Trend-Filtered Projection for Principal Component Analysis

Dissertation

Presented in Partial Fulfillment of the Requirements for the Degree
Doctor of Philosophy in the Graduate School of The Ohio State
University

By

Liubo Li, B.S., M.S.

Graduate Program in Statistics

The Ohio State University

2017

Dissertation Committee:

Vincent Q. Vu, Advisor
Yoonkyung Lee, Co-advisor
Sebastian A. Kurtek
Abstract

Principal component analysis is one of the most widely used dimension reduction techniques. We propose an approach for performing smoothed PCA of data that is observed over a dense and equally spaced grid. The proposed approach combines ideas from recent developments in convex relaxation of PCA and $l_1$ Trend Filtering of time series. Our method produces smooth estimates of the projection matrix of the principal subspace that are locally adaptive, and is based on a convex optimization problem that is solved by an augmented alternating direction method of multipliers (augADMM) algorithm. We describe the method and the algorithm in detail and compare the proposed method with existing methods by a numerical study. The effect of the choice of a penalty on the estimates given by the proposed method is also illustrated in a numerical study. Moreover, we present applications of the proposed method to real data and demonstrate its effectiveness.
This dissertation is dedicated to my dear parents, my advisor, my co-advisor and all my dear friends.
Acknowledgments

I would like to give my deep thanks and sincere gratitude to everyone for helping me to complete this dissertation.

I would like to thank my advisor, Dr. Vu, for his patience, motivation, and immense knowledge. He provided me with personal and academic guidance and taught me a great deal about both research and life in general. I would like to thank my co-advisor, Dr. Lee, who guided my research for the past several years and patiently corrected my writing when I worked on this dissertation.

Finally, I would also like to thank my parents. They were always encouraging me with their supports.

This research is supported in part by the National Science Foundation under grants DMS-15-13566 and DMS-15-13621.
Vita

2011 ............................................ B.S. Statistics
Beijing Normal University, China

2011-present ................................. Graduate Teaching Associate,
The Ohio State University
Columbus, OH

Fields of Study

Major Field: Statistics
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>ii</td>
</tr>
<tr>
<td>Dedication</td>
<td>iii</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>iv</td>
</tr>
<tr>
<td>Vita</td>
<td>v</td>
</tr>
<tr>
<td>List of Figures</td>
<td>viii</td>
</tr>
<tr>
<td>1. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Literature Review</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Two Motivating Examples</td>
<td>3</td>
</tr>
<tr>
<td>1.2.1 Local Adaptiveness</td>
<td>3</td>
</tr>
<tr>
<td>1.2.2 Small Eigengap</td>
<td>5</td>
</tr>
<tr>
<td>1.3 Overview</td>
<td>7</td>
</tr>
<tr>
<td>2. Trend-Filtered Projection for PCA</td>
<td>8</td>
</tr>
<tr>
<td>2.1 Review of the Existing Methods</td>
<td>8</td>
</tr>
<tr>
<td>2.2 Motivations</td>
<td>11</td>
</tr>
<tr>
<td>2.2.1 Convex Relaxation of PCA</td>
<td>11</td>
</tr>
<tr>
<td>2.2.2 $l_1$ Trend Filtering</td>
<td>13</td>
</tr>
<tr>
<td>2.3 Formulation of TFPCA</td>
<td>15</td>
</tr>
<tr>
<td>2.4 Other Choices of Roughness Penalty</td>
<td>18</td>
</tr>
<tr>
<td>2.5 Eigenvector Estimate</td>
<td>20</td>
</tr>
</tbody>
</table>
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 The estimated PC directions by the ordinary PCA (green dots) and Silverman’s method (red line). The smooth estimate in the left panel is given by a small value of roughness penalty. The smooth estimate in the right panel is given by a large value of roughness penalty.</td>
<td>4</td>
</tr>
<tr>
<td>1.2 The true projection matrix and the estimated projection matrices by TFPCA with penalty $|DX|_{1,1}$, the ordinary PCA, and PMA with fused lasso penalty. The top left panel is the true projection matrix. The top right panel is the estimates from our proposed method TF-PCA. The bottom row gives estimates by the ordinary PCA and PMA.</td>
<td>6</td>
</tr>
<tr>
<td>2.1 The example of local adaptiveness of $l_1$ Trend Filtering (Tibshirani, 2014, Chapter 2)</td>
<td>15</td>
</tr>
<tr>
<td>4.1 The examples of the alternating vectors used as the population eigenvectors in the first simulation study with different roughness.</td>
<td>35</td>
</tr>
<tr>
<td>4.2 The averaged SSEs of TFPCA with penalty $|DX|<em>{1,1}$ and $|DXD^T|</em>{1,1}$, PMA, and the ordinary PCA across 100 replicates for each combination of the simulation parameters ($k = 1$)</td>
<td>37</td>
</tr>
<tr>
<td>4.3 Comparison between TFPCA with $|DX|<em>{1,1}$ and PMA ($\frac{SSE</em>{dx}}{SSE_{pma}}, k = 1$)</td>
<td>38</td>
</tr>
<tr>
<td>4.4 Comparison between TFPCA with $|DXD^T|<em>{1,1}$ and PMA ($\frac{SSE</em>{dx}}{SSE_{pma}}, k = 1$)</td>
<td>39</td>
</tr>
<tr>
<td>4.5 The averaged SSEs of TFPCA with penalty $|DX|<em>{1,1}$ and $|DXD^T|</em>{1,1}$, PMA, and the ordinary PCA across 100 replicates for each combination of the simulation parameters ($k = 2$)</td>
<td>40</td>
</tr>
</tbody>
</table>
4.6 Comparison between TFPCA using $\|DX\|_{1,1}$ and PMA ($\frac{SSE_{dx}}{SSE_{pma}}$, $k = 2$) 41

4.7 Comparison between TFPCA using $\|DXD^T\|_{1,1}$ and PMA ($\frac{SSE_{dx}}{SSE_{pma}}$, $k = 2$) 42

4.8 The population eigenvectors used in the first setting has a small $l_1$ norm. In other words, most entries of the eigenvectors are close to zero, while a few entries are far from zero. We can see spikes in both eigenvectors. 44

4.9 The population eigenvectors used in the second setting has a large $l_1$ norm. In other words, the entries of the eigenvectors are relatively far away from zero, which produces vectors of larger $l_1$ norm 45

4.10 SSEs of the estimated projection matrix given by TFPCA with penalties $\|DX\|_{1,1}$, $\|DX\|_{2,2}$ and $\|DXD^T\|_{1,1}$ in the first setting 46

4.11 SSEs of the estimated projection matrix given by TFPCA with penalties $\|DX\|_{1,1}$, $\|DX\|_{2,2}$ and $\|DXD^T\|_{1,1}$ in the second setting 47

5.1 The pinch force data 49

5.2 The estimated mean by the $l_1$ Trend Filtering with tuning parameter selected by 5-fold cross-validation 50

5.3 Scree plot of the pinch force data by the ordinary PCA 52

5.4 Comparison between the PC directions given by TFPCA and Silverman’s method 53

5.5 The first four principal modes of variation given by TFPCA. The dotted lines correspond to the median principal component scores. The solid lines correspond to the maximum and the minimum principal component scores. 55

5.6 The Montreal summer temperature data 57

5.7 Scree plot of the Montreal temperature data by the ordinary PCA 58

5.8 The in-sample reconstruction MSE of TFPCA and PMA 59
5.9 The projected data for $k = 3, 6, \text{ and } 9$ by TFPCA . . . . . . . . . . . . 60
Chapter 1: Introduction

1.1 Literature Review

Principal Component Analysis (PCA) is one of the most widely used techniques in multivariate analysis. PCA reduces the dimensionality of data by finding a small number of linear combinations of the variables that preserve as much variability in the data as possible. In many applications, data are observed in the form of functions or curves. The ordinary PCA has been extended to functional data to describe the variability of the underlying continuous process, which is generally termed functional principal component analysis (FPCA). Since Rao (1958) attempted to apply the ordinary PCA to functional data without any modification, various methods for FPCA have been proposed, and FPCA has become a prevalent tool for functional data analysis. Analogous to the ordinary PCA, FPCA decomposes the covariance function into a sequence of eigenfunctions and focuses on the leading eigenfunctions for dimension reduction. This decomposition allows representation of infinite dimensional data by finite dimensional principal component scores. The leading eigenfunctions and associated principal component scores can be used for data visualization and further for modeling combined with such procedures as functional principal component regression and functional clustering.
How to estimate the leading eigenfunctions relies heavily on how the data were sampled. In biomedical studies involving longitudinal data, observations are typically sampled from sparse and irregular time points which may vary among the subjects. Several methods for FPCA in this regime have been developed. For instance, James et al. (2000) and James and Sugar (2003) proposed a method for FPCA based on mixed effect models. Yao et al. (2005) proposed a local linear smoother for estimation of the kernel function of the covariance operator and estimated the principal component scores through conditional expectation.

When data are recordings from instruments, such as fMRI devices, the sampled time points are usually dense and equally spaced. In this case, smoothing techniques are often incorporated with the ordinary PCA to capture the “intrinsic functional structure” (Huang et al., 2008). We call this type of methods smoothed PCA (SPCA). In this dissertation, we will focus on this regime. A variety of methods have been developed for data sampled from a common equally spaced grid. Rao (1958) applied ordinary PCA on the observed data and smoothed the estimated PC directions with a suitable smoother. This approach is easy to carry out but does not explore the functional structure of the observed curves. Ramsay (2006) reviewed the framework of smoothing data first and then estimating the PC directions and pointed out this approach relies heavily on the smoothing techniques used. An alternative approach is to incorporate smoothing techniques into the estimation step. Rice and Silverman (1991) and Silverman (1996) proposed methods based on the maximization of the regularized Rayleigh quotient using a roughness penalty of the quadratic form which favors smooth estimates of population eigenvectors. Huang et al. (2008) proposed a new penalized low-rank matrix approximation method. This method is based on
an optimization problem of finding the best rank-one approximation of data matrix with quadratic roughness penalties favoring smooth estimates of the singular vectors. Using the same framework, Witten et al. (2009) introduced a penalized matrix approximation method (PMA) with the fused lasso penalty.

1.2 Two Motivating Examples

In practice, the existing methods for smoothed PCA have some limitations. When the population eigenvectors have variable roughness, the methods proposed by Rice and Silverman (1991), Silverman (1996) and Huang et al. (2008) with quadratic penalties are not able to provide estimates adapting to the different levels of smoothness simultaneously. Additionally, if the gaps between the population eigenvalues are small or the population eigenvalues are identical, estimating $k$ leading eigenvectors sequentially is not as reliable as estimating the projection matrix of the subspace spanned by the $k$ leading eigenvectors. We call such a subspace the principal subspace. In the following discussion, we will use two examples to illustrate the limitations.

1.2.1 Local Adaptiveness

In the study of nonparametric regression, $l_1$ Trend Filtering estimates can adapt to the different levels of roughness better than smoothing splines estimates (Tibshirani, 2014). This finding reveals that there are some deficiencies in the quadratic penalty when estimating those functions with different levels of roughness. In the field of smoothed PCA, we expect similar weakness in the methods using quadratic penalty. For illustration, we apply the method proposed by Silverman (1996) to pinch force data. The pinch force data has 20 curves which measure the force of squeezing a
force meter with thumb and forefinger by a single subject during a short period. The
details of the data will be given in Chapter 5.

![Figure 1.1](image_url)

Figure 1.1: The estimated PC directions by the ordinary PCA (green dots) and Silverman’s method (red line). The smooth estimate in the left panel is given by a small value of roughness penalty. The smooth estimate in the right panel is given by a large value of roughness penalty.

Figure 1.1 shows the estimated first PC direction by the ordinary PCA (green
dots) and Silverman’s method (red line). The estimated leading PC direction by the
ordinary PCA suggests that the first population eigenvector may have different levels
of roughness. The estimated PC direction by the ordinary PCA has a huge peak
and valley in the left region while it is flat in the right region. The red curve in the
left panel of Figure 1.1 is the estimate given by Silverman’s method with a small
roughness penalty. The estimated PC direction fits the spike and the valley in the
left region quite well but becomes very jagged in the right area. The red curve in
the right panel is the estimate by the same method with a large roughness penalty. We could see that the estimate is smooth in the right region, but it oversmooths the signal in the left region.

1.2.2 Small Eigengap

It is unreliable to estimate \( k \) multiple population eigenvectors individually when the eigengaps are small or the population eigenvalues are identical (Vu et al., 2013). In such situations, it makes more sense to estimate the matrix of the subspace spanned by the \( k \) leading eigenvectors than to estimate individual PC directions and a convex relaxation framework of PCA can be used to estimate the projection matrix. We use a simulated example to illustrate this point. The simulated data are sampled from the multivariate normal distribution with zero mean and covariance \( \Sigma = \delta \Pi + I_p \), where \( \Pi \) is a rank-one projection matrix and \( I_p \) is a \( p \times p \) identity matrix. The eigenvector of \( \Pi \) is the realization of a step function over an equally spaced grid and \( \Pi \) has a block structure. We let \( \delta = 0.1 \), so that the eigenvalues of the population covariance matrix satisfy \( \lambda_1 - \lambda_2 = 0.1 \) and \( \lambda_2 = \lambda_3 = \cdots = \lambda_p = 1. \)
Figure 1.2: The true projection matrix and the estimated projection matrices by TFPCA with penalty $\|DX\|_{1,1}$, the ordinary PCA, and PMA with fused lasso penalty. The top left panel is the true projection matrix. The top right panel is the estimates from our proposed method TFPCA. The bottom row gives estimates by the ordinary PCA and PMA.

The gap between the first and the second population eigenvalues is small, and thus it becomes challenging to estimate the leading eigenvector. The top left panel of Figure 1.2 shows the image of the true projection matrix. The bottom left panel shows the estimated projection matrix by the ordinary PCA. We can see that when the eigengap is small, the estimate of the ordinary PCA hardly recovers the block structure of the true projection. And the estimate given by PMA (Witten et al.,
2009) is undersmoothed and does not work well in this scenario.

1.3 Overview

Motivated by the two observed limitations, we propose a new method for estimating the projection matrix of the principal subspace spanned by smooth eigenvectors. The proposed method combines ideas from the developments in convex relaxation of PCA (Overton and Womersley, 1992) and a nonparametric estimation method called \( l_1 \) Trend Filtering (Kim et al., 2009). This new method is called Trend-Filtered Projection for PCA (TFPCA).

The rest of this dissertation is organized as follows. Chapter 2 reviews some existing methods for smoothed PCA. For an introduction of TFPCA, we also review the formulation for convex relaxation of PCA and discuss the construction of various roughness penalties. Chapter 3 describes the details of the derivation of an alternating direction method of multipliers (ADMM) algorithm and an augmented ADMM for solving TFPCA problem with different penalties. The algorithms for the subproblems are discussed. Chapter 4 compares TFPCA with PMA and evaluates the performance of the different TFPCA penalties through the numerical studies. Chapter 5 provides two data analyses. We demonstrate the merits and utilities of TFPCA under different circumstances. Chapter 6 concludes the dissertation with a discussion. Appendix A includes some technical details.
Chapter 2: Trend-Filtered Projection for PCA

In this chapter, we first review the methods proposed by Rice and Silverman (1991), Silverman (1996), Huang et al. (2008), and Witten et al. (2009) for smoothed PCA. Then, we review the convex relaxation of PCA and $l_1$ Trend Filtering. The formulation of Trend-Filtered Projection for PCA is introduced next with construction of various roughness penalties. At last, we propose two ways to extract the basis of the estimated principal subspace from the TFPCA output.

