A COMPREHENSIVE APPROACH FOR SPARSE PRINCIPLE COMPONENT ANALYSIS USING REGULARIZED SINGULAR VALUE DECOMPOSITION

A Dissertation

by

SENMAO LIU

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Jianhua Huang
Committee Members,	Valen Johnson
	Bani Mallick
	Yu Ding
Head of Department,	Valen Johnson

August 2016

Major Subject: Statistics

ABSTRACT

Principle component analysis (PCA) has been a widely used tool for statistics and data analysis for many years. A good result of PCA should be both interpretable and accurate. However, neither interpretability nor accuracy could be achieved well in "big data" scenarios where there are large numbers of original variables. Therefore people developed sparse PCA, in which obtained principle components (PCs) are linear combinations of a limited number of original variables, which yields good interpretability. In addition, some theoretical results showed that, when the genuine model is sparse, PCs obtained via sparse PCA instead of traditional PCA are consistent estimators. These aspects have made sparse PCA a hot research topic in recent years.

In this dissertation, we developed a comprehensive and systematic way for doing sparse PCA by using an SVD-based approach. In detail, we proposed the formulation and algorithm and showed its consistency and convergence. We even showed convergence to global optima using a limited number of trials, which is a breakthrough in sparse PCA area. In addition, to guarantee orthogonality or uncorrelatedness when multiple PCs are extracted, we developed a method for sparse PCA with orthogonal constraint, proposed its algorithm, and showed the convergence. In addition, to deal with missing values in the design matrix which often happens in reality, we developed a method for sparse PCA with missing values, proposed its algorithm, and showed the convergence. Moreover, to provide a good way of selecting tuning parameter in these formulations, we designed an entry-wise cross validation method based on sparse PCA with missing values. All these contributions and breakthroughs make our results practically useful and theoretically complete. Simulation study and real-

world data analysis are also provided, which showed that our method has competing results with others in "without missing" cases, and good results in "with missing" cases in which currently we are the only practical method.

DEDICATION

This dissertation is gratefully dedicated to my parents, thank you for many years' education and support. This dissertation is also dedicated to my sister, my girlfriend, and all my dear friends, thank you for companion during my childhood, my teenage, and my college life. This dissertation is also dedicated to my supervisor, thank you for teaching me research principles and good practical suggestions.

ACKNOWLEDGEMENTS

Though only my name appears on the cover of this dissertation, a great many people have contributed to its production. I owe my gratitude to all those people who have made this dissertation possible and because of whom my graduate experience has been one that I will cherish forever.

My deepest gratitude is to my advisor, Dr. Jianhua Huang. I have been amazingly fortunate to have an advisor who gave me the freedom to explore on my own, and at the same time the guidance to recover when my steps faltered. Dr. Huang taught me how to question thoughts and express ideas. His patience and support helped me overcome many crisis situations and finish this dissertation. I hope that one day I would become as good a group leader to my group members as Dr. Huang has been to me.

I also appreciate a lot for my other committee members, Dr. Valen Johnson, Dr. Xiaoning Qian, Dr. Yu Ding, and Dr. Bani Mallick, for their insightful comments and constructive criticisms at different stages of my research. I am grateful to them for holding me to a high research standard and enforcing strict validations for each research result, and thus teaching me how to do research. Dr. Johnson gave me some useful suggestions from a Bayesian perspective. Dr. Qian also gave me many enlightening ideas from his research.

Dr. Zhaosong Lu had worked with me for one semester, he taught me a lot of coding skills, working with him is really a good experience.

Dr. Micheal Longnecker is the assistant head in our department. He is always glad to help us graduate students. I gained a lot from his useful advices to design my career. I also benefitted a lot from his teaching of experimental design course.

Dr. Mohsen Pourahmadi is one of the best teachers that I have had in my life. He sets high standards for his students and he encourages and guides them to meet those standards. He introduced me to Linear Modeling and his teachings inspired me to design the matrix-based techniques which is useful for this dissertation. I am indebted to him for his continuous encouragement and guidance.

I am grateful to Dr. Samiran Sinha and Dr. Suhasini Subba Rao for their teaching. They are the kind of instructors whom teaching with enthusiasm. They always make the class "full of knowledge and exploration". I learned a lot of statistical ideas and methods form them.

I am also indebted to the members of the whole research group, with whom I have interacted during the course of my graduate studies. Particularly, I would like to acknowledge Dr. Mehdi Maadooliat, Dr. Ganggang Xu, Dr. Bohai Zhang, Dr. Yuan Qu, Dr. Shuo Feng, Dr. Nan Zhang, Kejun He, Shiyuan He, Ya Su, and Yei Eun Shin, for the many valuable discussions that helped me understand my research area better.

My sincere thanks to Meng Lu, Xuefeng Cui, and Bo Liu for their contributions to the various domains.

Most importantly, none of this would have been possible without the love and patience of my family. My family to whom this dissertation is dedicated to, has been a constant source of love, concern, support and strength all these years. I would like to express my heart-felt gratitude to them.

