where

$$\mathbf{y}_{new} = \begin{bmatrix} (I - P_1) \mathbf{y} \\ \mathbf{0} \end{bmatrix}, \quad X_{new} = \begin{bmatrix} (I - P_1)(X, 0)(L, L^c)^{-1} \\ cP_{L^c} \end{bmatrix},$$

and $P_1 = X_1(X_1 X_1^T)^{-1} X_1^T$, X_1 are columns of $(X, 0)(L, L^c)^{-1}$ that correspond to $\boldsymbol{\beta}$ at the first location for each slope and the intercept. Thus, we directly apply the LARS algorithm to solve for $\boldsymbol{\theta}$, obtain the complete solution path and do the back transformation to get $\hat{\boldsymbol{\beta}}$, including adding back the unpenalized projection as described in section 4.1.3.

It remains to choose λ in 5.1. We use the function in the lars package (Hastie and Efron, 2013) to do cross validation to get the optimal fitting. In this simulation study, at each location, the observations are divided into 5 subsets and one of the subset is removed from fitting and serves as a validation sample each time. The λ value which provides the smallest cv error is selected as the optimal tuning parameter.

To test the performance of our algorithm, we consider the following six true $\boldsymbol{\beta}$ structures:

1. All locations share the same intercept and slopes, respectively. We refer this structure as overall $\boldsymbol{\beta}'$ s.
2. All locations share the same intercept, but slopes are smoothed and are generated

from (5.3).

3. All locations share the same slopes, but the intercept are different and correlated according to distances.
4. All intercepts and slopes at each location are smoothed according to (5.3).
5. All intercepts and slopes at each location are different, but they are not correlated: $\beta \sim Uniform(-3, 3)$. We refer this as "different".
6. The grid is divided into blocks and coefficients within each block share the same value. β values are generated from (5.3) according to adjacency of blocks. The intercept and four slopes are divided into one, two, four, nine and sixteen blocks, respectively.

In addition, we also test four different methods:

1. Fit the spatially varying coefficients regression using the generalized lasso model.
 2. Fit the spatially varying coefficients regression using the generalized ridge model.
- In the generalized ridge, we set constraint on l_2 norm of coefficients differences in their neighborhood, which is given by:

$$\min_{\beta} \frac{1}{2} \|\mathbf{y} - X\beta\|_2^2 + \lambda \|D\beta\|_2^2. \quad (5.4)$$

The solution to generalized ridge is: $\hat{\beta} = (X^T X + \lambda D^T D)^{-1} X^T \mathbf{y}$ and the degrees of freedom at each λ value can be computed as: $df(\lambda) = tr(X(X^T X + \lambda D^T D)^{-1} X^T)$.

We tune λ value via 5-fold cross validation as well.

3. Fit one OLS regression for all locations.

4. Fit OLS regression for each location separately.

All methods are evaluated by the mean square prediction error (MSPE) and each method was run 100 times. We report the median of MSPE and their bootstrap standard errors in Table 5.1, based on 100 simulated datasets.

Table 5.1: The table shows median MSPE and standard error of four different approaches for different structures for 100 simulated datasets.

Structures	lasso CV	ridge CV	Overall for All Locations	Separate for Each Location
All Same	0.0177 (0.0024)	0.0171 (0.0017)	0.0125 (0.0019)	0.9767 (0.0183)
Same Intercept Smooth Slopes	0.4296 (0.0145)	0.4103 (0.0068)	2.8598 (0.0569)	1.0134 (0.0137)
Smooth Intercept Same Slopes	0.2099 (0.0059)	0.2491 (0.0058)	0.6038 (0.0026)	1.0006 (0.0186)
All Smooth	0.5086 (0.0149)	0.4482 (0.0083)	2.7665 (0.0312)	0.9926 (0.0164)
Different	0.8449 (0.0221)	0.8454 (0.0171)	14.6612 (0.2349)	0.9948 (0.0182)
Blocks	0.2392 (0.0111)	0.4626 (0.0127)	7.3653 (0.0743)	0.9901 (0.0159)

It can be seen from the table that generalized lasso and generalized ridge perform well in the simulation study. The two methods generate smaller MSPE and standard error compared to fitting separate models at each location for all of the six β scenarios. Fitting one overall OLS performs the best when the intercept and slopes do not differ by locations, however, it gives high prediction errors when we have uncorrelated coefficients or when the coefficients are varied across blocks. For the first five structures, generalized ridge produces smaller MSPE and standard error compared to generalized lasso. This is because spatially varying coefficients β are generated from (5.3), thus are smoothed but not exactly the same as their neighbors. When the true β are varied by blocks,

generalized lasso improves on generalized ridge.