2.1 Review of the Existing Methods

Notation $\{x_i, i = 1, \cdots, n\}$ is a sample of $p$-dimensional observations of size $n$. $\Sigma$ is the population covariance matrix. $\Pi$ is the projection matrix of the $k$-dimensional principal subspace. The $k$ leading eigenvectors of $\Sigma$ span this subspace. The eigenvector of $\Pi$ is not necessarily the eigenvectors of $\Sigma$. This is because the projection is unique, but the choice of basis of $\Pi$ is arbitrary. $X$ is a $n \times p$ data matrix with $x_i^T$ being in the $i$th row. $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$ is the sample mean vector. $S = \frac{1}{n} \sum_{i=1}^{n} (x - \bar{x})(x - \bar{x})^T$ is the sample covariance matrix. $v_i = (v_{i1}, \cdots, v_{ip})^T \in \mathbb{R}^p$ is the $i$th eigenvector of $S$ or the $i$th right singular vector of $X$. $u_i = (u_{i1}, \cdots, u_{in})^T \in \mathbb{R}^n$ is the $i$th left singular vector of $X$. 


In the regime of the ordinary PCA, the order of $p$ variables does not matter. In the regime of smoothed PCA, we assume each sample is a realization of some continuous processes on a dense and equally spaced grid $\{t_1, \ldots, t_p\}$ and we can not switch the order of the variables. This setting is typical in time series analysis, functional data analysis, and longitudinal data analysis. The ordinary PCA seeks the linear combination of all variables $Xv$ among all unit vectors $v \in \mathbb{R}^p$ such that the variance of $Xv$ is maximized. The maximizer $\hat{v} \in \mathbb{R}^p$ is called the first principal component direction. In other words, the first PC direction is defined by

$$v_1 = \arg \max_{\|v\|_2=1} v^T S v.$$  \hfill (2.1)

Similarly, the $j$th principal component direction is defined as the unit vector $v_j \in \mathbb{R}^p$ maximizing the variance of the corresponding linear combination uncorrelated with the preceding principal components. The maximizer $\hat{v}_j$ is given by the $j$th eigenvector of $S$. To express the roughness penalties imposed on $v_j$ in smoothed PCA, let $D^{(d)}$ be the $d$th order differencing matrix. For convenience, we may use $D$ to denote a general differencing matrix without specifying the differencing order $d$. In order to ensure smoothness in the estimated principal component directions, Silverman (1996) and Rice and Silverman (1991) considered the following problems for smoothed PCA:

Rice and Silverman (1991): \[
\max_{v \in \mathbb{R}^p} \frac{\|Xv\|_2^2 - \alpha v^T \Omega v}{\|v\|_2^2}, \quad (2.2a)
\]

Silverman (1996): \[
\max_{v \in \mathbb{R}^p} \frac{\|Xv\|_2^2}{\|v\|_2^2 + \alpha v^T \Omega v}, \quad (2.2b)
\]

where $\Omega = D^{(2)^T} D^{(2)}$ and the quadratic penalty term $v^T \Omega v = \sum_{j=2}^{p-1} (v_{j-1} - 2v_j + v_{j+1})^2$ measures the roughness of vector $v$. Both (2.2a) and (2.2b) can be solved analytically by eigendecomposition. Huang et al. (2008) proposed a low-rank matrix
approximation approach to smoothed PCA, which has the form

\[
\text{Huang et al. (2008): } \min_{u \in \mathbb{R}^n, v \in \mathbb{R}^p} \|X - uv^T\|_F^2 + \alpha \|u\|_2^2 v^T \Omega v. \tag{2.3}
\]

(2.3) is solved by alternating minimization of \(u\) and \(v\) in an iterative algorithm. When \(\alpha = 0\), this algorithm is essentially the power algorithm for the ordinary PCA. Witten et al. (2009) considered the following formulation:

\[
\min_{u \in \mathbb{R}^n, v \in \mathbb{R}^p, d \in \mathbb{R}^+} \frac{1}{2} \|X - duv^T\|_F^2 \tag{2.4}
\]

subject to \(\|u\|_2 = 1, \|v\|_2 = 1, P_1(u) \leq c_1, P_2(v) \leq c_2\),

where \(P_1(u)\) and \(P_2(v)\) are convex penalties. (2.4) is motivated by singular value decomposition with additional constraints imposed on the left and right singular vectors for some special structures. For example, \(P_1(u) = \sum_{i=1}^n |u_i|\) and \(P_2(v) = \sum_{j=1}^n |v_j|\) favor sparse estimates of the left singular vector \(u\) and the right singular vector \(v\). (2.4) leads to the following formulation of the rank one penalized matrix approximation (PMA):

\[
\max_{u \in \mathbb{R}^n, v \in \mathbb{R}^p} u^T X v \tag{2.5}
\]

subject to \(\|u\|_2 \leq 1, \|v\|_2 \leq 1, P_1(u) \leq c_1, P_2(v) \leq c_2\).

The estimates of multiple factors can be obtained sequentially. The objective function in (2.5) is biconvex in \(u\) and \(v\), which means that the objective function is convex in \(v\) with \(u\) being fixed and vice versa. Choosing \(P_2(v) = \sum_{j} |v_j - v_{j-1}|\) yields smooth estimates of \(v\). The penalty can be extended to the \(l_1\) Trend Filtering penalty of difference order \(d > 1\). The corresponding estimate of \(v\) will have a \((d - 1)\)th order piece-wise polynomial structure and will be locally adaptive. In the rest of the dissertation, we will focus on the Lagrangian form of PMA with \(P_2(v) = \sum_{j} |v_j - v_{j-1}|\),
\( P_1(u) = \sum_i |u_i| \), and \( c_1 = \sqrt{n} \):
\[
\min_{u \in \mathbb{R}^n, v \in \mathbb{R}^p} u^T X v + \lambda \sum_j |v_j - v_{j-1}|
\]
subject to \( \|u\|_2 \leq 1, \|v\|_2 \leq 1 \). \hfill (2.6)

Since \( \|u\|_2 \leq 1 \) implies that \( \|u\|_1 = \sum_i |u_i| \leq \sqrt{n} \), the constraint \( P_1(u) \leq \sqrt{n} \) is redundant. So \( P_1(u) \) is not included in the Lagrangian (2.6).

### 2.2 Motivations

In Chapter 1, we demonstrated the two limitations of the existing methods. In order to overcome the lack of local adaptiveness and deficiency in estimating individual eigenvectors when eigengaps are small, we propose a new method which combines the ideas of convex relaxation of PCA and \( l_1 \) Trend Filtering.

#### 2.2.1 Convex Relaxation of PCA

In this section, we introduce the formulation for convex relaxation of PCA. The ordinary PCA estimates the leading principal component directions by solving the following constrained problem:
\[
\max_{\|v\|_2=1} v^T S v. \hfill (2.7)
\]

For matrices of the same size \( X \) and \( Y \), define the matrix trace product of \( X \) and \( Y \) as
\[
\langle X, Y \rangle = \text{trace}(X^T Y). \hfill (2.8)
\]

Since \( v^T S v = \text{trace}(S vv^T) = \langle S, vv^T \rangle \), the optimization problem (2.7) is equivalent to
\[
\max_{X = vv^T, \|v\|_2=1} \langle S, X \rangle, \hfill (2.9)
\]
where $X$ is a rank one projection matrix. A generalization and convex relaxation of (2.9) is based on the following equalities:

$$\sum_{i=1}^{k} \lambda_i(S) = \max_{\forall V V^T = I_k} \langle VV^T, S \rangle = \max_{X \in \mathcal{F}_k} \langle X, S \rangle, \quad (2.10)$$

where $\lambda_i(S)$ is the $i$th largest eigenvalue of $S$ and $\mathcal{F}_k = \{ X | 0 \preceq X \preceq I, \text{trace}(X) = k \}$ is the convex hull of the $k$-dimensional projection matrices called the $k$-Fantope. See Overton and Womersley (1992) for the proof of the equalities in (2.10). When the $k$th and $(k+1)$th eigenvalues of $S$ are distinct, the maximizer of the convex problem $\max_{X \in \mathcal{F}_k} \langle X, S \rangle$ is the projection matrix of a $k$-dimensional subspace spanned by the $k$ leading eigenvectors of $S$. Some regularized PCA methods have been derived in this framework. Vu et al. (2013) first introduced the Fantope to regularized PCA and proposed a method for sparse PCA by estimating a sparse $k$-dimensional principal subspace. Chen and Lei (2015) introduced a method for localized functional PCA using the same framework.

There are some explanations for the merits of the convex relaxation of PCA. In applications of PCA, we project data onto the $k$ dimensional principal subspace for dimension reduction and visualization. In this case, we need to estimate the projection matrix of the principal subspace. Since $\mathcal{F}_k$ is the convex hull of $k$-dimensional projection matrices, the estimates given by the convex relaxation framework are more flexible than the rank $k$ projection matrix reconstructed by $k$ estimated PC directions. In Chapter 5, we analyze the Montreal temperature data and illustrate this advantage. Secondly, the convexity of this framework guarantees that the solution is globally optimal. Moreover, the $k$ leading eigenvalues of the population covariance matrix $\Sigma$ are not necessarily distinct. Consider an extreme case where the population covariance matrix $\Sigma$ has the form of $\delta \Pi + I$. $\Pi$ is a rank $k$ projection matrix and $I$
is an identity matrix. The \( k \) leading eigenvalues of \( \Sigma \) are \( 1 + \delta \) and the rest of the eigenvalues are 1. In this case, estimating the projection matrix by the estimated PC directions suffers from un-identifiability. In sparse PCA settings, Vu et al. (2013) have shown that the Frobenius norm of the error of the estimated projection matrix \( \hat{X} \) based on the convex relaxation framework is bounded with high probability

\[
\|\hat{X} - \Pi\|_F = \mathcal{O}\left( (\lambda_1/\lambda_k - \lambda_{k+1}) s \sqrt{\log(p/n)} \right),
\]

(2.11)

where \( \lambda_i \) is the \( i \)th eigenvalue of the population covariance \( \Sigma \), and \( s \) is a constant that satisfies \( \|\text{diag}(\Pi)\|_0 \leq s \). \( \|X\|_0 \) is the number of non-zeroes in \( X \). This result shows that we can bound the error of the estimated projection matrix of the principal subspace even when the leading eigenvectors are unidentifiable. Based on these reasons, we develop our method through the convex relaxation of PCA.

**2.2.2 \( l_1 \) Trend Filtering**

A good example of a non-parametric estimation method with local adaptivity is the \( l_1 \) Trend Filtering introduced by Kim et al. (2009). By using the \( l_1 \) norm in the roughness penalty, the \( l_1 \) Trend Filtering estimate has a piecewise polynomial structure with adaptively chosen knot points.

Suppose that \( f_0 \) is an unknown function and vector \( y \in \mathbb{R}^p \) is the observed realization of \( f_0 \) over a set of dense and equally spaced grid points \( \{t_1, \cdots, t_p\} \). The \( d \)th order \( l_1 \) Trend Filtering estimate of \( (f_0(t_1), \cdots, f_0(t_p)) \) is defined by the solution \( \hat{\beta} \) to the convex problem:

\[
\min_{\beta \in \mathbb{R}^p} \frac{1}{2}\|y - \beta\|_2^2 + \lambda \|D^{(d)}\beta\|_1,
\]

(2.12)

where \( D^{(d)} \) is the \( d \)th order difference matrix, and \( \lambda \) is the tuning parameter which controls the roughness of the solution. When \( d = 1 \), \( \|D^{(1)}\beta\|_1 = \sum_{i=1}^{p-1} |\beta_i - \beta_{i+1}| \).
(2.12) is also known as the one-dimensional total variation denoising or the fused lasso signal approximator (FLSA) (Johnson, 2013). When $d \geq 2$, $D^{(d)} \in \mathbb{R}^{(p-d) \times p}$ is defined recursively by

$$D^{(d)} = D^{(d-1)} D^{(1)},$$

where $D^{(d-1)} \in \mathbb{R}^{(p-d) \times (p-1)}$ and $D^{(1)} \in \mathbb{R}^{(p-1) \times p}$. $D^{(d)}$ can be viewed as the discrete version of the $d$th derivative operator and the penalty $\|D^{(d)} \beta\|_1$ measures the sum of the magnitude of the discrete $d$th derivative of $\beta$ over the entire domain. Because of the nature of $l_1$ norm penalty, the vector $D^{(d)} \hat{\beta}$ is typically sparse. Namely, the $d$th order Trend Filtering estimates have a piecewise $(d - 1)$th order polynomial structure with adaptively chosen knots. As discussed by Tibshirani (2014), $l_1$ Trend Filtering estimates adapt to the variable smoothness by its piecewise polynomial structure.
Figure 2.1: The example of local adaptiveness of $l_1$ Trend Filtering (Tibshirani, 2014, Chapter 2)

Figure 2.1 is an example of the comparison between $l_1$ Trend Filtering and smoothing spline for estimating a function with variable smoothness (Tibshirani, 2014). The true function is smooth in the left region and becomes wiggly in the right region. The $l_1$ Trend Filtering estimates adapt to the different levels of smoothness quite well while the smoothing spline estimates can not fit both regions well simultaneously.

2.3 Formulation of TFPCA

**Notation** In the following discussion, we work on the linear space of real symmetric matrices $\mathbb{S}_p$ equipped with the trace product defined by (2.8). Let $X = [x_1, \cdots, x_p] \in \mathbb{S}_p$, where $x_i$’s are the column vectors of $X$. The $l_1$ norm $\|\cdot\|_{1,1}$ of $X$ is
defined by
\[
\|X\|_{1,1} = \sum_{i,j} |x_{i,j}|,
\]  
(2.14)
and a mixed norm \(\|\cdot\|_{1,2}\) of \(X\) is defined by
\[
\|X\|_{1,2} = \sqrt{\sum_j \left(\sum_i |x_{i,j}|\right)^2}.
\]  
(2.15)

We provide the definition of dual norm in \(\mathbb{R}^p\).

**Definition 1** (Dual norm (Boyd and Vandenberghe, 2004)). Let \(\|\cdot\|\) be a norm in \(\mathbb{R}^p\). The associated dual norm, denoted \(\|\cdot\|_*,\) is defined as
\[
\|z\|_* = \sup_{\|x\|\leq 1} z^T x
\]
Since the dual norm of \(l_2\) norm is still \(l_2\) norm, we can write the mixed norm \(\|\cdot\|_{1,2}\) in a dual form:
\[
\|X\|_{1,2} = \max_{\sum_i c_i^2 \leq 1} \sum_{i=1}^p c_i \|x_i\|_1.
\]  
(2.16)

This dual representation will be used later in the formulation of the TFPCA penalty.

A real symmetric matrix \(X \in \mathbb{S}_p\) can be decomposed as
\[
X = V \Lambda V^T,
\]  
(2.17)
where \(V = [v_1, \cdots, v_p]\) is an orthogonal matrix whose columns are the eigenvectors of \(X\), and \(\Lambda\) is a diagonal matrix whose entries are the eigenvalues of \(X\). If we take a subset from \(\{v_1, \cdots, v_p\}\), the space spanned by the selected eigenvectors is the eigenspace of the associated eigenvalues. If \(X \in \mathbb{S}_p\) is a rank \(k\) projection matrix, the eigenvalues of \(X\) are either 1 or 0, and the eigendecomposition of \(X\) can be written as \(X = VV^T\). The choice of \(V\) is infinite because \((VO)(VO)^T = VOOTV^T = VV^T\) holds for any orthogonal matrix \(O \in \mathbb{R}^{k \times k}\). The decomposition implies that
span\{x_1, \cdots, x_p\} = \text{span}\{v_1, \cdots, v_k\}. This observation will help us construct a variant of the TFPCA penalty.