NOMENCLATURE

PCA	Principle Component Analysis
SVD	Singular Value Decomposition
ADMM	Alternating Direction Method of Multipliers
QP	Quadratic Programming

TABLE OF CONTENTS

Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION: SPARSE PRINCIPLE COMPONENT ANALYSIS AND REGULARIZED SINGULAR VALUE DECOMPOSITION

1.1 Principle Component Analysis and Its Properties

Principle component analysis (PCA, or standard PCA, to be different from sparse PCA) has been a popular feature extraction and dimension reduction tool for several decades. PCA seeks the linear combinations of the original variables, the obtained variables are called principle components (PCs). The criterion of extraction is to capture maximal variance of the original data matrix and therefore can guarantee minimal information loss. Thus PCA usually can be obtained via either maximizing variance or minimizing reconstruction error.

Suppose $\mathbf{X}$ is an $n \times p$ data matrix, then the first k PCs can obtained via the following optimization problem

$$\max_{\mathbf{V}:p\times k, \mathbf{V}^T\mathbf{V}=\mathbf{I}} Tr(\mathbf{V}^T\mathbf{X}^T\mathbf{X}\mathbf{V}) = \sum_{i=1}^{k} \mathbf{v}_i^T\mathbf{X}^T\mathbf{X}\mathbf{v}_i, \tag{1.1}$$

where $\mathbf{V} = (\mathbf{v}_1, ..., \mathbf{v}_k)$ are the first k PCs.

As we know, this is a typical eigen-decomposition problem for variance-covariance matrix $\mathbf{X}^T\mathbf{X}$, therefore it has all mathematical properties of eigen-decomposition, which further makes PCs have corresponding statistical properties.

The first property of standard PCA is that the simultaneous way described in (1.1) is equivalent to the sequential way. The sequential way is that first we extract the leading PC via the following formulation

$$\max_{\mathbf{v}:p\times 1, \mathbf{v}^T\mathbf{v}=1} \mathbf{v}^T\mathbf{X}^T\mathbf{X}\mathbf{v}, \tag{1.2}$$

then we use the deflation method to update data matrix $\mathbf{X}$ via

$$\mathbf{X} \leftarrow \mathbf{X} - \mathbf{X}\mathbf{v}\mathbf{v}^T, \tag{1.3}$$

and then we solve the optimization problem (1.2) again using the updated data matrix. To extract k PCs, we repeat this procedure k times.

The second property is that the obtained PCs are both geometrically orthogonal and statistically uncorrelated with each other, for $i \neq j$,

$$\mathbf{v}_i \perp \mathbf{v}_j, \qquad \mathbf{X}\mathbf{v}_i \perp \mathbf{X}\mathbf{v}_j. \tag{1.4}$$

The third property is the convergence of the optimization problems. Either using the simultaneous way or using the sequential way, we are solving an eigen-decomposition problem, and typically the power iteration can solve it efficiently and if we can make a correct initialization (in the sense that it is not orthogonal to the truth, and we know this is not difficult if you repeat for several times) we can get the global optima.

1.2 Sparse Principle Component Analysis

PCA has made great power in traditional data analysis, e.g., it is an important exploratory data analysis technique, it can be used for feature extraction and dimension reduction, and it can be used in principle component regression to solve the multi-colinearity problem. However, the rapid development of data science has proposed great challenge on this method. In modern times, "Big Data" has been a popular topic, in which the data set usually has thousands of or even millions of variables. In this case, for each PC, the loadings are typically nonzero, which makes it a linear combination of thousands original variables, and this is quite difficult to

explain and almost impossible to identify important variables. In addition, theoretically, the obtained PCs are inconsistent estimates for the true PCs (see [15], [14] and [7]).

To address the drawbacks of standard PCA, various modified PCA methods have been proposed to form PCs where each PC is the linear combination of a small subset of the original variables and can still explain high percentage of variance. All these results can be called Sparse Principle Component (sparse PCA). The corresponding literature review is introduced in next section. In this section, we talk about the difficulty for sparse PCA.

As we mentioned in previous section, standard PCA is an eigen-decomposition problem, sparse PCA can be considered as a perturbation from this eigen-structure, therefore it does not have those good properties from eigen-decomposition.

The first problem is that the simultaneous way is not equivalent to the sequential way any more. In the literature review, we could see that some papers use the simultaneous way while others use the sequential way, and it is difficult to find their relationship or equivalence.

The second one is the orthogonality and uncorrelatedness can not be guaranteed naturally. Some papers just ignore this, while others try to put orthogonal constraint on the optimization problem, which could obtain orthogonality yet make the optimization more complicated.

The third one is that it's difficult to obtain the convergence to global optima, especially when the formulation used in many papers is not a convex optimization problem.

1.3 Singular Value Decomposition and Regularization

Singular Value Decomposition (SVD) is another powerful tool both in mathematical and statistical science. As we know, SVD is also an eigen-decomposition problem, therefore SVD is equivalent to PCA, in more details, the left singular vector of data matrix $\mathbf{X}$ is equivalent to PC loadings of variance-covariance matrix $\mathbf{X}^T\mathbf{X}$. Therefore people also use SVD to obtain PCs. SVD also has those mathematical properties, (1) the simultaneous way is equivalent to the sequential way; (2) orthogonality can be guaranteed; (3) convergence (even to global optima) can be guaranteed.