5.2 Comparison to the genlasso package

We also compare our algorithm to the dual algorithm introduced by Tibshirani and Taylor (2011) in terms of both solution path and computational efficiency. Arnold and Tibshirani (2016) implemented the dual algorithm and their implementation is available in R package **genlasso**. We use the `genlasso` function in the package to do the comparison. As an example, we use the 4th coefficient structure in section 5.1: spatially varying β 's are smoothed for both intercept and 4 slopes according to (5.3), and each location has 5 observations. The figures show runtime of the two methods when increasing the number of grid cells, and MSE along the solution path for both algorithms.

Figure 5.1: Runtime of both methods in seconds with increasing number of locations on the 2D grid.

Figure 5.2: MSE of both methods with increasing steps in the solution path. The dashed blue line indicate the optimal step chosen by cross validation.

From the figure, we can see that when grid size is small, the two methods perform similarly in terms of computational efficiency. When the number of locations increases, LARS approach becomes faster.

We also compare the solution path of the two methods. Again, when the number of parameters is not large, LARS and genlasso provide exactly the same solution paths. When p becomes larger, early steps in the solution paths are the same but begin to distinguish slightly at a certain step and can be different at the end, probably due to numerical errors. As we have the same number of parameters and observations per location, a perfect fitting should be achieved when $\lambda = 0$. We check the MSE of both methods along the path in Figure 5.2 and find that MSE decreases and reaches its minimum in our approach, but decreases first and then increases or becomes jumpy

in genlasso. It appears that there may be some numerical instability resulting in this unwanted behavior. In this case, λ chosen by cross validation occurs before this major issue, but that will not always be the case.

5.3 Real Data Analysis

The two-dimensional spatially varying coefficients model has many applications in a variety of fields. For example, it can be used to analyze neuroimaging (Zhu et al., 2014), to predict real estate (Hallac et al., 2015), to estimate crime rate (Wheeler and Waller, 2009), to evaluate air quality (Hamm et al., 2015) and many others.

In this section, we apply the spatially varying coefficients model to analyze data collected by the labs of Rob Dunn (NCSU) and Noah Fierer (CU) (Grantham et al., 2015, Barberán et al., 2015), which provides dust samples from 1133 homes across the US. It contains the measures of more than 100,000 bacteria and fungi species in and outside of the homes and records home features such as the geographic variables, number of bedrooms, home type and so on. In this study, we analyze a subset of the real data containing 763 fungi species measured by abundance of Operational Taxonomic Units (OTUs) and model the richness of fungi based on home location and the number of bedrooms. The richness of fungi is obtained by counting the number (out of 763) of fungi species observed at each location.

Figure 5.3 shows the richness of the 763 fungi species across the US. The distribution of the 1133 homes is on the map. More dust samples were collected in the east, especially in northeast America along the coast. Richness of fungi is relatively lower in the west and higher in central and eastern US.

Before analyzing the data, we pre-group the data using K-means clustering ($k =$

Figure 5.3: Distribution of richness of fungi across the US.

85) by home longitude and latitude. Based on the centroids of the clusters, a voronoi tessellation is used to determine the neighborhood structure , which is shown in Figure 4.