Inspired by the convex relaxation formulation of PCA in Vu et al. (2013), and Chen and Lei (2015) and l_1 Trend Filtering, we consider a convex optimization problem of the form:

\[
\begin{align*}
\max & \quad \langle S, X \rangle - \lambda \Omega(X) \\
\text{subject to} & \quad X \in \mathcal{F}^k,
\end{align*}
\]

(2.18)

where \(\Omega(X)\) is a roughness penalty which favors a smooth and locally adaptive estimate of the projection matrix of the principal subspace, and \(\lambda > 0\) is the tuning parameter that controls the level of roughness. We start with the case \(k = 1\). Let \(X\) be a rank one projection matrix with decomposition \(X = vv^T\). The \(l_1\) Trend Filtering penalty \(\|D^{(d)}v\|_1\) measures the roughness of the eigenvector \(v\). We also observe that

\[
\|D^{(d)}v\|_1^2 = \|D^{(d)}vv^TD^{(d)T}\|_{1,1} = \|D^{(d)}XD^{(d)T}\|_{1,1}.
\]

(2.19)

Accordingly, we construct the TFPCA penalty by \(\Omega(X) = \|D^{(d)}XD^{(d)T}\|_{1,1}\). Now, our TFPCA problem becomes

\[
\begin{align*}
\max & \quad \langle S, X \rangle - \lambda \|D^{(d)}XD^{(d)T}\|_{1,1} \\
\text{subject to} & \quad X \in \mathcal{F}^k.
\end{align*}
\]

(2.20)

(2.20) is formulated on the basis of variance maximization. When the tuning parameter is zero, (2.20) becomes the convex relaxation of the ordinary PCA. When \(k = 1\), Trend-Filtered PCA is similar to Rice and Silverman’s method with a different penalty. But the non-differentiable \(l_1\) norm of the penalty term brings some computational challenges. Especially for \(k \geq 2\), the mixture of the Fantope constraint and the \(l_1\) norm penalty makes the problem hard to solve.
2.4 Other Choices of Roughness Penalty

The penalty $\|DXD^T\|_{1,1}$ is based on the notion of roughness of the eigenvectors of $X$. However, solving the TFPCA problem with penalty $\|DXD^T\|_{1,1}$ leads to some difficulties. These difficulties are due to the fact that the difference matrix $D$ is ill-conditioned. The eigenvalues of $D^{(1)}D^{(1)^T} \in \mathbb{R}^{(p-1)\times(p-1)}$ are $2 - 2 \cos(\frac{i\pi}{p})$, $i = 1, \cdots, (p-1)$ (Liu et al., 2010). The condition number of $D^{(1)}D^{(1)^T}$ is $\kappa(D^{(1)}D^{(1)^T}) = \frac{2 - 2 \cos(\frac{p-1}{p} \pi)}{2 - 2 \cos(\frac{1}{p} \pi)}$. This number will increase as $p$ increases. Moreover,

$$\kappa(D^{(d)}D^{(d)^T}) = \kappa(D^{(1)}D^{(1)^T})^d$$

due to the recursive definition of $D^{(d)}$. Therefore, $D^{(d)}D^{(d)^T} \in \mathbb{R}^{(p-d)\times(p-d)}$ will be more ill-conditioned as $p$ and $d$ increase. We will provide detailed explanations on the computational difficulties in the next chapter. To simplify the computation, we consider two other roughness penalties. Suppose $X$ is a rank $k$ projection matrix that has the decomposition $X = VV^T$. The equalities $V = XV$ and $X = VV^T$ imply that $\text{span}\{x_1, \cdots, x_p\} = \text{span}\{v_1, \cdots, v_k\}$, which means the column space of $X$ and the $k$-dimensional eigenspace of $X$ coincide. So we may measure the roughness of the principal subspace indirectly by the roughness of the columns of $X$. From this perspective, we consider the penalty $\|D^{(d)}X\|_{1,1}$. The Trend-Filtered PCA problem using this new penalty $\|D^{(d)}X\|_{1,1}$ is given by

$$\max \langle S, X \rangle - \lambda \|D^{(d)}X\|_{1,1}$$

subject to $X \in \mathcal{F}^k$. 

We rewrite $\|D^{(d)}X\|_{1,1}$ as $\|D^{(d)}x_1\|_1 + \cdots + \|D^{(d)}x_p\|_1$ which is the sum of $\|D^{(d)}x_i\|_1$’s without additional constraints. The roughness of each column of $X$ contributes equally to $\|D^{(d)}X\|_{1,1}$. We may generalize the penalty $\|D^{(d)}X\|_{1,1}$ to a weighted sum
\[ \sum_{i=1}^{p} c_i \| D^{(d)} x_i \|_1. \] The weighted sum coincides with the penalty \( \| DX \|_{1,1} \) if \( c_i = 1 \) for \( i = 1, \ldots, p \). Next, we add an \( l_2 \) norm constraint on the weight vector \((c_1, \ldots, c_p)\) and take the maximum over all possible choices of the weights. We obtain a variant of the penalty \( \| DX \|_{1,1} \) of the form:

\[
\max_{\| c \|_2 \leq 1} \sum_{i=1}^{p} c_i \| D^{(d)} x_i \|_1.
\]

(2.22) reaches the maximum when \( c_i \) are proportional to \( \| D^{(d)} x_i \|_1 \). Namely, the rougher columns of \( X \) get penalized more. We notice that (2.22) has a dual form of \( \| D^{(d)} X \|_{1,2} \). We will use the dual from instead of (2.22) in our following discussion to avoid solving a minimax problem. We formulate TFPCA with this mixed norm penalty as

\[
\max \langle S, X \rangle - \lambda \| D^{(d)} X \|_{1,2}
\]

subject to \( X \in \mathcal{F}^k \).

By squaring the penalty term \( \| D^{(d)} X^T \|_{1,2} \), we obtain an equivalent form of (2.23) shown below:

\[
\max \langle S, X \rangle - \lambda \| D^{(d)} X \|_{1,2}^2
\]

subject to \( X \in \mathcal{F}^k \).

This modification helps us to derive an efficient algorithm for (2.23).

Both of the two penalties are motivated from roughness measures of the column space of the projection matrix. However, the choices of the norm in the penalty term have a different effect on the estimate. To understand this effect, we look into the structure of the two roughness penalties:

\[
\| D^{(d)} X \|_{1,1} = \| D^{(d)} x_1 \|_1 + \cdots + \| D^{(d)} x_p \|_1
\]

(2.25a)

\[
\| D^{(d)} X \|_{1,2}^2 = \| D^{(d)} x_1 \|_1^2 + \cdots + \| D^{(d)} x_p \|_1^2
\]

(2.25b)
Note that the two variants of the initial TFPCA penalty are the $l_1$ norm and $l_2$ norm of the vector $\left[\|D^{(d)}x_1\|_1, \cdots, \|D^{(d)}x_p\|_1\right]$, respectively. Like the difference between the lasso and the ridge regression, (2.21) favors a sparse structure in vector $\left[\|D^{(d)}x_1\|_1, \cdots, \|D^{(d)}x_p\|_1\right]$ and (2.24) favors $X$ with columns being similarly smooth. Because of this difference, it is expected that one penalty would outperform the other one in some specific situations. For instance, when $k = 1$, the projection matrix of the principal subspace of the leading eigenvector $v$ is $H = vv^T$. If $v$ has a localized support, namely many entries of $v$ are zero, $H$ will be sparse. Accordingly, $\left[\|D^{(d)}x_1\|_1, \cdots, \|D^{(d)}x_p\|_1\right]$ will be sparse and it is more likely that the estimate produced by (2.21) would capture the sparse structure better than the estimate produced by (2.24) in this setting. In terms of computational cost, since the $\|\cdot\|_{1,1}$ has a simpler structure than $\|\cdot\|_{1,2}$, the first variant could inherit some advantages in computation. More details on computation will be discussed in the next chapter.

2.5 Eigenvector Estimate

In some applications, such as principal component regression (PCR), we need to estimate principal component directions. We suggest the following two methods to extract the estimates of PC directions from the estimates of the projection matrix of principal subspace. Suppose that the columns of $V \in \mathbb{R}^{p \times k}$ are the $k$ leading eigenvectors of $S$ and $\Pi$ is the rank $k$ projection matrix with decomposition $\Pi = \tilde{V}\tilde{V}^T$. If $V = \tilde{V}$, then $\tilde{V}^TS\tilde{V}$ is a diagonal matrix. Motivated by this observation, we could extract $k$ leading eigenvectors $\hat{V}$ from the TFPCA output $\hat{X}$ and choose a rotation $O$ such that $(\hat{V}O)^T S (\hat{V}O)$ is diagonal. Then, we take $\hat{V}O$ as the orthonormal basis for the estimated subspace. Under the same assumption that $V$ and $\tilde{V}$ coincide, we
could have
\[
\Pi S \Pi = \tilde{V} \tilde{V}^T \Lambda \tilde{V}^T \tilde{V} = \tilde{V} \Lambda \tilde{V}^T = V S V^T \tag{2.26}
\]
So the \( k \) leading eigenvectors of \( \Pi S \Pi \) and \( S \) are the same. Accordingly, we could conduct eigendecomposition on \( \hat{X} S \hat{X} \) given the TFPCA output \( \hat{X} \) and use the \( k \) leading eigenvectors of \( \hat{X} S \hat{X} \) as the orthonormal basis for the estimated subspace. When \( \hat{X} \) is not a rank \( k \) projection matrix, the first approach discards the rest \( p - k \) eigenvectors of \( \hat{X} \), while the second approach fully uses \( \hat{X} \). When \( \hat{X} \) is the projection matrix of the subspace spanned by the leading eigenvectors of \( S \), two approaches provide the same estimates of PC directions. In practice, we recommend the second approach, since it is easy to carry out and fully uses the TFPCA output \( \hat{X} \).
Chapter 3: Computation and Algorithms

In this chapter, we begin with the standard ADMM formulation for TFPCA with various penalties. We introduce the algorithms for solving the subproblems of standard ADMM and discuss computational difficulties associated with them. Next, we present an alternative way to solve TFPCA by augmented ADMM which overcomes the aforementioned difficulties. We develop a specialized algorithm for solving the subproblem of the augmented ADMM for TFPCA with mixed norm penalty $\|DX\|_{1,2}$. We also discuss the selection of a tuning parameter for TFPCA and the computational complexity of the augmented ADMM at the end of this chapter.

3.1 ADMM Algorithm for TFPCA

The difficulty of solving the TFPCA problem is to deal with the Fantope constraint and the non-differentiable $l_1$ norm roughness penalty simultaneously. We apply the ADMM to the TFPCA problem to decouple them. The ADMM algorithm is designed to solve the problem of form

$$\min_{x,y} f(x) + g(y),$$

subject to $Ax + By = c$, \hspace{1cm} (3.1)

with the variable $x \in R^n$, and the variable $y \in R^m$, where $A \in R^{p \times n}$, $B \in R^{p \times m}$, and $c \in R^p$. We assume that the extended real valued functions, $f : R^n \to R \cup \{+\infty\}$ and
\( g : R^m \to R \cup \{+\infty\} \) are closed, proper, and convex. A function is closed, proper, and convex if and only if its epigraph

\[
\text{epi } f = \{ (x, t) \in R^n \times R | f(x) \leq t \}
\]

(3.2)
is a closed, non-empty, and convex set. Many convex optimization problems fall into this framework, including the lasso problem (Tibshirani, 1996), the generalized lasso problem (Tibshirani et al., 2005), the sparse inverse covariance selection problem (Friedman et al., 2008), etc. A comprehensive review of ADMM is given by Boyd et al. (2011). We reparameterize the TFPCA problem in the form of (3.1) and investigate how to solve the TPFCA problem with different choices of roughness penalty by the ADMM algorithm.

3.1.1 \( \| DXD^T \|_{1,1} \)

For TFPCA with penalty \( \| DXD^T \|_{1,1} \), there are two ways to construct the iterates of the standard ADMM. By introducing an auxiliary variable \( Y \in R^{(p-d) \times (p-d)} \) and the associated equality constraint \( DXD^T = Y \), we obtain an equivalent form of the TFPCA with penalty \( \| DXD^T \|_{1,1} \):

\[
\begin{align*}
\min_{X \in \mathcal{F}^k} & -\langle X, S \rangle + \lambda \| Y \|_{1,1}, \\
\text{s.t.} & DXD^T = Y.
\end{align*}
\]

(3.3)

Following the procedure of the standard ADMM, the augmented Lagrangian of the constrained problem (3.3) is given by

\[
L_\rho(X,Y,U) = -\langle X, S \rangle + \lambda \| Y \|_{1,1} + \frac{\rho}{2} \| DXD^T - Y + U \|_F^2 - \frac{\rho}{2} \| U \|_F^2 + 1\{X \in \mathcal{F}^k\},
\]

(3.4)

where \( \rho > 0 \) is the ADMM parameter and \( U \in R^{(p-d) \times (p-d)} \) is the scaled dual variable. \( 1\{X \in \mathcal{F}^k\} = 0 \) if \( X \in \mathcal{F}^k \) and \( \infty \) otherwise. The ADMM iterates are derived by
alternating the minimization of the primal variable pair \((X,Y)\) and updating the dual variable \(U\). Minimizing \(L_\rho(X,Y,U)\) over \(X\) requires solving a minimization problem:
\[
\min_{X \in \mathcal{F}^k} \langle S, X \rangle + \frac{\rho}{2} \| DXD^T - Y^{(i)} + U^{(i)} \|_F^2.
\]
Minimizing \(L_\rho(X,Y,U)\) over \(Y\) requires solving a minimization problem:
\[
\min_{Y \in \mathbb{R}^{(p-d) \times (p-d)}} \frac{\rho}{2} \| DXD^T - Y^{(i)} + U^{(i)} \|_F^2 + \frac{\lambda}{\rho} \| Y \|_{1,1}. \tag{3.6}
\]

The ADMM iterates are given by
\[
X^{(i+1)} = \arg \min_{X \in \mathcal{F}^k} -\langle S, X \rangle + \frac{\rho}{2} \| DXD^T - Y^{(i)} + U^{(i)} \|_F^2, \tag{3.7a}
\]
\[
Y^{(i+1)} = \arg \min_{Y \in \mathbb{R}^{(p-d) \times (p-d)}} \frac{1}{2} \| DX^{(i+1)}D^T - Y + U^{(i)} \|_F^2 + \frac{\lambda}{\rho} \| Y \|_{1,1}, \tag{3.7b}
\]
\[
U^{(i+1)} = DX^{(i+1)}D^T - Y^{(i+1)} + U^{(i)}. \tag{3.7c}
\]
The \(X\)-update is a least squares problem with a Fantope constraint, which is difficult to carry out due to two factors. First, the projection onto the Fantope needs a full eigendecomposition of a \(p \times p\) symmetric matrix. Using any projection-based approach, such as the projected gradient decent method, requires intensive computation in each iteration. Second, the vectorized form of the \(X\)-update is given by
\[
\text{vec}(X^{(i+1)}) = \arg \min_{X \in \mathcal{F}^k} -\text{vec}(S)^T \text{vec}(X) + \frac{\rho}{2} \| (D \otimes D)\text{vec}(X) - \text{vec}(Y^{(i)} + U^{(i)}) \|_F^2, \tag{3.8}
\]
the condition number of \(D \otimes D\) for the least square problem (3.8) is very large, and the first-order algorithms for this subproblem converge very slowly. Alternatively, the Hessian matrix of this subproblem has a large size of \(p^2 \times p^2\), and it entails issues in both computation and storage. On the other hand, the \(Y\)-update can be carried out easily by the entry-wise soft-threshold operator with parameter \(\frac{\lambda}{\rho}\):
\[
S_{\frac{\lambda}{\rho}}(x) = \begin{cases} 
0 & \text{if } |x| \leq \frac{\lambda}{\rho}, \\
 x - \frac{\lambda}{\rho} \text{sign}(x) & \text{if } |x| > \frac{\lambda}{\rho}.
\end{cases} \tag{3.9}
\]
The second version of the ADMM is constructed by introducing an auxiliary variable \( Y \in \mathbb{R}^{p \times p} \) and the associated equality constraint \( X = Y \). The corresponding equivalent form of the TFPCA problem can be written as

\[
\min_{X \in \mathcal{F}_k} -\langle X, S \rangle + \lambda \|DYD^T\|_{1,1},
\]

s.t. \( X = Y \). \( (3.10) \)