For data matrix $\mathbf{X}$, its singular value decomposition can be obtained simultaneously via the following optimization problem

$$\max_{\mathbf{V},\mathbf{U}:\mathbf{V}^T\mathbf{V}=\mathbf{I},\mathbf{U}^T\mathbf{U}=\mathbf{I}} ||\mathbf{X}-\mathbf{U}\mathbf{V}^T||_2^2, \tag{1.5}$$

or sequentially via the following optimization problem

$$\max_{\mathbf{v},\mathbf{u}} ||\mathbf{X}-\mathbf{u}\mathbf{v}^T||_2^2, \tag{1.6}$$

and the following deflation

$$\mathbf{X} \leftarrow \mathbf{X}-\mathbf{u}\mathbf{v}^T. \tag{1.7}$$

Similar as PCA, the traditional SVD also has some problem when dealing with modern data sets. For some data analysis, we require either sparse (or smooth) left (or right) singular vectors. For example in fMRI data analysis, the left singular vectors corresponds to temporal domain therefore is required to be smooth (i.e., to be continuous along with time), the right singular vector corresponds to spatial domain

therefore is required to be sparse (i.e., a sparse active region is expected). To deal with this, people designed regularized SVD (see [5] and [4]).

The main topic of our thesis is to use regularized SVD to obtain sparse PCA for the data matrix. We make a thorough investigation on all kinds of regularized SVD and build a systematic way of proposing regularization terms for SVD problem given different requirements. Especially when regularized SVD is for sparse PCA, we build a complete approach, which includes convergence, consistency, orthogonal constraint, missing values, cross validation, and convergence to global optima, which make our approach to be the most comprehensive. A detailed comparison could be seen in section of literature review.

2. LITERATURE REVIEW: A DETAILED COMPARISON ON DIFFERENT SPARSE PCA METHODS

2.1 Performance Table

We found tens of papers for sparse PCA and related issue. To make a systematic comparison, before the literature review, we build a list of measurements to help compare the performance of sparse PCA methods.

2.1.1 Separate-Processing/Regularization/Iteration

This aspect is about the basic strategy used to obtain sparse loading vectors for PCA. Some papers performed a separate processing before or after the standard PCA to achieve sparsity (e.g., [6]). Many other papers used the regularization strategy, which means a regularization term is added to the model-fitting term to form a formulation, which is then solved to obtain sparse PCs. Some others just proposed an algorithm based on iterations between multiplication by $\mathbf{X}$ (or $\mathbf{X}^T$) and vector filtering, without a formulation for their method, and we call this way iteration.

2.1.2 Simultaneous-Extraction/Sequential-Extraction

As we mentioned in introduction section, both PCA and SVD can be obtained by either the sequential way or the simultaneous way. The sequential way is that we extract the leading layer from data matrix $\mathbf{X}$ first, then do deflation to update $\mathbf{X}$ and continue extraction. The simultaneous way is that we extract all necessary layers simultaneously via some matrix-based formulation. We know that these two methods are the same for traditional PCA, however for sparse PCA, they are not equivalent any more.

Which strategy is better is still an open question. On one hand, if the extraction

of leading layers are far from the truth, then the deflation is greatly affected by leading layer and the extraction of following layers becomes less convincible. In this aspect, we may conclude that the simultaneous way is more robust. On the other hand, one should do tuning parameter selection for k parameters simultaneously for the simultaneous way, whose repetition number is m^k if we use m equally spaced candidate values for each tuning parameter. As a comparison, the sequential way only needs $m \times k$ repetitions.

2.1.3 Convergence

Some papers not just proposed their method, but also showed that the algorithm provided converges to a stationary point (local optima), in some cases even to a global optima. Many other papers failed to show this. A method that can not provide this guarantee may make the algorithm failing to stop or converging to an incorrect solution.

2.1.4 Consistency

In addition to algorithmic convergence, people are also interested in the convergence in probabilistic aspect, i.e., the (statistical) consistency property. Some methods could make corresponding consistency guarantee, while others couldn't. Besides, to prove consistency, different models are used. Specifically, people usually used the spike model to show the consistency for SVD-based methods (e.g., see [16]).

2.1.5 Tuning Parameter Selection

There usually exists a tuning parameter (even more than one) for sparsity-induced penalty in the formulation, whose goal is to make balance between model-fitting and required property (sparsity). So proper value for tuning parameter should be chosen. Some papers used cross validation to do the selection, while others did not consider

this part, which is practically not complete.

2.1.6 Orthogonality

As mentioned in comparison between the sequential and the simultaneous extraction, for traditional PCA, different layers are orthogonal and we could get orthogonality from either the sequential way or the simultaneous way. However in sparse PCA, this may not be true. To achieve orthogonality, some papers used a post-processing after their algorithm, some others added an orthogonal constraint to the formulation, while many other papers just ignored this aspect.