The generalized lasso problem is:

$$\min_{\beta} \sum (y_{ij} - \beta_{i0} - X_{ij1}\beta_{i1})^2 + \lambda |D\beta|, \quad (5.5)$$

where $i = 1, 2, \dots, 85$ represents 85 locations, $j = 1, 2, \dots$ denotes the j^{th} home at each location. β_{i0} is the intercept at the i^{th} location, β_{i1} is the coefficient of number of bedrooms at the i^{th} location and X_{ij1} is the corresponding predictor. The predictor is standardized before doing the fitting. Similarly as in the simulation, the penalty matrix D in (5.5) is:

$$D_k = (0, \dots, 1, \dots, -1, \dots, 0), \quad (5.6)$$

Figure 5.4: Using voronoi tessellation, we determine the neighborhood structure of the pre-clustered groups.

where D_k is the k^{th} row of D , 1 and -1 correspond to coefficients of two adjacent locations that are determined by the voronoi tessellation. Again, we apply four models to solve the problem and compare their performance with each other.

1. Fit the spatially varying coefficients regression using the generalized lasso model.
2. Fit the spatially varying coefficients regression using the generalized ridge model.
3. Fit one overall OLS regression for all location groups.
4. Fit OLS for each group separately.

To evaluate the performance of the four approaches, we divide the data and randomly select 20% of the observations from each of the 85 pre-clustered groups to serve as the test sample. We repeat the procedure for 50 times and calculate the mean of MSPE and

standard error. Table 5.2 shows the mean MSPE and standard error provided by the four methods.

Table 5.2: Comparison of mean MSPE and standard error for four different models.

lasso	ridge	overall	separate
0.1065 (0.0012)	0.1037 (0.001)	0.1143 (0.001)	0.1335 (0.0017)

From Table 5.2, we can see that fitting an OLS for each group separately performs worst in terms of both mean squared prediction error and standard error. Fitting an overall OLS for all locations gives low standard error but ends up with relatively high prediction error. Both the generalized lasso and the generalized ridge provide small MSPEs and standard errors compared to the other two. This indicates that the underlying structure of the data prefers a model with smoothed grid or a clustering structure.

In Figure 5.5, we can observe the fitted intercepts at different locations in the US. It can be seen that intercepts are lower in the west and higher in the middle and east part of US, which is consistent with the richness of OTUs across the whole country. Figure 5.6 shows the fitted coefficients of bedroom numbers and it is noteworthy that the differences among slopes are relatively small across the 85 locations. Thus we also fit an intercept-varying only model, in which only intercepts are spatially varying but the slope is a constant. It is then compared to the model where both the intercept and slope are different across the 85 locations. Note that it may also be desired to fit a model where some of the slopes are spatially varying while the others are constant. This partially-varying approach can be easily achieved by projecting out columns in the design matrix that are corresponding to the non-varying coefficients.

Figure 5.5: Plot of the intercept when both the intercept and slope are spatially varying.

Figure 5.6: Plot of the coefficient of number of bedrooms when both the intercept and slope are spatially varying.

Table 5.3: Comparison of mean MSPE between four different methods with or without spatially varying the slope.

Model	lasso	ridge	overall	separate
Intercept And Slope Are Spatially Varying	0.1065 (0.0012)	0.1037 (0.001)	0.1143 (0.001)	0.1335 (0.0017)
Only Intercept Is Spatially Varying	0.1063 (0.0012)	0.1026 (9e-04)	0.1143 (0.001)	0.1092 (0.001)

Table 5.3 gives the comparison of mean squared prediction error between these two approaches. When the slope is not spatially varying, fitting an OLS separately for each location performs a lot better than the model with more freedom. Varying the coefficients of room number across different locations does not describe the true structure. The generalized lasso and generalized ridge provide similar MSPEs and standard errors when only intercepts are spatially different. Figure 5.7 shows how the intercepts are different in the US and it is similar compared to figure 5.5. We can make the conclusion that when only some of the slopes are truly spatially varying in a problem, the generalized lasso and generalized ridge approaches can still capture the true structure other than trying to making all coefficients different.

As is discussed in section 4.2, the 95% confidence intervals of the coefficients can be calculated using the modified bootstrap method. We note that choosing a_n can be viewed as a model selection problem, as a_n defines a threshold that sets estimates to 0, thus changing the model, but only in discrete steps. Thus we use AIC to choose a_n .