Following the standard procedure of deriving ADMM, we write out the augmented Lagrangian as

\[
L_\rho(X, Y, U) = -\langle X, S \rangle + \lambda \|DYD^T\|_{1,1} + \frac{\rho}{2} \|X - Y + U\|^2_F - \frac{\rho}{2} \|U\|^2_F,
\]

where \( \rho > 0 \) is the ADMM parameter and \( U \in \mathbb{R}^{p \times p} \) is the scaled dual variable. The ADMM iterates are

\[
X^{(i+1)} = \arg \min_{X \in \mathcal{F}_k} -\langle S, X \rangle + \frac{\rho}{2} \|X - Y^{(i)} + U^{(i)}\|^2_F, \quad (3.12a)
\]

\[
Y^{(i+1)} = \arg \min_{Y \in \mathbb{R}^{p \times p}} \frac{1}{2} \|X^{(i+1)} - Y + U^{(i)}\|^2_F + \frac{\lambda}{\rho} \|DYD^T\|_{1,1}, \quad (3.12b)
\]

\[
U^{(i+1)} = X^{(i+1)} - Y^{(i+1)} + U^{(i)}. \quad (3.12c)
\]

In this case, (3.12a) is a Fantope projection problem which can be computed exactly by a full eigendecomposition and searching for the root of a piecewise linear function (Vu et al., 2013). Then (3.12b) becomes a generalized lasso problem. Alternatively, we can update \( Y \) by solving the dual problem of (3.12b) instead. Consider the generalized lasso problem of the following form:

\[
\min_{Y \in \mathbb{R}^{p \times p}} \|S - Y\|^2_F + \lambda \|DYD^T\|_{1,1}, \quad (3.13)
\]

which can be rewritten as

\[
\min_{Y \in \mathbb{R}^{p \times p}} \|S - Y\|^2_F + \max_{\|Z\|_{\infty,\infty} \leq \lambda, Z \in \mathbb{R}^{(p-d) \times (p-d)}} \langle D^T Z D, Y \rangle, \quad (3.14)
\]
where \( Z \) is a \((p-d) \times (p-d)\) matrix, and \( d \) is the difference order. We could obtain the dual problem of (3.13) by minimizing (3.14) over variable \( Y \). The dual problem of (3.13) is given by

\[
\min_{Z \in \mathbb{R}^{(p-d) \times (p-d)}} \|S - D^T Z D\|_F^2
\]

subject to \( \|Z\|_{\infty,\infty} \leq \lambda \) \hspace{1cm} (3.15)

(3.15) is a least squares problem with box constraint. How to efficiently solve the generalized lasso problem or its dual problem (3.15) has been widely discussed. For instance, a solution path algorithm is discussed by Tibshirani and Taylor (2011) and Arnold and Tibshirani (2016). However, the problem size is \((p-d)^2\) since (3.12b) involves a \((p-d) \times (p-d)\) matrix. Thus, the computation is heavy. We could also solve the dual problem by an interior point method, but it is still difficult to apply due to the size of the problem.

### 3.1.2 \( \|DX\|_{1,1} \) and \( \|DX\|_{1,2} \)

As discussed in Chapter 2, we consider using the penalty of a simpler form to reduce the computation for solving the subproblem of the standard ADMM. We will demonstrate the idea in this section. Consider the TFPCA with penalty \( \|DX\|_{1,1} \). Similarly, there are two ways to derive the ADMM algorithms depending on the equality constraints and the associated auxiliary variables. Using an auxiliary variable \( Y \in \mathbb{R}^{(p-k) \times p} \) and the associated equality constraint \( Y = DX \), the \( X \)-update has the form of

\[
X^{(i+1)} = \arg\min_{X \in \mathbb{R}^k} -\langle S, X \rangle + \frac{\rho}{2}\|DX - Y^{(i)} + U^{(i)}\|_F^2,
\]

which is still a least squares problem with a Fantope constraint. If we use an auxiliary variable \( Y \in \mathbb{R}^{p \times p} \) and the associated equality constraint \( Y = X \), we obtain the
ADMM iterates as below:

\[ X^{(i+1)} = \arg \min_{X \in \mathcal{F}^k} -\langle S, X \rangle + \frac{\rho}{2} \| X - Y^{(i)} + U^{(i)} \|^2_F, \]  
(3.17a)

\[ Y^{(i+1)} = \arg \min_{Y \in \mathbb{R}^{p \times p}} \frac{1}{2} \| X^{(i+1)} - Y + U^{(i)} \|^2_F + \frac{\lambda}{\rho} \| DY \|_{1,1}, \]  
(3.17b)

\[ U^{(i+1)} = X^{(i+1)} - Y^{(i+1)} + U^{(i)}. \]  
(3.17c)

The \( X \)-update is carried out by the Fantope projection. The \( Y \)-update now becomes a column-wise Trend Filtering problem with an \( l_1 \) norm penalty. Some specialized algorithms solve the \( l_1 \) Trend Filtering problem very fast. When the difference order \( d = 1 \), the subproblem is associated with the method known as the fused lasso signal approximator (FLSA) and can be solved by a dynamic programming algorithm with a linear complexity (Johnson, 2013). For the general choice of the difference order \( d \), a specialized primal-dual interior point method (Kim et al., 2009) can be used. Ramdas and Tibshirani (2014) proposed a specialized ADMM for solving the \( l_1 \) Trend Filtering with difference order \( d \geq 2 \). The details of these specialized algorithms are reviewed in Appendix A.

For penalty \( \| DX \|_{1,2} \), the first formulation of ADMM leads to the same \( X \)-update. The second formulation results in the \( Y \)-update of the form as below:

\[ Y^{(i+1)} = \arg \min_{Y \in \mathbb{R}^{p \times p}} \frac{1}{2} \| X^{(i+1)} - Y + U^{(i)} \|^2_F + \frac{\lambda}{\rho} \| DY \|_{1,2}. \]  
(3.18)

This problem looks similar to the column-wise \( l_1 \) Trend Filtering. The difference is that the penalty in (3.18) uses a mixed norm. To simplify the computation, we make some modification by squaring of the penalty term. We obtain an equivalent problem of (3.18)

\[ Y^{(i+1)} = \arg \min_{Y \in \mathbb{R}^{p \times p}} \frac{1}{2} \| X^{(i+1)} - Y + U^{(i)} \|^2_F + \frac{\lambda}{\rho} \| DY \|_{1,2}^2. \]  
(3.19)
Now, the $Y$-update becomes a column-wise Trend filtering problem with squared penalty, which can be solved by solving a sequence of $l_1$ Trend Filtering problems.

### 3.2 Augmented ADMM Algorithm for TFPCA

In the previous section, we have discussed the standard ADMM algorithms for the TFPCA with various penalties. The standard ADMM solves the TFPCA problem with penalties $\|DX\|_{1,1}$ and $\|DX\|_{1,2}$ efficiently if we properly choose auxiliary variables, equality constraints and the specialized algorithms for the subproblem. For penalty $\|DXD^T\|_{1,1}$, it is difficult to solve the subproblems of the standard ADMM by both formulations. To get around this difficulty, we implement an augmented ADMM algorithm proposed by Zhu (2017). By introducing an augmented variable $(Y, \tilde{Y})$, we write the original equality constraint in the vector form $(D \otimes D)\text{vec}(X) = \text{vec}(Y)$ and associate $\tilde{Y}$ with $X$ by an equality constraint $(cI - D^TD \otimes D^TD)^{1/2}\text{vec}(X) = \text{vec}(\tilde{Y})$, where $c > 0$ is a constant such that $cI - D^TD \otimes D^TD \succeq 0$. Accordingly, we obtain an equivalent form of the TFPCA problem which is given by

$$\min_{X \in \mathcal{F}_k} -\langle X, S \rangle + \lambda \|Y\|_{1,1},$$

subject to

$$(D \otimes D)\text{vec}(X) = \text{vec}(Y)$$

$$(cI - D^TD \otimes D^TD)^{1/2}\text{vec}(X) = \text{vec}(\tilde{Y}).$$

The second equality constraint is redundant, but it helps to simplify the $X$-update step. Applying the standard ADMM procedure to (3.20), we can obtain the ADMM
updates as follow:

\[
X^{(i+1)} = \arg \min_{X \in \mathcal{F}^k} \langle S, X \rangle + \frac{\rho}{2} \| DXD^T - Y^{(i)} + U^{(i)} \|_F^2,
\]

\[
+ \frac{\rho}{2} \| (cI - DT \otimes DT) \|^2 \langle \text{vec}(X) - \text{vec}(\tilde{Y}^{(i)}) + \text{vec}(\tilde{U}^{(i)}) \|_2^2,
\]

\[
Y^{(i+1)} = \arg \min_{Y \in \mathbb{R}^{(p-d) \times (p-d)}} \frac{1}{2} \| DX^{(i+1)}D^T - Y + U^{(i)} \|_F^2 + \frac{\lambda}{\rho} \| Y \|_{1,1},
\]

\[
\text{vec}(\tilde{Y}^{(i+1)}) = (cI - DT \otimes DT)^{1/2}\text{vec}(X^{(i+1)}) + \text{vec}(\tilde{U}^{(i)}),
\]

\[
U^{(i+1)} = DX^{(i+1)}D^T - Y^{(i+1)} + U^{(i)},
\]

\[
\text{vec}(\tilde{U}^{(i+1)}) = (cI - DT \otimes DT)^{1/2}\text{vec}(X^{(i+1)}) - \text{vec}(\tilde{Y}^{(i+1)}) + \text{vec}(\tilde{U}^{(i)}).
\]

Next, we derive a simplified version of the augmented ADMM. \((3.21c)\) and \((3.21e)\) imply \(\tilde{U}^{(i)} = 0\). Therefore, we get the following simplified augmented ADMM updates:

\[
X^{(i+1)} = \arg \min_{X \in \mathcal{F}^k} \langle S, X \rangle + \langle 2U^{(i+1)} - U^{(i)}, DXD^T \rangle + \frac{\sqrt{c\rho}}{2}\| X - X^{(i+1)} \|_F^2,
\]

\[
Y^{(i+1)} = \arg \min_{Y \in \mathbb{R}^{(p-d) \times (p-d)}} \frac{1}{2} \| DX^{(i+1)}D^T - Y + U^{(i)} \|_F^2 + \frac{\lambda}{\rho} \| Y \|_{1,1},
\]

\[
U^{(i+1)} = DX^{(i+1)}D^T - Y^{(i+1)} + U^{(i)},
\]

which do not involve the augmented variable. \((3.22a)\) is a Fantope projection problem introduced by Vu et al. (2013). \((3.22b)\) is a lasso problem of matrix form. Both \((3.22a)\) and \((3.22b)\) have analytic solutions. This formulation of augmented ADMM can be generalized to penalty \(\| DX \|_{1,1}\) and \(\| DX \|_{2,2}\) with some modifications. For the penalties \(\| DX \|_{1,1}\) and \(\| DX \|_{2,2}\), the augmented ADMM algorithms involve solving the subproblems of the forms

\[
\min_{Y \in \mathbb{R}^{(p-d) \times p}} \frac{1}{2} \| DX^{(i+1)} - Y + U^{(i)} \|_F^2 + \frac{\lambda}{\rho} \| Y \|_{1,1},
\]

\[
\min_{Y \in \mathbb{R}^{(p-d) \times p}} \frac{1}{2} \| DX^{(i+1)} - Y + U^{(i)} \|_F^2 + \frac{\lambda}{\rho} \| Y \|_{2,2},
\]

respectively. \((3.23a)\) and \((3.23b)\) are the column-wise lasso problems with the \(l_1\) norm penalty and the squared \(l_1\) norm penalty. We introduce an specialized algorithm for
solving (3.23b) in the next section.

### 3.3 Lasso Problem with Squared $l_1$ Norm Penalty

For the penalty $\|DX\|_{1,1}$, the subproblem (3.23a) can be solved by a soft-thresholding operator. For the mixed norm penalty $\|DX\|_{1,2}^2$, the subproblem (3.23b) is a column-wise lasso problem with squared $l_1$ norm penalty which has the form

$$\min_{x \in \mathbb{R}^p} \frac{1}{2}\|y - x\|_2^2 + \lambda \|x\|_1^2.$$  

(3.24)

To solve (3.24), we first introduce the definition of subdifferential and two lemmas. The first lemma helps us to find the subdifferential of the function $\|x\|_1^2$. The second lemma connects the lasso problem and the lasso problem with the squared $l_1$ norm penalty.

**Definition 2** (Subgradient and subdifferential). A subgradient of a convex function $f : \mathbb{R}^p \to \mathbb{R}$ at point $x_0$ in the domain of $f$ is a vector $v \in \mathbb{R}^p$ such that

$$f(x) - f(x_0) \geq v^T(x - x_0)$$  

(3.25)

for all $x$ in the domain of $f$. The set of all subgradients is called the subdifferential of the function $f$ at $x_0$.

**Lemma 1.** Suppose that $x \in \mathbb{R}^p$, $u \in \partial \|x\|_1$ are given. Then $2\|x\|_1 u \in \partial \|x\|_1^2$.

**Lemma 2.** Given a fixed $\lambda > 0$, there exists a $\mu > 0$ such that the solution $x_\mu$ to the lasso problem

$$\min_{x \in \mathbb{R}^p} \frac{1}{2}\|y - x\|_2^2 + \mu \|x\|_1$$  

(3.26)

satisfies $\mu = 2\lambda \|x_\mu\|_1$.
Since the solution \( x_\mu \) to (3.26) satisfies the first-order optimality condition

\[
y - x_\mu \in \mu \partial \| x_\mu \|_1 = 2\lambda \| x_\mu \|_1 \partial \| x_\mu \|_1,
\]

(3.27)

Lemma 2 implies that (3.27) is also the optimality condition of (3.24). Thus \( x_\mu \) is a solution to (3.24). This observation provides us with a strategy to get a solution to (3.24). Let \( g(\mu) = 2\lambda \| x_\mu \|_1 \) be a function of \( \mu > 0 \). \( g(\mu) \) is continuous and decreasing in \( \mu \) since \( \| x_\mu \|_1 \) decreases as \( \mu \) increases. So \( G(\mu) = g(\mu) - \mu \) is also continuous and decreasing. Then, we can construct a lasso problem by using the root of the function \( G \) as the tuning parameter. The solution to this lasso problem is also a solution to (3.24). Accordingly, the key step is to find the root of \( G(\mu) \). We need the following lemma to compute the root of the function \( G \).