2.1.7 Missing Values

In reality, the data matrix is not always complete, and there could be kinds of reasons that cause missing values in the data matrix. Therefore, a practically complete method should be able to do sparse PCA with missing values. This is one of the main contribution for our method, given that only few papers mentioned how to do sparse PCA with missing values. In addition, if we have a method of sparse PCA with missing values, then we could develop the cross-validation for tuning parameter selection, by leaving some entries of data matrix out, and splitting the whole matrix as training set and validation set.

2.2 Descriptions and Summaries for Reviewed Papers

In the following we make a brief introduction to the papers that we mainly investigated (mainly in temporal order). We first talk about the methods used, show the formulation and then make a summary on the performance as listed above (shown in a table).

2.2.1 *Jolliffe, Trendafilov, and Uddin [8]*

This paper proposed a constraint optimization problem:

$$\max \mathbf{a}_k^T \mathbf{R} \mathbf{a}_k, \text{ sub to } \mathbf{a}_k^T \mathbf{a}_k = 1, \mathbf{a}_h^T \mathbf{a}_k = 0 (h < k), ||\mathbf{a}_k||_1 \leq t, \tag{2.1}$$

where $\mathbf{R}$ is the sample correlation (or covariance) matrix and $\mathbf{a}_k' s$ are PC loading vectors. Then they used a gradient-based method to solve the problem (2.1). We can see this is the sequential way of doing extraction. The convergence and consistency results are not provided, while orthogonality is guaranteed. Tuning parameter selection and missing value are not considered neither. A summary result could be seen in table (2.1).

Performance List	Results and Comments
Regularization/Pre/Post	Regularization
Sequential/Simultaneous	Simultaneous
Convergence	No
Consistency	No
Tuning-Parameter Selection	Not Provided
Orthogonality	Yes
Missing-Values	No

Table 2.1: Performance summary for Jolliffe, Trendafilov, and Uddin [8]

2.2.2 *Johnstone and Lu [6]*

In this paper, the authors used a separate-processing strategy, in the very beginning they used a wavelet-based algorithm for selecting a subset of coordinates with largest sample variances, and they showed that if PCA is done on the selected subset, the consistency is recovered, even if p is much larger than n. They used 1-s.d. rule

to select tuning parameter, they also suggested using median absolute difference to estimate standard deviation. A summary result could be seen in table (2.2).

Performance List	Results and Comments
Regularization/Pre/Post	Pre-Processing
Sequential/Simultaneous	Simultaneous
Convergence	Yes (Standard PCA)
Consistency	Yes
Tuning-Parameter Selection	1-sd Rule
Orthogonality	Yes (Standard PCA)
Missing-Values	No

Table 2.2: Performance summary for Johnstone and Lu [6]

2.2.3 Zou, Hastie, and Tibshirani [26]

In this paper, the authors transformed the PCA problem into a regression problem:

$$\min_{\mathbf{A},\mathbf{B}} \sum_{i=1}^{n} ||\mathbf{x}_i - \mathbf{A}\mathbf{B}^T\mathbf{x}_i||_2^2 + \lambda \sum_{j=1}^{k} ||\boldsymbol{\beta}_j||_2^2, \text{ sub to } \mathbf{A}^T\mathbf{A} = \mathbf{I}, \tag{2.2}$$

where $\mathbf{B} = [\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_k]$ and $\mathbf{x}_i$ is i-th sample of data matrix $\mathbf{X} = (\mathbf{x}_1^T, \ldots, \mathbf{x}_n^T)^T$.

Then they added an elastic-net penalty on the regression problem to encourage sparsity:

$$\min_{\mathbf{A},\mathbf{B}} \sum_{i=1}^{n} ||\mathbf{x}_i - \mathbf{A}\mathbf{B}^T\mathbf{x}_i||_2^2 + \lambda \sum_{j=1}^{k} ||\boldsymbol{\beta}_j||_2^2 + \sum_{j=1}^{k} \lambda_{1,j}||\boldsymbol{\beta}_j||_1, \text{ sub to } \mathbf{A}^T\mathbf{A} = \mathbf{I}, \tag{2.3}$$

where $\mathbf{B} = [\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_k]$.

The authors used alternating-direction algorithm to solve the above optimization problem. More details can be seen in table (2.3).

Performance List	Results and Comments
Regularization/Pre/Post	Regularization
Sequential/Simultaneous	Simultaneous
Convergence	No
Consistency	No
Tuning-Parameter Selection	Not Provided
Orthogonality	No
Missing-Values	No

Table 2.3: Performance summary for Zou, Hastie, and Tibshirani [26]

2.2.4 Shen and Huang [20]

This paper also used an SVD-based method. Their formulation is

$$||\mathbf{X} - \tilde{\mathbf{u}}\tilde{\mathbf{v}}^T||_F^2 + P_\lambda(\tilde{\mathbf{v}}), \text{ sub to } ||\tilde{\mathbf{u}}||_2 = 1, \tag{2.4}$$

where $P_\lambda(\tilde{\mathbf{v}})$ could be any sparsity-induced penalty, such as lasso, SCAD, or MCP.