Figure 5.8 shows the confidence intervals of the spatially varying intercepts when a_n is tuned to be 0.00878. The overall OLS estimate which represents a same intercept across different locations is indicated by the green reference line. From this confidence interval plot, we can see a clear partition of intercepts by the generalized lasso estimators.

Figure 5.7: Intercept for different locations when the slope is fixed.

Some of the lasso estimators are larger compared to the rest. While the overall OLS fits a value in between the two partitions. For many locations, it either underestimates or overestimates the intercept. And it is clear that many of the confidence intervals do not cover the overall estimate.

More information can be observed in figure 5.9. Some of the confidence intervals for the generalized lasso in southeastern US cover the overall least square estimate. While richness of OTU is higher in the northeast, the overall least square underestimates their intercepts. In contract, the overall OLS overestimates the intercepts in the west.

Not only can we do estimation and prediction using the generalized lasso approach, we can also perform clustering based on the grouped β' s values. When locations share the same intercept value in the model, they are clustered into the same group. Figure 5.10 shows the partitions of 85 locations into 9 distinct clusters after the data is fitted

Figure 5.8: The overall OLS estimator, generalized lasso estimators and confidence intervals for the intercept.

Figure 5.9: Intercept estimated by the generalized lasso and whether the overall OLS is covered by the confidence interval at each location.

by the generalized lasso. Most of the west US locations are in the same cluster while the south US including south California, south Arizona, Texas and Louisiana share the same intercept. The eastern coast are in two separate groups divided by the boundary of Virginia and North Carolina. Locations in central north area are clustered together. While Kansas, Missouri, Arkansas, Mississippi and Oklahoma form a group. There are several single-element clusters: one in Florida, one in east Tennessee and one on the boundary of Minnesota and North Dakota.

The 95% confidence intervals of differences between neighbored locations can also be achieved to determine whether the differences are significant. Figure 5.11 shows the clustering structure by significant differences between neighbors. The previous nine clusters collapse to four groups. The west and south US do not have significantly different intercepts according to the confidence intervals. And the whole northeast area falls into the same group. None of the previous three single element locations is still significantly different from the rest.

Figure 5.10: Clustering of locations by the generalized lasso.

Figure 5.11: Clustering of locations by the confidence intervals of the generalized lasso.

Chapter 6

Discussion and Conclusion

In this thesis, we discuss the generalized lasso problem and propose a new algorithm to solve it which allows the use of existing lasso solvers and build confidence intervals for the generalized lasso estimators.

We show our motivation to study the generalized lasso by discussing its various applications when the penalty matrix D takes different forms. Then we discuss some of the current studies of the generalized lasso. To solve this problem, standard quadratic programming methods can be used by introducing new variables and decomposing the parameters into positive and negative parts, but are usually inefficient and do not give solution paths. Tibshirani and Taylor (2011) proposed an algorithm to solve the generalized lasso problem by solving its dual. Gaines and Zhou (2016) further implemented the method using the QR decomposition and provided fast implementation for special cases. The methods provides full solution path to the generalized lasso problem. Zhu (2015) used Alternating Direction Method of Multipliers (ADMM) and augmented ADMM to solve the generalized lasso problem. However, ADMM approach does not provide the complete solution path. One needs to arbitrarily provide a sequence of λ values. More

recently, Gaines and Zhou (2016) showed that any generalized lasso can be rewritten as a constrained lasso problem. They proposed different algorithms to solve the problem, including quadratic programming, ADMM and a path following algorithm.

The key idea in our study is to reduce the generalized lasso to a regular lasso problem. We can benefit from this approach by taking advantage of existing lasso solvers to get the full solution path, and in such a way, we can also apply the inferences methods proposed for the lasso problem to build confidence intervals or perform tests for the generalized lasso estimators. However, when considering the connection between the generalized lasso and the regular lasso, current approaches (Tibshirani and Taylor, 2011, Duan et al., 2016) only apply to problems where $D_{m \times p}$ is a full row rank matrix. And usually in practice, such transformation methods do not apply due to the structure of D . In our proposed algorithm, we transform the generalized lasso to a regular lasso problem given any penalty matrix $D_{m \times p}$. In the simulation study, we see that the generalized lasso and generalized ridge outperform in spatially varying coefficients regression. And when comparing to the dual algorithm proposed by Tibshirani and Taylor (2011), our algorithm performs both accurately and efficiently.