**Lemma 3.** The monotonically decreasing function \( G \) is piecewise linear.

Lemma 3 is obtained directly by analyzing the closed form of the solution to (3.26)

\[
x_{\mu,i} = \begin{cases} 
0 & \text{if } |y_i| \leq \mu, \\
\text{Sign}(y_i)(|y_i| - \mu) & \text{if } |y_i| > \mu.
\end{cases}
\]

(3.28)

The knots of the piecewise linear function \( G(\mu) \) consist of the absolute value of the entries of \( y \). We sequentially evaluate the \( G(\mu) \) from its smallest knot to its largest one. According to Lemma 2, we can find that \( \mu_i \) and \( \mu_{i+1} \) satisfy \( G(\mu_i) > 0 \) and \( G(\mu_{i+1}) < 0 \). Then \( \mu^* = \mu_i - \frac{\mu_i - \mu_{i+1}}{G(\mu_i) - G(\mu_{i+1})} G(\mu_i) \) is the root of \( G(\mu) \) based on Lemma 3. Therefore, the solution to (3.26) with tuning parameter \( \mu^* \) is also the solution to (3.24) with tuning parameter \( \lambda \). We summarize the algorithm for solving (3.24) as below.

where \( \text{softTh}(y, \mu) \) is a soft-thresholding operator with input vector \( y \) and tuning parameter \( \mu \).
Algorithm 1 Algorithm for lasso problem with squared $l_1$ norm penalty

Initialize $i = 0$, $\mu_i = 0$ and $x^*_\mu = y$.

while $i \leq p$ do
  $i \leftarrow i + 1$
  if $G(\mu) = 2\|x^*_\mu\|_1 - \mu_{i-1} > 0$ then
    $\mu_i \leftarrow |y|_{[i]}$ and $x^*_\mu \leftarrow \text{softTh}(y, \mu_i)$
  else
    Compute $\mu^*$ by $\mu^* = \mu_i - \frac{\mu_i - \mu_{i-1}}{G(\mu_i) - G(\mu_{i-1})}G(\mu_i)$
    $x^* \leftarrow \text{softTh}(y, \mu^*)$, break.
  end if
end while

3.4 Computational Complexity

The $X$-update in the augmented ADMM algorithm is a Fantope projection problem with the computational complexity $O(p^3)$. For penalty $\|DX\|_{1,1}$ and $\|DXDT\|_{1,1}$, the $Y$-update in the augmented ADMM algorithm is a lasso problem which can be solved by a soft-thresholding operator with complexity $O(p^2)$. For the mixed norm penalty, the $Y$-update is a lasso problem with the squared $l_1$ norm penalty. We solve it by the specialized algorithm introduced in the previous section, which has the complexity $O(p^3)$. Therefore, the total computational complexity of a single augmented ADMM iterate is $O(p^3)$.

3.5 Selection of $\lambda$

The TFPCA problem involves a single tuning parameter $\lambda$ which controls the roughness of estimates. We use the leave-out-one-curve cross-validation to choose the tuning parameter $\lambda$, which is conducted as follow:

1: Subtract the mean curve from the observed data to obtain the residual matrix $R$.
2: Let $R^{(-i)}$ be the residual matrix with $i$th residual curve $\epsilon^{(i)}$ out.
3: Use $R^{(-i)T}R^{(-i)}$ as input matrix of TFPCA and estimate the projection matrix $X^{(i)}_{\lambda}$ of the principal subspace.

4: Select the optimal tuning parameter $\lambda$ which minimizes the cross-validation criterion

$$CV(\lambda) = \frac{1}{n} \sum_{i=1}^{n} \epsilon^{(i)T}(I - X^{(i)}_{\lambda})\epsilon^{(i)}$$

which is the averaged reconstruction SSE.

It is also necessary to determine the range of the tuning parameter values. In the lasso problem, we search for the tuning parameter between 0 and $\lambda_{\text{max}}$, where $\lambda_{\text{max}}$ is the value of tuning parameter that results in the zero solution. Similarly, we let $\hat{X}_{\lambda_{\text{max}}}$ be the solution to the TFPCA problem with tuning parameter $\lambda_{\text{max}}$ which satisfies $\hat{X}_{\lambda_{\text{max}}} \in \arg\min_{X \in F} \Omega(X)$. We call $\hat{X}_{\lambda_{\text{max}}}$ the trivial solution. In order to determine the range of tuning parameter values, we first need to find the set of the trivial solutions. Then, we compute the $\lambda_{\text{max}}$ that results in the trivial solutions.

The technical details are introduced in Appendix A. As for the mixed norm, we use

$$\|DX\|_{1,2}^2$$

instead of $\|DX\|_{1,2}$ to simplify the computation. Therefore, there is no finite $\lambda$ such that the TFPCA solution is trivial (this is similar to the ridge regression problem). In practice, as we gradually increase $\lambda$, the roughness penalty gets smaller. So we increase the tuning parameter until the penalty $\|DX\|_{1,2}$ does not change much. Then, we can use the $\lambda$ we get finally as the upper bound of the search range.
Chapter 4: Simulation Studies

In this chapter, we carry out simulation studies to understand TFPCA further. We compare the TFPCA with PMA under different simulation designs and provide numerical evidence that TFPCA outperforms PMA when the eigengaps of the sample covariance matrix are small. The principal subspace of is spanned by the leading eigenvectors of the population covariance matrix. We also show that TFPCA with penalty $\|DX\|_{1,1}$ favors an estimated projection matrix with its corresponding eigenvectors that have a small $l_1$ norm. Therefore, TFPCA with penalty $\|DX\|_{1,1}$ is better than TFPCA with penalty $\|DXD^T\|_{1,1}$ when the eigenvectors of the population covariance matrix have a small $l_1$ norm.

4.1 Comparison between TFPCA and PMA

We first conduct a simulation to illustrate the effectiveness of TFPCA in estimating the projection matrix $\Pi$ of a $k$-dimensional principal subspace compared with PMA. PMA estimates the $k$ leading principal component directions $\hat{v}_1, \cdots, \hat{v}_k$ sequentially. We use the projection matrix of the subspace spanned by $\hat{v}_1, \cdots, \hat{v}_k$ as the estimated projection matrix of PMA. The tuning parameter of each method is selected by searching for the best parameter value over a grid such that the SSE $\|\hat{X}_\lambda - \Pi\|_F^2$ of the corresponding estimates $\hat{X}_\lambda$ is the minimum. Since the selected
tuning parameter value minimizes the SSE $\|\hat{X}_\lambda - \Pi\|_F^2$ with the known true projection matrix $\Pi$, we call such a choice of the tuning parameter value the oracle one.

We simulate data from the following scheme. The data of sample size $n$ are generated from a multivariate Gaussian distribution with zero mean vector and population covariance $\Sigma$. We consider the projection matrix $\Pi$ of the subspace spanned by the columns of a $p \times k$ orthogonal matrix $V$. We generate the columns of $V$ by selecting $k$ alternating vectors shown in Figure 4.1.

Figure 4.1: The examples of the alternating vectors used as the population eigenvectors in the first simulation study with different roughness.
The alternating eigenvectors displayed in Figure 4.1 have a unit $l_2$ norm with the same $l_1$ norm of $\sqrt{p}$. We vary the roughness of these vectors by changing the number of knots $s$. Next, we orthonormalize the $k$ selected alternating vectors such that $V$ is orthogonal and let $\Pi = VV^T$. We then generate the population covariance $\Sigma$ by

$$\Sigma = \delta \Pi + I,$$

(4.1)

where $I$ is a $p \times p$ diagonal matrix and $\Pi = VV^T$ is a $p \times p$ projection matrix. For the population covariance of the form (4.1), the leading $k$ eigenvalues of $\Sigma$ are $(1 + \delta)$ and $\Pi$ is the projection matrix of the $k$-dimensional principal subspace spanned by the corresponding eigenvectors. The rest of the eigenvalues are 1. We can vary the gap between the $k$th and the $(k + 1)$th eigenvalues of the population covariance matrix $\Sigma$ by changing the parameter $\delta$. If $k > 1$, the leading $k$ eigenvectors are unidentifiable. A small value of $\delta$ is for the scenario where the eigengap between the $k$th and the $(k + 1)$th is small. Another motivation for the covariance structure in (4.1) is that data sampled from $\mathcal{N}_p(0, \delta \Pi + I)$ also follow the factor model:

$$x_i = \sum_{j=1}^{k} v_i \xi_{i,j} + \epsilon_i,$$

(4.2)

where $\xi_{i,j} \sim \mathcal{N}(0, \delta^2)$ and $\epsilon_i \sim \mathcal{N}_p(0, I)$. Estimating $\Pi$ helps us to recover the orthogonal latent factors $v_1, \cdots, v_p$, and to evaluate the variance explained by $v_1, \cdots, v_p$. In this simulation study, we set the dimension $p = 50$, $k \in \{1, 2\}$, the sample size $n \in \{25, 50, 100, 200\}$, $\delta \in \{0.1, 0.25, 0.5, 1, 2, 4, 8\}$, and $s \in \{4, 6, 8, 10\}$. For each combination of the simulation parameters, we generate 100 replicates of the simulated data and estimate the projection matrix of the $k$-dimensional principal subspace by TFPCA with penalty $\|DXD^T\|_{1,1}$ and $\|DX\|_{1,1}$, PMA, and the ordinary PCA.
Figure 4.2: The averaged SSEs of TFPCA with penalty $\|DX\|_{1,1}$ and $\|DX^{T}\|_{1,1}$, PMA, and the ordinary PCA across 100 replicates for each combination of the simulation parameters ($k = 1$)

The results for $k = 1$ are summarized in Figure 4.2. The column of the grid corresponds to the number of knots $s$ and the row of the grid corresponds to the sample size $n$. In each panel, the averaged SSEs $\|\hat{\Pi} - \Pi\|_F^2$ across the 100 replicates are plotted against the eigengap parameter $\delta$. PMA and TFPCA give better estimates of $\Pi$ than that of the ordinary PCA as expected. The performance of TFPCA with penalty $\|DX^{T}\|_{1,1}$ is comparable to that of PMA. TFPCA with $\|DX\|_{1,1}$ is worse than PMA in most cases. To make a clearer comparison, we compute the ratios...
\[
\frac{SSE_{dx}}{SSE_{pma}} \text{ and } \frac{SSE_{de}}{SSE_{pma}} \text{ and plot the ratios against the eigengap } \delta. \text{ The 95\% confidence bands of the mean ratios are also included.}
\]

Figure 4.3: Comparison between TFPCA with \( \|DX\|_{1,1} \) and PMA (\( \frac{SSE_{de}}{SSE_{pma}}, k = 1 \))

The x-axis and y-axis in Figure 4.3 are in log_{10} scale. If the ratio is greater than 1, the SSE of the estimates given by TFPCA with penalty \( \|DX\|_{1,1} \) is greater than the SSE of PMA and vice versa. The red horizontal line at 1 is for reference. Figure 4.3 shows that TFPCA with penalty \( \|DX\|_{1,1} \) is not uniformly worse than PMA. When the eigengap \( \delta \) is small, TFPCA with penalty \( \|DX\|_{1,1} \) produces smaller SSEs.
Figure 4.4: Comparison between TFPCA with $\|DXD^T\|_{1,1}$ and PMA ($\frac{SSE_{did}}{SSE_{pma}}$, $k = 1$)

Figure 4.4 shows that TFPCA with penalty $\|DXD^T\|_{1,1}$ is as good as PMA in our simulation settings since the 95% confidence band covers the horizontal red line at 1 in most settings.

Next, we set $k = 2$ and run a similar study. Since we have two population eigenvectors, we will use the number of non-zeros in $D\Pi D^T$ to quantify the roughness. In the grid displayed in Figure 4.5, the column corresponds to the number of non-zeros in $D\Pi D^T$. 

39
For $k = 2$, the results are similar to the results with $k = 1$. TFPCA with penalty $\|DXD^T\|_{1,1}$ and PMA are better than TFPCA with penalty $\|DX\|_{1,1}$. And we could also see a clearer difference between TFPCA with $\|DXD^T\|_{1,1}$ and PMA. The SSE of TFPCA with $\|DXD^T\|_{1,1}$ is smaller than the SSE of PMA when the eigengap is small.
Figure 4.6: Comparison between TFPCA using $\|DX\|_1$ and PMA ($\frac{SSE_{dx}}{SSE_{pma}}$, $k = 2$)

After investigating the ratio $\frac{SSE_{dx}}{SSE_{pma}}$, we observe that TFPCA with $\|DX\|_1$ outperforms PMA only when the eigengap is small. In most of the other cases, the ratio of SSEs is greater than 1. When the eigengap is sufficiently large, the ratio is getting close to 1.
Figure 4.7: Comparison between TFPCA using $\|DXD^T\|_{1,1}$ and PMA ($\frac{SSE_{ed}}{SSE_{pma}}$, $k = 2$)

Figure 4.7 shows that TFPCA with $\|DXD^T\|_{1,1}$ is better than PMA when eigengap is small. For larger eigengap, the TFPCA estimate is at least as good as PMA estimate.

The findings in this simulation study support our claim that TFPCA has some advantages in estimating the principal subspace spanned by multiple eigenvectors when the eigengap is small. And the results also show that the TFPCA with $\|DXD^T\|_{1,1}$ could give an estimate that is as good as the PMA estimate when the eigengap is large. TFPCA with $\|DX\|_{1,1}$ is advantageous for the smaller eigengap, but it becomes
worse as the eigengap increases.

4.2 Comparison among the TFPCA penalties $\|DX\|_{1,1}$, $\|DXD^T\|_{1,1}$ and $\|DX\|_{1,2}^2$

In this section, we design a simulation study to illustrate the shrinkage effect of the penalty $\|DX\|_{1,1}$ on the $l_1$ norm of the estimated eigenvectors. We simulate data following the same scheme described in the previous study. $\Pi$ is the projection matrix of a two-dimensional subspace that we specified in (4.1). We design two settings and display the two leading population eigenvectors of each setting in Figure 4.8 and Figure 4.9.
Figure 4.8: The population eigenvectors used in the first setting has a small $l_1$ norm. In other words, most entries of the eigenvectors are close to zero, while a few entries are far from zero. We can see spikes in both eigenvectors.

Figure 4.8 shows the orthonormal bases $(y_1, y_2)$ of the two-dimensional subspace in the first simulation setting. $\|y_1\|_1 = 4.264377$ and $\|y_2\|_1 = 3.933451$. The maximum $l_1$ norm of a unit $l_2$ norm vector in $\mathbb{R}^{50}$ is $\sqrt{50} = 7.071068$. 
Figure 4.9: The population eigenvectors used in the second setting have a large $l_1$ norm. In other words, the entries of the eigenvectors are relatively far away from zero, which produces vectors of larger $l_1$ norm.

Figure 4.9 shows the orthonormal bases $(y_1, y_2)$ which span the subspace for the second setting with $\|y_1\|_1 = 6.259669$ and $\|y_2\|_1 = 5.881145$. Compared with the first setting, the population eigenvectors in the second setting have a larger $l_1$ norm. We set $\delta = 1$ and simulate 50 datasets. For each dataset, we let the trace constraint $k = 2$ and the difference order $d = 2$. With the oracle choice of the tuning parameter, we apply the TFPCA to the simulated data, and compute the SSE of the estimated projection matrix given by TFPCA with penalty $\|DX\|_{1,1}$, $\|DX\|_{1,2}^2$, and $\|DXD^T\|_{1,1}$. 

45
Figure 4.10: SSEs of the estimated projection matrix given by TFPCA with penalties \(\|DX\|_{1,1}, \|DX\|_{1,2}^2\) and \(\|DXD^T\|_{1,1}\) in the first setting.