The authors used the alternating direction strategy to solve the problem. They also provided a way of doing cross validation. They did not show the model consistency, however in a following paper ([19]) they completed this part. More details on all listed aspects can be seen in table (2.4).

Performance List	Results and Comments
Regularization/Pre/Post	Regularization
Sequential/Simultaneous	Sequential
Convergence	No
Consistency	Yes
Tuning-Parameter Selection	Cross-Validation
Orthogonality	No
Missing-Values	No

Table 2.4: Performance summary for Shen and Huang [20]

2.2.5 Leng and Wang [10]

This paper is an extension of [26]. First extension is that the tuning parameter was generalized from λ_j to λ_{kj}, which made the model more general yet difficult to do selection. Second aspect is that the authors added weights for sample points.

The authors also showed some consistency results, yet based on a strong assumption: $\bar{\boldsymbol{\alpha}}_j - \boldsymbol{\beta}_j = O_p(n^{-1/2})$, where $\bar{\boldsymbol{\alpha}}_j$ is a parameter in their optimization problem, and it is fixed when they used the alternating direction strategy to solve the optimization problem. However we could not know the true value for $\boldsymbol{\beta}_j$ ($\boldsymbol{\beta}_j$ is the j-th true PC loading vector). In addition, they used BIC for tuning parameter selection, however the selection is within each iteration, which means in actual they used a varying tuning parameter, not exactly followed their formulation. More details could be seen in table (2.5).

Performance List	Results and Comments
Regularization/Pre/Post	Regularization
Sequential/Simultaneous	Simultaneous
Convergence	No
Consistency	Yes (assumption too strong)
Tuning-Parameter Selection	BIC (nested selection)
Orthogonality	No
Missing-Values	No

Table 2.5: Performance summary for Leng and Wang [10]

2.2.6 Witten, Tibshirani, and Hastie [22]

This paper used an SVD-based approach, the formulation for extracting leading layer is:

$$||\mathbf{X} - d\mathbf{u}\mathbf{v}^T||_F^2, \text{ sub to } ||\mathbf{u}||_2 = 1, ||\mathbf{v}||_2 = 1, \mathbf{P}_1(\mathbf{u}) \leq c_1, \mathbf{P}_2(\mathbf{v}) \leq c_2, d \geq 0, \quad (2.5)$$

where $\mathbf{P}_1(\mathbf{u})$ and $\mathbf{P}_2(\mathbf{v})$ are sparsity-induced penalty functions. When only $\mathbf{v}$ needs to be sparse, one can remove the constraint $\mathbf{P}_1(\mathbf{u}) \leq c_1$.

They used the alternating-direction strategy to do optimization (fix $\mathbf{u}$ and update $\mathbf{v}$, then fix $\mathbf{v}$ and update $\mathbf{u}$). They also considered missing values (only in methodology part, not in data analysis part) and orthogonal constraint. More details could be seen in table (2.6).

Performance List	Results and Comments
Regularization/Pre/Post	Regularization
Sequential/Simultaneous	Sequential
Convergence	No
Consistency	No
Tuning-Parameter Selection	Cross-Validation
Orthogonality	Yes
Missing-Values	Yes

Table 2.6: Performance summary for Witten, Tibshirani, and Hastie [22]

2.2.7 Richtarik, Takac, and Ahipasaoglu [17]

This paper proposed a family of methods with eight different ways, by starting from a standard constraint optimization problem:

$$\max ||\mathbf{Ax}||_2, \text{ sub to } ||\mathbf{x}||_2 \leq 1, ||\mathbf{x}||_0 \leq s. \tag{2.6}$$

Then by changing constraint form to penalty form, or changing ℓ_2 norm in the objective function to ℓ_1 norm, or changing ℓ_0 norm in regularization term to ℓ_1 norm, they came up with seven variants.

Similarly, alternating-direction is used to solve the problem by creating an augmented variable $\mathbf{y}$ as follows:

$$\begin{aligned} ||\mathbf{Ax}||_2 &= \max_{\{\mathbf{y}:||\mathbf{y}||_2 \leq 1\}} \mathbf{y}^T\mathbf{Ax} \\ ||\mathbf{Ax}||_1 &= \max_{\{\mathbf{y}:||\mathbf{y}||_\infty \leq 1\}} \mathbf{y}^T\mathbf{Ax} \end{aligned} \tag{2.7}$$

A detailed comparison could be seen in table (2.7).