We list several existing approaches to perform hypothesis tests or construct confidence intervals for lasso estimators. By building a connection between the generalized lasso and the regular lasso, we can adapt some of the lasso inference approaches for use in constructing confidence intervals for the generalized lasso parameters. To achieve this, the lasso method needs to provide confidence intervals for $\hat{\beta}^{Lasso}$ jointly, in such a way, we are able to perform inferences for the generalized lasso estimators via linear back transformation. In this study, we modify the “modified residual bootstrap lasso” approach to build confidence intervals for the generalized lasso. Our approach provides confidence intervals for both zero and non-zero estimates. We can obtain inferences for

parameters associated with the structure of the problem, for example, the confidence intervals of pairwise differences between coefficients. If the estimated difference is non-zero, but the confidence interval contains zero, we can claim this difference is not statistically significant. In addition, we can also get the inferences of the original parameters in the generalized lasso.

References

- Arnold, T. B. and Tibshirani, R. J. (2016) Efficient implementations of the generalized lasso dual path algorithm. *Journal of Computational and Graphical Statistics*, **25**, 1–27.
- Bae, K. and Mallick, B. K. (2004) Gene selection using a two-level hierarchical bayesian model. *Bioinformatics*, **20**, 3423–3430.
- Barberán, A., Dunn, R. R., Reich, B. J., Pacifici, K., Laber, E. B., Menninger, H. L., Morton, J. M., Henley, J. B., Leff, J. W., Miller, S. L. et al. (2015) The ecology of microscopic life in household dust. In *Proc. R. Soc. B*, vol. 282, 20151139. The Royal Society.
- Boyd, S., Parikh, N., Chu, E., Peleato, B. and Eckstein, J. (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, **3**, 1–122.
- Bühlmann, P. et al. (2013) Statistical significance in high-dimensional linear models. *Bernoulli*, **19**, 1212–1242.
- Camponovo, L. (2015) On the validity of the pairs bootstrap for lasso estimators. *Biometrika*, **102**, 981–987.
- Chatterjee, A. and Lahiri, S. (2010) Asymptotic properties of the residual bootstrap for lasso estimators. *Proceedings of the American Mathematical Society*, **138**, 4497–4509.
- Chatterjee, A. and Lahiri, S. N. (2011) Bootstrapping lasso estimators. *Journal of the American Statistical Association*, **106**, 608–625.
- Chen, S. S., Donoho, D. L. and Saunders, M. A. (2001) Atomic decomposition by basis pursuit. *SIAM review*, **43**, 129–159.
- Devlin, B., Roeder, K. and Wasserman, L. (2003) Analysis of multilocus models of association. *Genetic epidemiology*, **25**, 36–47.
- Duan, J., Soussen, C., Brie, D., Idier, J., Wan, M. and Wang, Y.-P. (2016) Generalized lasso with under-determined regularization matrices. *Signal Processing*, **127**, 239–246.
- Eckstein, J. and Bertsekas, D. P. (1992) On the douglas-rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, **55**, 293–318.