Figure 4.10 displays the scatter plots of SSEs for the three penalties used in TFPCA in the first simulation setting. We set the upper limit of the scatterplot to be 4. Let \(X \in \mathcal{F}^2\) and \(\Pi\) be a rank-2 projection matrix. Since both \(X\) and \(\Pi\) are non-negative definite, \(\langle X, \Pi \rangle \geq 0\). Next, we have

\[
\|X - \Pi\|^2_F = \|X\|^2_F + \|\Pi\|^2_F - 2\langle X, \Pi \rangle. \tag{4.3}
\]

\(\|X - \Pi\|^2_F\) reaches the maximum when the subspaces associated with \(X\) and \(\Pi\) are orthogonal, which implies \(\langle X, \Pi \rangle = 0\). Combining with \(\|X\|^2_F \leq 2\) and \(\|\Pi\|^2_F = 2\), we have \(\max_{X \in \mathcal{F}^2} \|X - \Pi\|^2_F = 4\). Each point represents a simulated data set. The solid black line has a slope 1 and an intercept 0. We observe that the estimates given by the penalty \(\|DX\|_{1,1}\) have smaller SSE in most cases, compared with the penalties \(\|DXD^T\|_{1,1}\) and \(\|DX\|_{1,2}^2\).
Figure 4.11 shows the scatterplots of SSEs for the three penalties used in TFPCA in the second simulation setting. The points in the left panel are around the solid line, which means that $\|DX\|_{1,1}$ and $\|DXD\|_{1,1}$ have similar performance. For the other two panels, more points lie below the red line, which indicates that penalty $\|DX\|_{2,1}^2$ is slightly better than $\|DX\|_{1,1}$ and $\|DXD^T\|$ in this setting. In summary, the estimates given by TFPCA with penalty $\|DX\|_{1,1}$ outperform the estimates given by TFPCA with penalties $\|DX\|_{1,2}^2$ and $\|DXD^T\|$ when the population eigenvectors have small $l_1$ norm. However, how to quantify which penalty is better requires further investigation. In practice, we could combine the selection of tuning parameter with the selection of penalty type. Specifically, we could use leave-out-one-curve cross-validation and select the combination of penalty and tuning parameter that give the smallest cross-validation criterion.
Chapter 5: Real Data Analysis

In this chapter, we apply our proposed method to two real datasets to demonstrate the utilities and merits of TFPCA. In multivariate analysis, the principal components directions are used to describe the principal modes of variation among the observed data (Castro et al., 1986). In the first analysis, we apply TFPCA and the method introduced by Silverman (1996) to the pinch force data which we used as the motivating example in Chapter 1. The pinch force data has 20 curves which measure the force of squeezing a force meter with thumb and forefinger by a single subject during a short period. Following the procedure presented by Ramsay (2006), we display the principal modes of variation of the pinch force data to understand the process of squeezing an object. The second dataset consists of records of daily temperature in Montreal from 1961 to 1994. We apply TFPCA to the data of summer months (June, July, and August) and demonstrate how to display the majority of variation by the projection of the data on the principal subspace.

5.1 Pinch Force Data

The pinch force data were first collected at the Medical Research Council of the Applied Psychology Unit, Cambridge, in order to understand the muscle groups controlling the thumb and forefinger in the act of gripping an object. The dataset includes
twenty experimental records of a single subject. The subject was required to maintain 2 Newtons force on a force meter and then to squeeze the meter aiming at 10 Newtons, finally to return to 2 Newtons. The data was analyzed by Ramsay (2006, Chapter 7). We use the version of data available from the R package fda (Ramsay et al., 2014).

![Figure 5.1: The pinch force data](image)

The original dataset has 300 measurements per curve, and the recording of each curve begins at an arbitrary time index. To make the analysis across the curves meaningful, each curve has been shifted horizontally such that every curve reaches its maximum at the same time index. More specifically, a subset of 151 consecutive measurements is taken from each curve such that measurement in every subset reaches
the maximum at 39th time index. Figure 5.1 displays the curves after the alignment. The black horizontal line is at 2 Newtons. The first question is how to visualize the variation when we begin to explore the data. In the following analysis, we will use TFPCA to estimate the leading principal component directions and display the principal modes of the variation by the estimated PC directions.

![Figure 5.2: The estimated mean by the $l_1$ Trend Filtering with tuning parameter selected by 5-fold cross-validation](image)

The principal modes of variation are typically displayed by multiple curves. Each curve is obtained by adding the estimated PC directions multiplied by scalars to the estimated mean. We can visualize the amount of variation by the width of the band formed by the multiple curves. In order to get a smooth band, we need both the estimated mean curve and the estimated PC directions to be smooth. So our first
step is to get a smooth estimate of the mean curve. We apply the $l_1$ Trend filtering to smooth the sample mean curve. The tuning parameter is selected by a specialized 5-fold cross-validation designed for $l_1$ Trend filtering. Suppose we have a curve of 20 points and number of folds is 5. The points are assigned to folds as below:

$x, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 1, 2, 3, x,$

where $x$ indicates no assignment. When calculating the cross-validation error, the first and the last points with no assignment are excluded, and the predicted value at a point is given by the average of the fitted value of the two neighbors (Arnold and Tibshirani, 2014). See the manual of the R package ‘genlasso’ for more technical details. Then, we smooth the sample mean vector by $l_1$ Trend filtering with the selected tuning parameter. The resulting $l_1$ Trend filtered mean curve will be used to display the principal modes. Before applying TFPCA to estimate the PC directions, we need to select the trace constraint parameter $k$. In the ordinary PCA, the number of principal components $k$ is often determined by the cumulative variance that they capture. We may take first $k$ eigenvectors that capture at least 95% or 99% cumulative variance in the data. For TFPCA, we can evaluate the variance in the $k$-dimensional principal subspace by the matrix trace product. Let $\hat{X}_k$ be the TFPCA estimate with the tuning parameter selected by leave-out-one-curve cross-validation. The trace product $\langle \hat{X}_k, S \rangle$ measures the variation in the principal subspace. In our analysis, we simply choose the $k$ by the heuristic method of the ordinary PCA. The method will give a conservative choice of $k$ since the cumulative variance captured by the $k$ leading eigenvectors of $S$ is greater than or equal to $\langle \hat{X}_k, S \rangle$ for all $\lambda \geq 0$. This observation is based on the equality $VV^T = \arg \max_{X \in \mathcal{F}_k} \langle X, S \rangle$, where $V \in \mathbb{R}^{p \times k}$ consists of the $k$ leading eigenvectors of $S$. 

51
Figure 5.3 shows that the leading four principal components obtained by the ordinary PCA have explained about 95% of the variability among the residuals. Thus, we decide to choose $k = 4$. In this analysis, we want to describe how the variation changes across the time index. A principal mode with the upper bound and lower bound being piece-wise linear functions will be enough for this purpose. Higher difference order will achieve the same goal. However, it also requires heavier computation. So we let $d = 2$. The estimates of the first four principal component directions given by TFPCA with penalty $\|D^{(2)}X\|_{1,1}$, $d = 2$, and $k = 4$ are plotted in Figure 5.4.
Figure 5.4: Comparison between the PC directions given by TFPCA and Silverman’s method

For comparison, we also apply the method proposed by Silverman (1996). The leave-out-one-curve CV is applied to select the smoothing parameter for TFPCA and Silverman’s method. Silverman’s method selects $\alpha = 20.2359$ while TFPCA selects $\lambda = 0.234$. The estimated four leading principal component directions are visualized.
in Figure 5.4. The four panels correspond to the estimated PC directions given by the TFPCA and Silverman’s method. We could see the level of roughness of the estimated PC directions are different across the domain. We use the estimates of the fourth principal component direction for an illustration. Both TFPCA and Silverman’s method give similar fits before index 75 exhibiting the two peaks and one valley. However, TFPCA estimate is nearly linear after index 75 while the estimate by Silverman’s method becomes very jagged. We observe the similar patterns in the estimates of the other three principal component directions. This result exemplifies the merits of TFPCA in estimating principal component directions with variable roughness.

Next, we display the principal modes of variation by adding the estimated principal component directions multiplied by the minimum, median and maximum of the associated principal components scores to the estimated mean curve.
Figure 5.5: The first four principal modes of variation given by TFPCA. The dotted lines correspond to the median principal component scores. The solid lines correspond to the maximum and the minimum principal component scores.

Figure 5.5 shows the four principal modes of variation with the PC directions estimated by TFPCA. We see that the first mode of variation corresponds to the variation
when the subject releases the meter. The second mode of variation describes the vari-
ation when the “squeeze” takes place. The third and the fourth modes illustrate the
variation when the subject maintains the force at the baseline, at the peak, and after
squeezing.

5.2 Montreal Summer Temperature Data

In some scenarios, the gaps between the eigenvalues of the sample covariance
matrix are small. In this case, estimating the projection matrix of the principal
subspace spanned by $k$ leading eigenvectors is more reliable than estimating the $k$
individual eigenvectors. In this section, we use Montreal temperature data as an
example to demonstrate how to use the projection of the data to describe the variation
in a lower dimensional principal subspace. This dataset is analyzed by Ramsay et al.
(2009, Section 4.3). We use the version of data available from the R package fda
(Ramsay et al., 2014). We only use the data from June to August in the following
analysis.
Figure 5.6: The Montreal summer temperature data

Figure 5.6 displays the Montreal daily temperature in degrees Celsius during June, July, and August for each year from 1961 through 1994. The red curve is the daily mean temperature during the three summer months (June, July, and August).
Figure 5.7 shows the scree plot given by ordinary PCA for initial exploration. It shows that the gaps between eigenvalues of the sample covariance matrix are small. Therefore, estimating the individual principal component directions will be unreliable. To demonstrate the merits of TFPCA in this scenario, we apply both TFPCA with penalty $\|D^{(1)}X\|_{1,1}$ and PMA to this dataset. For PMA, we use the projection matrix of the subspace spanned by the estimated eigenvectors as the estimated projection matrix. Then, we evaluate the results by the in-sample reconstruction MSE for the
different choices of $k$. The in-sample reconstruction MSE is given by

$$\text{MSE}(\hat{X}\hat{\Pi}_k) = \frac{1}{np} \| X - X\hat{\Pi}_k \|_F^2,$$

where $\| \cdot \|_F$ is the Frobenius norm, $X$ is the data matrix and $\hat{\Pi}_k$ is the estimated projection matrix. For each $k$ from 1 to 10, we select the tuning parameters by leave-out-one-curve cross-validation and compute the in-sample reconstruction MSEs of the estimates given by PMA and TFPCA. The results are illustrated in Figure 5.8.

![In-sample reconstruction MSE of TFPCA and PMA](image_url)
From Figure 5.8, we see that the TFPCA has smaller in-sample reconstruction MSEs than PMA for all $k$’s. This result provides numerical evidence for our claim that TFPCA provides a better estimate of the principal subspace when the gaps between eigenvalues of the sample covariance are small. Since it is unreliable to estimate individual eigenvectors in this example, we will study the variation of data in the principal subspace instead of estimating individual eigenvectors and displaying the principal modes. To achieve this goal, we project the data on the estimated principal subspace. This procedure reduces the dimension of data and helps us to explore the variation of data in a lower dimensional subspace.

![Figure 5.9: The projected data for $k = 3, 6, 9$ by TFPCA](image)

The top left panel of Figure 5.9 shows the observed data. The other three panels show the reconstructed curves given by adding the projections of the de-meaned curves on the $k$-dimensional principal subspace to the estimated mean curve. The
projections are obtained by TFPCA with $d = 1$ and $k = 3, 6, 9$. We can observe that the projected curves in each panel form a band around the estimated mean curve. Narrower band indicates less variation and vice versa. We observe that the reconstructed curves for $k = 3$ are oversmoothed, since there is almost no variation during the two time periods in July and August. Comparing the cases of $k = 6$ and $k = 9$, we see that they describe similar patterns. Both results reveal that there is only a few variation during the short time around July 15th and August 20th. And $k = 6$ is enough to describe the majority of the variation in the data since we could not extract additional information from the case of $k = 9$. 
Chapter 6: Conclusion and Discussion

In this dissertation, we proposed a new method for smoothed PCA based on the convex relaxation of PCA and $l_1$ Trend filtering. The first advantage of our proposed method is the local adaptiveness. TFPCA estimates inherit this merit from the penalty based on a $l_1$ Trend filtering. The second advantage is that TFPCA gives better estimates of the projection matrix of the principal subspace. There are two explanations for this merit. When the leading eigenvalues are identical, estimating individual eigenvectors suffers from identifiability issue. Additionally, the convex relaxation framework provides more flexibility since the constraint set $\mathcal{F}^k$ is the convex hull of all rank-$k$ projection matrix. We provided a simulation study and a real data analysis to support our claims. We also introduced the motivations of each penalty and discussed the difference between the penalties along with the effect they could have on the TFPCA estimate. The penalty $\|DX\|_1$ favors estimates with eigenvectors that have a small $l_1$ norm. A simulation study is conducted to support our conjecture on this effect. The current work still has some limitations, and our future work will include the following aspects.

So far, we only discussed the settings where data are sampled from an equally spaced grid. With the modification of the difference matrix $D$, we could extend our method to more general situations. Suppose data are sampled from a set of common but
unequally-spaced grid \( \{t_1, \cdots, t_p\} \). As discussed by Tibshirani (2014), we consider the difference matrix defined recursively of the following form:

\[
D^{(t,d+1)} = D^{(1)} \text{diag}(\frac{d}{t_{d+1} - t_1}, \frac{d}{t_{d+2} - x_2}, \cdots, \frac{d}{t_n - t_{n-d}}) \cdot D^{(t,d)},
\]

(6.1)

where \( D^{(1)} \) on the left hand side of (6.1) is the regular first order difference matrix of size \( (n - d - 1) \times (n - d) \). \( D^{(t,1)} \in \mathbb{R}^{(n-d)\times(n-d-1)} \) is also the regular difference matrix of the first order. \( D^{(t,d+1)} \) can be viewed as a difference matrix of order \( d + 1 \) adjusted to the unevenly spaced grid \( \{t_1, \cdots, t_p\} \). By using this extended version of the difference matrix, we can extend TFPCA to the regime when data are sampled unevenly.

Second, the \( l_1 \) trend filtering estimate is closely related to falling factorial basis (Wang et al., 2014). The falling factorial basis functions \( h_1, \cdots, h_p \) are defined over an equally spaced grid \( \{\frac{1}{p}, \frac{2}{p}, \cdots, \frac{p-1}{p}, 1\} \) of the following form

\[
\begin{align*}
  h_1(t) &= 1 \\
  h_2(t) &= t \\
  h_3(t) &= t^2 \\
  &\vdots \\
  h_{d+1}(t) &= t^d \\
  h_{d+1+j}(t) &= \prod_{l=1}^{d} (t - t_{j+l}) \cdot 1\{t \geq t_{j+d}\}, j = 1, \cdots, n - d - 1
\end{align*}
\]

(6.2)

Let \( H_{ij} = h_j(t_i) \), for \( i, j = 1, \cdots, p \). Tibshirani (2014) has shown that the solution \( \hat{\beta} \) to the \( l_1 \) Trend filtering problem

\[
\min_{\beta \in \mathbb{R}^p} \frac{1}{2} \|y - \beta\|_2^2 + \lambda \|D^{(d)} \beta\|_1,
\]

(6.3)
satisfies $\hat{\beta} = H\hat{\alpha}$, where $\hat{\alpha}$ is the solution to the following lasso problem:

$$
\min_{\alpha \in \mathbb{R}^p} \frac{1}{2} \|y - H\alpha\|_2^2 + \lambda \sum_{j=d+2}^{p} |\alpha_j|.
$$

(6.4)

This indicates that the $l_1$ Trend filtering estimate is a realization of a function spanned by the falling factorial basis functions. In the further study, we expect to connect the TFPCA estimates to the falling factorial basis and extend our method to Functional PCA.