Performance List	Results and Comments
Regularization/Pre/Post	Regularization
Sequential/Simultaneous	Sequential
Convergence	No
Consistency	No
Tuning-Parameter Selection	Not Provided
Orthogonality	No
Missing-Values	No

Table 2.7: Performance summary for Richtarik, Takac, and Ahipasaoglu [17]

2.2.8 Yang, Ma, and Buja [24]

This paper used the simultaneous way to extract principal components. The method is a modification from power iteration, while there is no formulation for it, which is not good for convergence and consistency proof (therefore these two aspects are not provided in the paper). The algorithm is as follows:

1. Right-to-left multiplication $\mathbf{U} \leftarrow \mathbf{XV}$
2. Left thresholding $\mathbf{U} \leftarrow \eta(\mathbf{U})$
3. Left ortho-normalization using QR decomposition $\mathbf{U} = \mathbf{QR}$, $\mathbf{U} \leftarrow \mathbf{Q}$
4. Left-to-right multiplication $\mathbf{V} \leftarrow \mathbf{X}^T\mathbf{U}$
5. Right thresholding $\mathbf{V} \leftarrow \eta(\mathbf{V})$
6. Right ortho-normalization using QR decomposition $\mathbf{V} = \mathbf{QR}$, $\mathbf{V} \leftarrow \mathbf{Q}$

A detailed comparison could be seen in table (2.8).

Performance List	Results and Comments
Regularization/Pre/Post	Regularization
Sequential/Simultaneous	Simultaneous
Convergence	No
Consistency	No
Tuning-Parameter Selection	Estimation
Orthogonality	No
Missing-Values	No

Table 2.8: Performance summary for Yang, Ma, and Buja [24]

2.2.9 Allen [1]

This paper also used an SVD-based approach, the formulation is

$$\begin{aligned} &\max_{\mathbf{u},\mathbf{v}} \mathbf{u}^T\mathbf{X}\mathbf{v} - \lambda_{\mathbf{u}}\mathbf{P}_{\mathbf{u}}(\mathbf{u}) - \lambda_{\mathbf{v}}\mathbf{P}_{\mathbf{v}}(\mathbf{v}) \\ &\text{sub to } \mathbf{u}^T(\mathbf{I}+\alpha_{\mathbf{u}}\mathbf{\Omega}_{\mathbf{u}})\mathbf{u} \leq 1, \mathbf{v}^T(\mathbf{I}+\alpha_{\mathbf{v}}\mathbf{\Omega}_{\mathbf{v}})\mathbf{v} \leq 1, \end{aligned} \tag{2.8}$$

where $\mathbf{\Omega}_{\mathbf{u}}$ and $\mathbf{\Omega}_{\mathbf{v}}$ are the second-order difference matrices to induce smoothness.

When $\lambda_{\mathbf{u}} = \alpha_{\mathbf{u}} = \alpha_{\mathbf{v}} = 0$, the formulation is actually equivalent to [20], but the algorithms used for two methods are different. In addition, in [1], the convergence is proved. A detailed comparison list could be seen in table (2.9).

Performance List	Results and Comments
Regularization/Pre/Post	Regularization
Sequential/Simultaneous	Sequential
Convergence	Yes
Consistency	Yes
Tuning-Parameter Selection	Not Provided
Orthogonality	No
Missing-Values	No

Table 2.9: Performance summary for Allen [1]

2.2.10 Qi, Luo, and Zhao [16]

This paper proposed an approach for sparse PCA by introducing a new penalty based on mixed ℓ_1 and ℓ_2 norm:

$$||\mathbf{u}||_\alpha = \left[(1-\alpha)||\mathbf{u}||_2^2 + \alpha||\mathbf{u}||_1^2\right]^{1/2}, \forall \mathbf{u} \in \mathbb{R}^p. \tag{2.9}$$

The main formulation is:

$$\max_{\mathbf{u}} \mathbf{u}^T \mathbf{\Sigma} \mathbf{u}, \text{ sub to } ||\mathbf{u}||_\alpha \leq 1, \tag{2.10}$$

where $\mathbf{\Sigma} = \mathbf{X}^T\mathbf{X}$ is the variance-covariance matrix.

By investigation on convexity and strict convexity, the authors found that this form is better than

$$\max_{\mathbf{u}} \mathbf{u}^T \Sigma \mathbf{u}, \text{ sub to } ||\mathbf{u}||_2 = 1 \text{ and } ||\mathbf{u}||_1 \leq t, \tag{2.11}$$

which is one of the mainstream formulation used in sparse PCA area.

The authors also showed the convergence and consistency. They also tried to complete the method for sparse PCA with orthogonal constraint. They proposed the method and proved the convergence, however, the convergence can not be always guaranteed.

More importantly, this method has some equivalence result with our approach, although we start from a totally different philosophy. The equivalence is shown in the methodology section.

A detailed comparison could be seen in table (2.10).