- Efron, B., Hastie, T., Johnstone, I., Tibshirani, R. et al. (2004) Least angle regression. *The Annals of statistics*, **32**, 407–499.
- Feuk, L., Carson, A. R. and Scherer, S. W. (2006) Structural variation in the human genome. *Nature Reviews Genetics*, **7**, 85–97.
- Figueiredo, M. A. (2003) Adaptive sparseness for supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, **25**, 1150–1159.
- Friedman, J., Hastie, T., Höfling, H., Tibshirani, R. et al. (2007) Pathwise coordinate optimization. *The Annals of Applied Statistics*, **1**, 302–332.
- Friedman, J., Hastie, T. and Tibshirani, R. (2010) Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, **33**, 1.
- Gaines, B. R. and Zhou, H. (2016) Algorithms for fitting the constrained lasso. *arXiv preprint arXiv:1611.01511*.
- Van de Geer, S., Bühlmann, P., Ritov, Y., Dezeure, R. et al. (2014) On asymptotically optimal confidence regions and tests for high-dimensional models. *The Annals of Statistics*, **42**, 1166–1202.
- Gelfand, A. E., Kim, H.-J., Sirmans, C. and Banerjee, S. (2003) Spatial modeling with spatially varying coefficient processes. *Journal of the American Statistical Association*, **98**, 387–396.
- Gill, P. E., Murray, W. and Saunders, M. A. (1997) User’s guide for sqopt 5.3: A fortran package for large-scale linear and quadratic programming. *Tech. rep.*, Tech. Rep. NA 97-4, Department of Mathematics, University of California, San Diego.
- Grantham, N. S., Reich, B. J., Pacifici, K., Laber, E. B., Menninger, H. L., Henley, J. B., Barberán, A., Leff, J. W., Fierer, N. and Dunn, R. R. (2015) Fungi identify the geographic origin of dust samples. *PLoS One*, **10**, e0122605.
- Greenland, S. and Longnecker, M. P. (1992) Methods for trend estimation from summarized dose-response data, with applications to meta-analysis. *American journal of epidemiology*, **135**, 1301–1309.
- Hallac, D., Leskovec, J. and Boyd, S. (2015) Network lasso: Clustering and optimization in large graphs. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 387–396. ACM.
- Hamm, N., Finley, A., Schaap, M. and Stein, A. (2015) A spatially varying coefficient model for mapping pm10 air quality at the european scale. *Atmospheric Environment*, **102**, 393–405.

- Hastie, T. and Efron, B. (2013) *lars: Least Angle Regression, Lasso and Forward Stage-wise*. URL <https://CRAN.R-project.org/package=lars>. R package version 1.2.
- He, Q. and Lin, D.-Y. (2010) A variable selection method for genome-wide association studies. *Bioinformatics*, **27**, 1–8.
- Hoerl, A. E. and Kennard, R. W. (1970) Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, **12**, 55–67.
- Hu, Y., Chi, E. and Allen, G. I. (2015) Admm algorithmic regularization paths for sparse statistical machine learning. *arXiv preprint arXiv:1504.06637*.
- James, G. M., Paulson, C. and Rusmevichientong, P. (2013) Penalized and constrained regression. *Tech. rep.*, Technical report, 2013. 15.
- Javanmard, A. and Montanari, A. (2014) Confidence intervals and hypothesis testing for high-dimensional regression. *The Journal of Machine Learning Research*, **15**, 2869–2909.
- Kim, S.-J., Koh, K., Boyd, S. and Gorinevsky, D. (2009) ℓ_1 trend filtering. *SIAM review*, **51**, 339–360.
- Knight, K. and Fu, W. (2000) Asymptotics for lasso-type estimators. *Annals of statistics*, 1356–1378.
- Lee, J. D., Sun, D. L., Sun, Y. and Taylor, J. E. (2013) Exact post-selection inference with the lasso. *arXiv preprint arXiv:1311.6238*.
- Lee, J. D., Sun, D. L., Sun, Y., Taylor, J. E. et al. (2016) Exact post-selection inference, with application to the lasso. *The Annals of Statistics*, **44**, 907–927.
- Leser, C. (1961) A simple method of trend construction. *Journal of the Royal Statistical Society. Series B (Methodological)*, 91–107.
- Levitt, S. D. (2004) Understanding why crime fell in the 1990s: Four factors that explain the decline and six that do not. *The Journal of Economic Perspectives*, **18**, 163–190.
- Liu, J., Wang, K., Ma, S. and Huang, J. (2011) Regularized regression method for genome-wide association studies. In *BMC proceedings*, vol. 5, 1. BioMed Central.
- Lockhart, R., Taylor, J., Tibshirani, R. J. and Tibshirani, R. (2014) A significance test for the lasso. *Annals of statistics*, **42**, 413.
- Lu, S., Liu, Y., Yin, L. and Zhang, K. (2017) Confidence intervals and regions for the lasso by using stochastic variational inequality techniques in optimization. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **79**, 589–611.