Third, we still have difficulties in computing the tuning parameter that can provide the trivial solution in more general cases where the trace constraint $k$ is greater than the difference order $d$. We will figure out how to find the trivial solutions in the general scenario and how to compute $\lambda_{max}$. Moreover, the numerical experiments reveal the difference between TFPCA penalties. We only propose to choose the type of penalty by leave-out-one-curve cross-validation. In the future, we expect to quantify the difference in the estimates using different TFPCA penalties and develop a more convincing rule for choosing the type of TFPCA penalty.
Appendix A: Some Technical Details

A.1 Computation of $\lambda_{\text{max}}$

In the following discussion, we only consider the scenario where differencing order $d$ is greater than the trace constraint $k$. When applying $l_1$ norm regularized method to dataset, we need to determine the range for selecting the tuning parameter $\lambda$, which is between 0 and $\lambda_{\text{max}}$. For example, $\lambda_{\text{max}}$ is the parameter resulting in the zero solution in the lasso problem. For TFPCA, $\lambda_{\text{max}}$ works in a similar role which shrinks the penalty $\|DX\|_{1,1}$ or $\|DXD\|_{1,1}$ towards the minimum. We first write out the general form of the TFPCA problem,

$$
\begin{align*}
\text{maximize} & \quad \langle S, X \rangle - \lambda h_C(X) \\
\text{subject to} & \quad X \in \mathcal{F}^d,
\end{align*}
$$

(A.1)

where

$$
h_C(X) = \max_{Z \in C} \langle Z, X \rangle
$$

(A.2)

is the support function of a nonempty closed convex set $C$. 

$$
C = \begin{cases} 
\{D^TZD : \|Z\|_\infty \leq \lambda\}, & h_C(X) = \|DXD^T\|_{1,1} \\
\{D^TZ + ZD : \|Z\|_\infty \leq \lambda\}, & h_C(X) = \|DX\|_{1,1} + \|DXD^T\|_{1,1}
\end{cases}
$$

(A.3)

The larger value of the tuning parameter $\lambda$ favors smaller value of $h_C(X)$. Before computing the value or the upper bound of $\lambda_{\text{max}}$, we first define the trivial point of TFPCA as follow.
**Definition 3** (trivial points of TFPCA).

\[ T = \arg \min_{X \in \mathcal{F}} h_C(X) = \{ X_\ast \in \mathcal{F}^d : h_C(X_\ast) = \min_{X \in \mathcal{F}^k} h_C(X) \}, \]

\( T \) is the subset of feasible set that minimize the penalty \( h_C(X) \), which is also the solution to (A.1) with input matrix \( S = I \). Then, we look into the TFPCA with penalty \( \|DXD^T\|_{1,1} \) to characterize the trivial solution. Since \( C \) and \( \mathcal{F}^d \) are non-empty closed convex sets,

\[
\min_{X \in \mathcal{F}^d} h_C(X) = \min_{X \in \mathcal{F}^d} \max_{Y \in C} \langle Y, X \rangle \\
= \max_{Y \in C} \min_{X \in \mathcal{F}^d} \langle Y, X \rangle \\
= -\min_{Y \in C} \max_{X \in \mathcal{F}^d} \langle -Y, X \rangle \\
= -\max_{\|Z\|_{\infty, \infty} \leq 1} \min_{X \in \mathcal{F}^d} \langle -DTZD, X \rangle \\
= -\min_{\|Z\|_{\infty, \infty} \leq 1} \sum_{j=1}^{d} \lambda_j \langle -DTZD \rangle \\
= \langle -DTZD \rangle \\
\]

(A.4)

Since \( Z^{(p-k)\times(p-k)} \), at least \( k \) eigenvalues of \( DTZD \) are zeros. Since \( k \geq d \), the minimum of the \( d \) largest eigenvalue should be zeroes. The row space and column space of \( DTZD \) is spanned by the row space of \( D \). Namely, the \( d \)-dimensional eigenspace associated with the zero eigenvalue is lies the null space of \( D \). Therefore, the \( \min_{X \in \mathcal{F}^d} h_C(X) \) is zero and we can find the trivial solution when \( k \geq d \) as follow.

Let \( U \in R^{p \times k} \) be a basis of null(\( (k) \)), then the trivial solution \( H = UCA^TUT \), \( C \in R^{k \times k} \). \( A \) is a diagonal matrix with \( \sum_{i=1}^{k} \Lambda_{ii} = d \) and \( \Lambda_{ii} \geq 0 \) for \( i = 1, 2, \ldots, k \). \( C \) is an orthogonal matrix. Hence, \( CAC^T \in \mathcal{F}^d \). In summary, the trivial solution has the form \( H = UAV^T \) with \( A \in \mathcal{F}^d \). This help us to compute the trivial solution.

Consider the problem

\[
\max_{A \in \mathcal{F}^d} \langle U^T SU, A \rangle. \\
\] (A.5)
Let $\hat{A} \in \arg\max_F\langle U^T SU, A \rangle$. We can get the trivial solution by $H = U\hat{A}U^T$.

With the found trivial solution, we can compute the $\lambda_{\text{max}}$. Suppose $H$ is the trivial solution to eq. (A.1) with penalty $\|DXD^T\|_{1,1}$. Then $\mathcal{P}(X) = HXH$ and $\mathcal{P}(X) = HX(I - H) + (I - H)X(I - H) + (I - H)XH$. By the trivial space decomposition of $X$, we have

$$
\langle S, X \rangle - \lambda\|DXD^T\|_{1,1} \\
\leq \left\{ \langle HSH, X \rangle - \lambda\|DHD^T\|_{1,1} \right\} + \left\{ \langle \mathcal{P}^\perp(S), X \rangle - \lambda h_C(\mathcal{P}^\perp(X)) \right\} \\
\leq \langle S, H \rangle + \langle (I - H)S(I - H), X \rangle + \langle HS(I - H), X \rangle \\
+ \langle (I - H)SH, X \rangle - \lambda h_C(\mathcal{P}^\perp(X))
$$

(A.6)

Theses inequalities imply that if the following inequality holds

$$
\langle S, H \rangle - \langle S, X \rangle \geq -\langle (I - H)S(I - H), X \rangle - \langle HS(I - H), X \rangle - \langle (I - H)SH, X \rangle
$$

(A.7)

$h_C(X) = \|DXD^T\|_{1,1}$ indicates that

$$
\lambda\|DXD^T\|_{1,1} = \max_{\|Z\|_{\infty,\infty} \leq \lambda} \langle Z, DXD^T \rangle \\
\geq \langle Z, DXD^T \rangle
$$

(A.8)

for any $\|Z_i\|_{\infty,\infty} \leq \lambda$, $i = 1, 2$. Thus, we let

$$(DD^T)^{-1}D((I - H)S(I - H) + SH + HS)D^T(DD^T)^{-1} = \tilde{Z}
$$

(A.9)

then

$$
-\langle (I - H)S(I - H), X \rangle - \langle HS(I - H), X \rangle - \langle (I - H)SH, X \rangle \\
= -\langle \tilde{Z}, DXD^T \rangle \\
\geq -\max_{\|Z\|_{\infty,\infty} \leq \lambda} \langle Z, DXD^T \rangle \\
= -\lambda\|DXD^T\|_{1,1}.
$$

(A.10)
We can let $\lambda = \| \tilde{Z} \|_{\infty, \infty}$ and this $\lambda$ results in trivial solution of eq. (A.1), because

$$
\langle S, H \rangle - \langle S, X \rangle \\
\geq - \langle (I - H)S(I - H), X \rangle - \langle HS(I - H), X \rangle - \langle (I - H)SH, X \rangle \\
\geq - \lambda \| DXD^T \|_{1,1},
$$

which gives

$$
\langle S, H \rangle - \lambda \| DHD^T \|_{1,1} \geq \langle S, X \rangle - \lambda \| DXD^T \|_{1,1}.
$$

Similarly, we can derive the $\lambda$ resulting trivial solution for penalty $\| DX \|_{1,1}$ by

$$
\lambda = \max\{ \| Z_1 \|_{\infty, \infty}, \| Z_2 \|_{\infty, \infty} \},
$$

where

$$
(DD^T)^{-1}D(I - H)(\frac{S}{2}(I - H) + SH) = \tilde{Z}_1, \quad (A.14)$$

$$
(HS + (I - H)\frac{S}{2})(I - H)D^T(DD^T)^{-1} = \tilde{Z}_2. \quad (A.15)
$$

### A.2 Lasso problem with penalty being squared

We will prove the lemmas that used to derive the algorithm for the lasso problem with penalty being squared. **Proof of Lemma 1.**

**Proof.** Based on the definition of subdifferential, we just need to prove that $v$ satisfies the inequality

$$
\| y \|_1^2 - \| x \|_1^2 - \langle v, y - x \rangle \geq 0
$$

for all $y \in \mathbb{R}^p$. Since $u \in \partial \| x \|_1$, we have

$$
u \in \begin{cases} 
\{1\} & \text{if } x > 0, \\
\{-1\} & \text{if } x < 0, \\
(-1, 1) & \text{if } x = 0.
\end{cases}
$$

68
The definition of subdiffertial implies that \( \|y\|_1 \geq u^T y \) for any \( y \in \mathbb{R}^p \) and the equality holds when \( y = x \). Hence, we can obtain the following inequality

\[
-\langle v, y - x \rangle = -2\|x\|_1 (\langle u, y \rangle - \langle u, x \rangle) \geq -2\|x\|_1 \|y\|_1 + 2\|x\|_2^1
\]

(A.18)

Adding both sides of eq. (A.18) by \( \|y\|_1 - \|x\|_1^2 \), we have

\[
\|y\|_1^2 - \|x\|_1^2 - \langle v, y - x \rangle \geq \|y\|_1^2 - \|x\|_1^2 - 2\|x\|_1\|y\|_1 + 2\|x\|_1^2
\]

\[
= (\|y\|_1^2 - \|x\|_1^2)^2 \geq 0
\]

(A.19)

Therefore, inequality eq. (A.16) holds for all \( y \in \mathbb{R}^p \). Namely, \( v = 2\|x\|_1^2 \in \partial \|x\|_1^2 \in \partial \|x\|_1^2 \).

Proof of Lemma 2.

Proof. Let \( g(\mu) = 2\lambda\|x_{\mu}\|_1 \). \( g(\mu) \) is a continuous and decreasing function of \( \mu \) since \( \|x_{\mu}\|_1 \) decreases as \( \mu \) increases. \( h(\mu) = \mu \) is a continuous and increasing function of \( \mu \). Thus, \( G(\mu) = g(\mu) - h(\mu) \) is continuous and decreasing. \( G(0) = 2\lambda\|y\|_1 > 0 \) and \( G(\|y\|_\infty) = -\|y\|_\infty < 0 \) indicates that there exists a \( \mu \in (0, \|y\|_\infty) \) such that \( G(\mu) = 0 \). Namely, \( \mu = 2\lambda\|x_{\mu}\|_1 \)

A.3 Dynamic Programing for the fused lasso signal approximation (FLSA)

There are two innovative approaches to solve the fused lasso problem. The first approach is derived from the perspective of the submodular functions and the associated Lovász extension. Since the objective function of the fused lasso problem is the Lovász extension of a cut-function, thus maximum-flow/minimum-cut algorithm can be applied to solve the fused lasso. The second approach is introduced by Johnson Johnson (2013). This approach is based on a dynamic programming algorithm. To
the best of my knowledge, it is the fastest algorithm to solve the fused lasso. The proposed dynamic programing based algorithm can solve the fused lasso problem exactly in \( O(N) \) flops. The details of this algorithm are introduced as follow. First, we denote \((\beta_1, \cdots, \beta_p)^T \in \mathbb{R}^p\) by \(\beta_{1:p}\), where \(p = 2, \cdots, N\), the fused lasso problem can be decomposed by

\[
\min_{\beta_N} \left[ \frac{1}{2} (y_N - \beta_N)^2 + \left( \min_{\beta_{1:N-1}} \frac{1}{2} \sum_{k=1}^{N-1} (y_k - \beta_k)^2 + \lambda \sum_{k=2}^N |\beta_k - \beta_{k-1}| \right) \right]. \tag{A.20}
\]

Then, we define the second term in eq. (A.20) as a univariate function \(f_N\) of the variable \(\beta_N\)

\[
f_N(\beta_N) = \left[ \min_{\beta_{1:N-1}} \frac{1}{2} \sum_{k=1}^{N-1} (y_k - \beta_k)^2 + \lambda \sum_{k=2}^N |\beta_k - \beta_{k-1}| \right]
\]

\[
= \min_{\beta_{N-1}} \left[ \frac{1}{2} (y_{N-1} - \beta_{N-1})^2 + \lambda |\beta_N - \beta_{N-1}| \right] + \min_{\beta_{2:N-2}} \left[ \frac{1}{2} \sum_{k=1}^{N-2} (y_k - \beta_k)^2 + \lambda \sum_{k=2}^{N-1} |\beta_k - \beta_{k-1}| \right]
\]

\[
= \min_{\beta_{N-1}} \left[ \frac{1}{2} (y_{N-1} - \beta_{N-1})^2 + \lambda |\beta_N - \beta_{N-1}| + f_{N-1}(\beta_{N-1}) \right]. \tag{A.21}
\]

According to the definition of \(f_N(\cdot)\) in eq. (A.21), we can define \(f_{N-1}(\cdot), \cdots, f_2(\cdot)\) recursively. And for a given \(\beta_N\), the value of \(\beta_{N-1}\) is determined by a minimization problem based on the value of \(\beta_N\). Sequentially, we can see that the value of \(\beta_{p-1}\) is determined the value of \(\beta_p\) for \(p = N, N - 1, \cdots, 2\) through a minimization problem. Therefore, the fused lasso problem can be viewed as a univariate minimization problem of \(\beta_N\),

\[
\min_{\beta_N} \frac{1}{2} (y_N - \beta_N)^2 + f_N(\beta_N) \quad \tag{A.22}
\]
Once we find the solution \( \hat{\beta}_N \), we can determine the value of \( \hat{\beta}_{N-1}, \cdots, \hat{\beta}_1 \) based on eq. (A.21). According to this idea, we first define \( \delta_k \) for \( k \geq 2 \) as follow:

\[
\begin{align*}
\delta_1(b) &= \frac{1}{2}(y_1 - b)^2 \\
\delta_k(b) &= \frac{1}{2}(y_k - b)^2 + f_{k-1}(b) \\
\delta_{k-1}(b) &= \frac{1}{2}(y_{k-1} - b)^2 + f_{k-1}(b) \\
\delta_{N-1}(b) &= \frac{1}{2}(y_{N-1} - b)^2 + f_{N-1}(b) \\
\delta_N(b) &= \frac{1}{2}(y_N - b)^2 + f_N(b)
\end{align*}
\]

(A.23)

Each \( f_k(b) \) is a minimization problem of variable \( \tilde{b} \). And the minimizer only depends on the value of \( b \). We can define a sequence of functions by

\[
\psi_k(\beta_k) = \arg\min_b \{ \delta_{k-1}(\tilde{b}) + \lambda |\beta_k - \tilde{b}| \} \quad (\text{A.24})
\]

for \( k = 2, \cdots, N \). And the function \( \psi_k(\cdot) \) provides the relationship between \( \hat{\beta}_k \) and \( \hat{\beta}_{k-1} \). If we can find \( \psi_k(\cdot) \) for \( k = 2, \cdots, N \) and \( \hat{\beta}_N \), we can find the solution to the fused lasso problem. According to the definition of \( \delta_k \), we can see that \( \delta_k(\cdot) \) is convex, differentiable and piecewise quadratic. Thus, \( \delta'_k(\cdot) \) is non-decreasing, continuous and piecewise linear. \( \psi_k(b) \) has the form \((b \land b^+_k) \lor b^-_k\) where \( \delta'_k(b^-_k) = -\lambda \) and \( \delta'_k(b^+_k) = \lambda \).