Performance List	Results and Comments
Regularization/Pre/Post	Regularization
Sequential/Simultaneous	Sequential
Convergence	Yes
Consistency	Yes
Tuning-Parameter Selection	Not Provided
Orthogonality	No
Missing-Values	No

Table 2.10: Performance summary for Qi, Luo, and Zhao [16]

2.2.11 Other Papers

In addition to papers introduced above, there are many other papers that we investigated. For example, in [13], the authors investigated many different ways of deflations (which is a key step for the sequential way) and made a comparison w.r.t. orthogonality, explained variance, etc. In [9], the authors developed a smart way to transform their original formulation into a computationally efficient one. They also considered multi-layer cases with orthogonal constraint. In [11], the authors made some meaningful transformations on the mainstream sparse PCA formulation (2.11) and used a first-order gradient optimization method to solve the problem. In [3], the authors made some convex relaxations from their original formulation (sparse PCA using ℓ_0 penalty), and then used the warm-start trick for optimization. In [25], the authors used a block coordinate descent algorithm and reduced the computational complexity for [3]. These papers made important contributions to sparse PCA, however due to description space limit, we did not show more details for them.

3. MAIN FORMULATION

3.1 Sparse PCA via Regularized SVD: from a Scale-Invariance Viewpoint

A standard single-layer SVD for data matrix $\mathbf{X}$ could be done via the following optimization problem:

$$\min_{\mathbf{u},\mathbf{v}} ||\mathbf{X} - \mathbf{u}\mathbf{v}^T||_F^2. \tag{3.1}$$

If we do some investigations on this optimization problem, we could see it has the following properties:

(i) Scale-invariance property, under $\mathbf{u} \leftarrow c\mathbf{u}$, $\mathbf{v} \leftarrow \mathbf{v}/c$, $\forall c > 0$.

(ii) Equivariance property, under $\mathbf{X} \leftarrow c\mathbf{X}$, $\mathbf{u} \leftarrow c\mathbf{u}$, $\forall c > 0$.

This is very different from standard regression problem:

$$\min_{\boldsymbol{\beta}} ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2, \tag{3.2}$$

where it only has equivariance property, under $\mathbf{X} \leftarrow c\mathbf{X}$, $\mathbf{y} \leftarrow c\mathbf{y}$, $\forall c > 0$.

This big difference results in totally different story when we change traditional problem into regularized problem for SVD and regression.

For regression problem, if we would like to make regression coefficient vector $\boldsymbol{\beta}$ to be sparse, we can directly add a sparsity-induced penalty:

$$\min_{\boldsymbol{\beta}} ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||\boldsymbol{\beta}||_1, \tag{3.3}$$

which is the lasso problem.

On the other hand, if we do the same thing on SVD problem, we would have:

$$\min_{\mathbf{u},\mathbf{v}} ||\mathbf{X} - \mathbf{u}\mathbf{v}^T||_F^2 + \lambda||\mathbf{v}||_1. \tag{3.4}$$

This looks reasonable but actually doesn't work according to the following investigations. Given any point $(\mathbf{u}_0, \mathbf{v}_0)$, we could let $\mathbf{u}_1 \leftarrow 2\mathbf{u}_0$ and $\mathbf{v}_1 \leftarrow \mathbf{v}_0/2$, then we could see that the fitting-error term $||\mathbf{X} - \mathbf{u}\mathbf{v}^T||_F^2$ in (3.4) doesn't change, and the penalty term $\lambda||\mathbf{v}||_1$ reduces its value by half. We could do this kind of transformation by multiplying a large value c (even infinity), which makes the penalty term have no influence on the whole formulation.

Therefore any finite solution $(\mathbf{u}^*, \mathbf{v}^*)$ of (3.4) could be improved via $\mathbf{u}^\dagger \leftarrow 2\mathbf{u}^*$, $\mathbf{v}^\dagger \leftarrow \mathbf{v}^*/2$, and thus the problem (3.4) is actually ill posed.

The scale-invariance and equivariance properties are first proposed by [5]. In this paper, the authors tried to analyze two-way functional data via two-way regularized SVD, their formulation is as follows:

$$\begin{aligned}\min_{\mathbf{u},\mathbf{v}} ||\mathbf{X} - \mathbf{u}\mathbf{v}^T||_F^2 + \lambda_{\mathbf{u}}\mathbf{u}^T\mathbf{\Omega_u}\mathbf{u} \cdot \mathbf{v}^T\mathbf{v} + \lambda_{\mathbf{v}}\mathbf{u}^T\mathbf{u} \cdot \mathbf{v}^T\mathbf{\Omega_v}\mathbf{v} + \\ \lambda_{\mathbf{u}}\lambda_{\mathbf{v}}\mathbf{u}^T\mathbf{\Omega_u}\mathbf{u} \cdot \mathbf{v}^T\mathbf{\Omega_v}\mathbf{v},\end{aligned} \tag{3.5}$$

where $\mathbf{\Omega_u}$ and $\mathbf{\Omega_v}$ are the second-order difference matrices to induce smoothness.