- Malioutov, D. M., Cetin, M. and Willsky, A. S. (2005) Homotopy continuation for sparse signal representation. In *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, vol. 5, v–733. IEEE.
- Minnier, J., Tian, L. and Cai, T. (2011) A perturbation method for inference on regularized regression estimates. *Journal of the American Statistical Association*, **106**, 1371–1382.
- Nowak, G., Hastie, T., Pollack, J. R. and Tibshirani, R. (2011) A fused lasso latent feature model for analyzing multi-sample acgh data. *Biostatistics*, kxr012.
- Osborn, D. R. (1995) Moving average detrending and the analysis of business cycles? *Oxford Bulletin of Economics and Statistics*, **57**, 547–558.
- Osborne, M. R., Presnell, B. and Turlach, B. A. (2000) A new approach to variable selection in least squares problems. *IMA journal of numerical analysis*, **20**, 389–403.
- Park, T. and Casella, G. (2008) The bayesian lasso. *Journal of the American Statistical Association*, **103**, 681–686.
- Reich, D. E., Cargill, M., Bolk, S., Ireland, J., Sabeti, P. C., Richter, D. J., Lavery, T., Kouyoumjian, R., Farhadian, S. F., Ward, R. et al. (2001) Linkage disequilibrium in the human genome. *Nature*, **411**, 199–204.
- She, Y. (2008) *Sparse regression with exact clustering*. ProQuest.
- Stein, C. M. (1981) Estimation of the mean of a multivariate normal distribution. *The annals of Statistics*, 1135–1151.
- Sun, T. and Zhang, C.-H. (2010) Comments on: ? 1-penalization for mixture regression models. *Test*, **19**, 270–275.
- Tibshirani, R. (1996) Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 267–288.
- Tibshirani, R., Saunders, M., Rosset, S., Zhu, J. and Knight, K. (2005) Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **67**, 91–108.
- Tibshirani, R. and Wang, P. (2008) Spatial smoothing and hot spot detection for cgh data using the fused lasso. *Biostatistics*, **9**, 18–29.
- Tibshirani, R. J. and Taylor, J. (2011) The solution path of the generalized lasso. *Ann. Statist.*, **39**, 1335–1371. URL<http://dx.doi.org/10.1214/11-AOS878>.

- Tsay, R. S. (2005) *Analysis of financial time series*, vol. 543. John Wiley & Sons.
- Wang, Y.-X., Sharpnack, J., Smola, A. J. and Tibshirani, R. J. (2015) Trend filtering on graphs. In *AISTATS*.
- Wheeler, D. C. and Waller, L. A. (2009) Comparing spatially varying coefficient models: a case study examining violent crime rates and their relationships to alcohol outlets and illegal drug arrests. *Journal of Geographical Systems*, **11**, 1–22.
- Xin, B., Kawahara, Y., Wang, Y. and Gao, W. (2014) Efficient generalized fused lasso and its application to the diagnosis of alzheimer’s disease. In *AAAI*, 2163–2169. Citeseer.
- Yang, T., Liu, J., Gong, P., Zhang, R., Shen, X. and Ye, J. (2016) Absolute fused lasso and its application to genome-wide association studies. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1955–1964. ACM.
- Yuan, M. and Lin, Y. (2005) Efficient empirical bayes variable selection and estimation in linear models. *Journal of the American Statistical Association*, **100**, 1215–1225.
- Zhang, C.-H. and Zhang, S. S. (2014) Confidence intervals for low dimensional parameters in high dimensional linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **76**, 217–242.
- Zhu, H., Fan, J. and Kong, L. (2014) Spatially varying coefficient model for neuroimaging data with jump discontinuities. *Journal of the American Statistical Association*, **109**, 1084–1098.
- Zhu, Y. (2015) An augmented admm algorithm with application to the generalized lasso problem. *Journal of Computational and Graphical Statistics*.
- Zhu, Y., Shen, X. and Pan, W. (2013) Simultaneous grouping pursuit and feature selection over an undirected graph. *Journal of the American Statistical Association*, **108**, 713–725.
- Zou, H. (2006) The adaptive lasso and its oracle properties. *Journal of the American statistical association*, **101**, 1418–1429.