The details of the algorithm can be summarized as follow. For \( k = 1, \cdots, N - 1 \)

- We search from left to right to get \( b^-_k \) such that \( \delta'_k(b^-_k) = -\lambda \).
• We search from right to left to get $b_k^+$ such that $\delta_k'(b_k^+) = \lambda$

• $f'_{k+1}(b)$ is obtained by $f'_{k+1}(b) = (\delta_k'(b) \lor -\lambda) \land \lambda$

• $\delta_{k+1}'(b)$ is obtained by $f'_{k+1}(b) + b - y_{k+1}$

Once we get all $\delta_k'(b)$'s, $b_k^+$'s and $b_k^-$'s, we can first compute $\hat{\beta}_N$ and then obtain the optimal sequence $\hat{\beta}_{N-1}, \ldots, \hat{\beta}_1$ through $\psi_N, \ldots, \psi_2$.

The intuition behind this algorithm is the fused lasso penalty has a chain structure of variables $\beta$. Therefore, the fused lasso can be written as a sequence of univariate problems as in eq. (A.23). However, for penalty $\| D^{(k)} Y D^{(k)T} \|_{1,1}$ in problem eq. (3.12b) does not have a such chain structure. If we following the same scheme to decompose problem eq. (3.12b), we can have

$$\delta_1(b_{11}) = \frac{1}{2}(y_{11} - b)^2$$

$$f_2(b_{12}, b_{21}, b_{22}) = \min_b \left[ \delta_1(\bar{b}) + \lambda |b_{22} + \bar{b} - b_{21} - b_{12}| \right]$$

$$\delta_2(b_{22}, b_{12}, b_{21}) = \frac{1}{2} [ (y_{22} - b_{22})^2 + (y_{12} - b_{12})^2 + (y_{21} - b_{21})^2 ] + f_2(b_{12}, b_{21}, b_{22})$$

$$\ldots$$

(A.25)

For $k > 2$, $f_k$ is a minimization problem with multiple variables. Thus, the minimization problem $f_k$ does not have a nice explicit form. Therefore, dynamic programming is not applicable to eq. (3.12b).
A.4 Primal-Dual Interior Point method for the Trend Filtering problem

In this section, we derive a primal-dual interior point method to solve the trend filtering problem. The trend filtering problem has the form of

\[
\text{minimize} \frac{1}{2} \| y - x \|^2_2 + \lambda \| Dx \|_1.
\] (A.26)

Eq. (A.26) has an equivalent form:

\[
\text{minimize} \frac{1}{2} \| y - x \|^2_2 + \lambda \| z \|_1
\]

subject to \( Dx = z \) (A.27)

To solve this problem, we introduce a dual variable \( v \in \mathbb{R}^{(p-\text{diff order})} \) and the Lagrangian is

\[
L(x, z, v) = \frac{1}{2} \| y - x \|^2_2 + \lambda \| z \|_1 + v^T(Dx - z).
\] (A.28)

The dual function is

\[
\inf_{x, z} L(x, z, v) = \begin{cases} 
-\frac{1}{2} v^T DD^T v + y^T D^T v, & -\lambda 1 \leq v \leq \lambda 1 \\
-\infty, & \text{otherwise}
\end{cases}
\] (A.29)

\(1 \in \mathbb{R}^{(p-\text{diff order})}\) is a constant vector with all entries are equal to 1 and \(-\lambda 1 \leq v \leq \lambda 1\) is equivalent to \(\|v\|_\infty \leq \lambda\). The dual problem of eq. (A.26) is

\[
\text{minimize}_v g(v) = \frac{1}{2} v^T DD^T v - y^T D^T v
\]

subject to \(-\lambda 1 \leq v \leq \lambda 1\), (A.30)

which is a least square problem with a box constraint. The solution \(\hat{v}\) of the dual and the solution \(\hat{x}\) of the primal are connected via:

\[
\hat{x} = y - D^T \hat{v}
\] (A.31)
A.4.1 Barrier method for convex problem with inequality constraints

Before we discuss the details of the primal-dual interior point method for trend filtering, we first go through the barrier method. Consider problem:

minimize $f_0(x)$
subject to $f_i(x) < 0, \ i = 1, 2, \ldots, m.$ \hspace{1cm} (A.32)

Define logarithmic barrier function $\phi$ as $\phi(x) = \begin{cases} -\sum_{i=1}^{m} \log(-f_i(x)) & f_i(x) < 0 \\ +\infty & otherwise \end{cases}.$

So the original problem eq. (A.32) can be approximated by

$$\min \ tf_0(x) + \phi(x)$$ \hspace{1cm} (A.33)

As $t$ goes to infinity, solution of eq. (A.33) will be close to the solution of eq. (A.32).

So the idea of Barrier method is to solve a sequence of eq. (A.33) as $t$ increases. The solution of each iterate will be treat as starting point in the next iterate.

Newton’s method is used to solve eq. (A.33), the update direction $\delta v$ are given by solving:

$$(t \nabla^2 f_0(v) + \nabla^2 \phi(v))\Delta v = t \nabla f_0(v) + \nabla \phi(v)$$ \hspace{1cm} (A.34)

Solving above equality is equivalent to solving the modified KKT condition of eq. (A.32):

$$\nabla f_0(v) + \sum_{i=1}^{m} \mu_i \nabla f_i(v) = 0$$ \hspace{1cm} (A.35)

$$-\mu_i f_i(v) = 1/t, i = 1, 2, \ldots, m$$ \hspace{1cm} (A.36)

eliminating $\mu_i$ to solve modified KKT condition and solving eq. (A.33) directly are equivalent.
A.4.2 PDIP method

The difference between barrier method and interior point method is that interior point method solves the modified KKT condition directly without eliminating $\mu_i$. In the dual of eq. (A.26), $f_0(v) = \frac{1}{2}v^TDD^Tv - y^TD^Tv$, $f_i(v)'s$ are $-\lambda - v < 0$ and $v - \lambda < 0$. Introducing two dual variables $\mu_1 \in \mathbb{R}^{(p-2)^2}$ and $\mu_2 \in \mathbb{R}^{(p-2)^2}$, the modified KKT condition is:

$$DD^Tv - Dv + \mu_1(v - \lambda 1) - \mu_2(v + \lambda 1) = 0 \quad (A.37)$$

$$-\mu_1(v - \lambda 1) - \frac{1}{t}1 = 0 \quad (A.38)$$

$$\mu_2(v + \lambda 1) - \frac{1}{t}1 = 0 \quad (A.39)$$

Apply Newton’s method to above linear system, we need to solve the following in each iterate to find out searching direction

$$\begin{bmatrix} \Delta v \\ \Delta u_1 \\ \Delta u_2 \end{bmatrix} = \begin{bmatrix} DD^T - J_1^{-1}J_2^{-1} \\ \frac{1}{t}diag(f_1)^{-1}1 \\ \frac{1}{t}diag(f_2)^{-1}1 \end{bmatrix}$$

where

$$f_1 = v - \lambda 1 \in \mathbb{R}^{(n-2)\times(n-2)}$$

$$f_2 = v - \lambda 1 \in \mathbb{R}^{(n-2)\times(n-2)}$$

$$J_i = diag(u_i)^{-1}diag(f_i) \in \mathbb{R}^{(n-2)\times(n-2)}$$

eliminating $\Delta u_1$ and $\Delta u_2$ we can get the linear system:

$$(DD^T - J_1^{-1}J_2^{-1}) \Delta v = -(DD^Tv - D(x + u) - \frac{1}{t}diag(f_1)^{-1}1 + \frac{1}{t}diag(f_2)^{-1}1)$$

$$\quad (A.41)$$
Solving the above linear system, we can get $\Delta \mathbf{v}$. Then $\Delta \mathbf{u}_1$ and $\Delta \mathbf{u}_2$ can be computed by:

$$
\Delta \mathbf{u}_1 = -(\mathbf{u}_1 + \frac{1}{t} \text{diag}(f_1)^{-1} \mathbf{1} + \mathbf{J}_1^{-1} \text{vec}(\Delta \mathbf{v})) \quad (A.42)
$$

$$
\Delta \mathbf{u}_2 = -(\mathbf{u}_2 + \frac{1}{t} \text{diag}(f_2)^{-1} \mathbf{1} + \mathbf{J}_2^{-1} \text{vec}(\Delta \mathbf{v})) \quad (A.43)
$$

After determining the searching direction, the step size can be obtained by backtracking line search. So the majority of complexity is due to eq. (A.41). Since $(DD^T - \mathbf{J}_1^{-1} \mathbf{J}_2^{-1})$ is a banded matrix, we can use LU decomposition to solve it.

### A.4.3 Solve banded linear system by LU decomposition

In this section, we describe the how to exploit the banded structure of $I + \rho_0 \mathbf{D}^T \mathbf{D} \otimes \mathbf{D}^T \mathbf{D}$.

Every nonsingular matrix $A \in \mathbb{R}^{n \times n}$ can be factored as $A = PLU$, where $P \in \mathbb{R}^{n \times n}$ is a permutation matrix, $L \in \mathbb{R}^{n \times n}$ is unit lower triangular, and $U \in \mathbb{R}^{n \times n}$ is upper triangular and nonsingular. This is called the LU factorization of $A$. The standard algorithm for computing an LU factorization is called Gaussian elimination with partial pivoting or Gaussian elimination with row pivoting. The cost is $(2/3)n^3$ flops if no structure in $A$ is exploited.

Then consider $A$ is a banded matrix of size $n \times n$, $a_{ij} = 0$ if $||i - j|| > k$, where $k$ is much smaller than $n$ is called the bandwidth of $A$. In this case an LU factorization of $A$ can be computed in roughly $4nk^2$ flops: The resulting upper triangular matrix $U$ has bandwidth at most $2k$, and the lower triangular matrix $L$ has at most $k + 1$ nonzeros per column, so the forward and back substitutions can be carried out in order $6nk$ flops. Therefore if $A$ is banded, the linear equations $Ax = b$ can be solved
Algorithm 2 Solving banded linear equations by LU factorization

Given a set of linear equations $Ax = b$, with $A$ nonsingular and banded.

LU factorization. Factor $A$ as $A = PLU (4nk^2$ flops)  
Permutation. Solve $Pz_1 = b$ (0 flops)  
Forward substitution. Solve $Lz_2 = z_1$ ($(k + 1)^2$ flops)  
Backward substitution. Solve $Ux = z_2$ ($(2k)^2$ flops)

The total cost is $4nk^2 + (k + 1)^2 + (2k)^2$ or roughly $4nk^2$ flops if we keep only the leading term. Therefore, we can exploit the banded structure of matrix $DD^T$ and implement the primal-dual interior point methods efficiently.

A.5 Specialized ADMM for the Trend Filtering problem

In this section, we denote the $D^{(k)}$ to be the $k$-th order difference matrix. ADMM algorithm can be used to solve the trend filtering problem. By introducing the auxiliary variable $z \in \mathbb{R}^{p-\text{difforder}}$ and equality constraint $Dx = z$, we obtain the augmented Lagrangian of the trend filtering problem

$$
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \| y - x \|^2_2 + \lambda \| z \|_1 \\
\text{subject to} & \quad D^{(k)}x = z
\end{align*}
$$

(A.44)

The augmented Lagrangian can be written as

$$
L(x, z, u) = \frac{1}{2} \| y - x \|^2_2 + \lambda \| z \|_1 + \frac{1}{2} \| D^{(k)}x - z + u \|^2_2 - \frac{1}{2} \| u \|^2_2,
$$

(A.45)
from which the standard ADMM algorithm can be derived

\[
x \leftarrow (I + \rho(D^{(k)})^T D^{(k)})^{-1}(y + \rho(D^{(k)})^T(z + u)), \quad (A.46a)
\]

\[
z \leftarrow S_{\lambda/\rho}(D^{(k)}x - u), \quad (A.46b)
\]

\[
u \leftarrow u + z - D^{(k)}x \quad (A.46c)
\]

The \(x\)-update is a banded linear system. The \(z\)-update is a coordinate-wise soft-thresholding at the level \(\lambda/\rho\). The specialized ADMM begins by rewrite the trend filtering problem as

\[
\text{minimize} \frac{1}{2}\|y - x\|^2_2 + \lambda\|D^{(1)}z\|_1 \quad (A.47)
\]

subject to \(D^{(k-1)}x = z\)

where \(D^{(k-1)}D^{(1)} = D^{(k)}\). The augmented Lagrangian becomes

\[
L(x, z, u) = \frac{1}{2}\|y - x\|^2_2 + \lambda\|D^{(1)}z\|_1 + \frac{1}{2}\|D^{(k-1)}x - z + u\|^2_2 - \frac{1}{2}\|u\|^2_2, \quad (A.48)
\]

which yields the specialized ADMM

\[
x \leftarrow (I + \rho(D^{(k-1)})^T D^{(k-1)})^{-1}(y + \rho(D^{(k-1)})^T(z + u)), \quad (A.49a)
\]

\[
z \leftarrow \text{arg min}_z (D^{(k-1)}x - u - z) + \lambda/\rho\|D^{(1)}uz\|, \quad (A.49b)
\]

\[
u \leftarrow u + z - D^{(k-1)}x. \quad (A.49c)
\]

The \(x\)-update and \(z\)-update are analogous to those of the standard ADMM. But specialized ADMM has smaller difference order. And the \(z\)-update becomes a fused lasso signal approximator (FLSA). The specialized ADMM converges faster than the standard version. The intuition behind this is that specialized ADMM utilizes a dynamic programming subroutine to solve a more difficult problem with same among of time. Thus, the algorithm makes more progress to minimize the objective. In order
to have a more concrete understanding of the superiority of the specialized ADMM over the standard version, we invert $D^{(k)}$ and get $H^{(k)}$ which is the falling factorial basis matrix. For the standard ADMM, we reparameterize the trend filtering in an equivalent lasso forms of trend filtering.

$$\min_{x,z} \frac{1}{2} \|y - H^{(k)}x\|_2^2 + \lambda k! \sum_{j=k+2}^p |x_j|,$$

(A.50a)

s.t. $x = z.$

(A.50b)

For the specialized ADMM, we have

$$\min_{x,z} \frac{1}{2} \|y - H^{(k-1)}x\|_2^2 + \lambda k! \sum_{j=k+2}^p |x_j|,$$

(A.51a)

s.t. $Lx = z.$

(A.51b)

where we have the recursion $H^{(k)} = LH^{(k-1)}$, which is analogous to $D^{(k)} = D^{(1)}D^{(k-1)}$.

If we ignore the dual update, we can rewrite the ADMM iterates of the algorithm as a joint minimization problem. The standard ADMM has the form of

$$\min_{x,z} \frac{1}{2} \|y - [H^{(k)} \sqrt{\rho}I - \sqrt{\rho}L][x, z]\|_2^2 + \lambda k! \sum_{j=k+2}^p |z_j|$$

(A.52)

The specialized ADMM has the form of

$$\min_{x,z} \frac{1}{2} \|y - [H^{(k-1)} \sqrt{\rho}I - \sqrt{\rho}L][x, z]\|_2^2 + \lambda k! \sum_{j=k+2}^p |z_j|$$

(A.53)


Ryan J Tibshirani and Jonathan Taylor. The solution path of the generalized lasso.  