We can see that this formulation satisfies the scale-invariance and equivariance properties. Besides, the stationary equations for $\mathbf{u}$ and $\mathbf{v}$ are

$$\mathbf{u} = \frac{(\mathbf{I} + \lambda_{\mathbf{u}}\mathbf{\Omega_u})^{-1}\mathbf{X}\mathbf{v}}{\mathbf{v}^T(\mathbf{I} + \lambda_{\mathbf{v}}\mathbf{\Omega_v})\mathbf{v}}, \qquad \mathbf{v} = \frac{(\mathbf{I} + \lambda_{\mathbf{v}}\mathbf{\Omega_v})^{-1}\mathbf{X}^T\mathbf{u}}{\mathbf{u}^T(\mathbf{I} + \lambda_{\mathbf{u}}\mathbf{\Omega_u})\mathbf{u}}, \tag{3.6}$$

and we can see the two smoothers $(\mathbf{I}+\lambda_{\mathbf{u}}\mathbf{\Omega_u})^{-1}$ and $(\mathbf{I}+\lambda_{\mathbf{v}}\mathbf{\Omega_v})^{-1}$ only involve tuning

parameters. As a comparison, if we use

$$\min_{\mathbf{u},\mathbf{v}} ||\mathbf{X} - \mathbf{u}\mathbf{v}^T||_F^2 + \lambda_{\mathbf{u}}\mathbf{u}^T\mathbf{\Omega}_{\mathbf{u}}\mathbf{u} + \lambda_{\mathbf{v}}\mathbf{v}^T\mathbf{\Omega}_{\mathbf{v}}\mathbf{v}, \tag{3.7}$$

as the formulation for two-way functional data analysis, we could see it does not satisfy the two properties, and has the stationary equations

$$\mathbf{u} = \frac{(\mathbf{I} + \lambda_{\mathbf{u}}/(\mathbf{v}^T\mathbf{v})\mathbf{\Omega}_{\mathbf{u}})^{-1}\mathbf{X}\mathbf{v}}{\mathbf{v}^T\mathbf{v}}, \qquad \mathbf{v} = \frac{(\mathbf{I} + \lambda_{\mathbf{v}}/(\mathbf{u}^T\mathbf{u})\mathbf{\Omega}_{\mathbf{v}})^{-1}\mathbf{X}^T\mathbf{u}}{\mathbf{u}^T\mathbf{u}}, \tag{3.8}$$

from which we can see the two smoothers $(\mathbf{I}+\lambda_{\mathbf{u}}/(\mathbf{v}^T\mathbf{v})\mathbf{\Omega}_{\mathbf{u}})^{-1}$ and $(\mathbf{I}+\lambda_{\mathbf{v}}/(\mathbf{u}^T\mathbf{u})\mathbf{\Omega}_{\mathbf{v}})^{-1}$ involves not only two tuning parameters but also scales of $\mathbf{u}$ and $\mathbf{v}$.

Similarly, if we check the stationary equations for sparse PCA formulation (3.4), we would get:

$$\mathbf{u} = \frac{\mathbf{X}\mathbf{v}}{\mathbf{v}^T\mathbf{v}}, \qquad \mathbf{v} + \frac{1}{2}\cdot\frac{\lambda}{\mathbf{u}^T\mathbf{u}}\cdot\mathbf{sgn}(\mathbf{v}) = \frac{\mathbf{X}^T\mathbf{u}}{\mathbf{u}^T\mathbf{u}}, \tag{3.9}$$

where $\mathbf{sgn}(\cdot)$ is the sign function, which returns value 1 for positive number, -1 for negative number, and value between -1 and 1 for 0.

As a comparison, the stationary equation for standard Lasso Signal Approximation (**LSA**) problem

$$\min_{\mathbf{v}} ||\mathbf{x} - \mathbf{v}||_2^2 + \lambda||\mathbf{v}||_1 \tag{3.10}$$

is

$$\mathbf{v} + \frac{1}{2}\cdot\lambda\cdot\mathbf{sgn}(\mathbf{v}) = \mathbf{x}. \tag{3.11}$$

Therefore, by using formulation (3.4), we are actually processing the standardized signal $\mathbf{X}^T\mathbf{u}/(\mathbf{u}^T\mathbf{u})$ using lasso signal approximation with tuning parameter $\lambda/(\mathbf{u}^T\mathbf{u})$.

Based on the analysis above, we propose our new formulation for sparse PCA using regularized SVD as follows (Eckart-Young form):

$$\min_{\mathbf{u},\mathbf{v}} ||\mathbf{X} - \mathbf{u}\mathbf{v}^T||_F^2 + \lambda\mathbf{u}^T\mathbf{u} \cdot ||\mathbf{v}||_1^2, \tag{3.12}$$

which is equivalent to (Rayleigh-Quotient form)

$$\min_{\mathbf{u},\mathbf{v}} -2\mathbf{u}^T\mathbf{X}\mathbf{v} + \mathbf{u}^T\mathbf{u} \cdot (\lambda||\mathbf{v}||_1^2 + ||\mathbf{v}||_2^2). \tag{3.13}$$

If we check its stationary equation, we get:

$$\mathbf{u} = \frac{\mathbf{X}\mathbf{v}}{\lambda||\mathbf{v}||_1^2 + ||\mathbf{v}||_2^2}, \qquad \mathbf{v} + \lambda||\mathbf{v}||_1 \cdot \mathbf{sgn}(\mathbf{v}) = \frac{\mathbf{X}^T\mathbf{u}}{\mathbf{u}^T\mathbf{u}}. \tag{3.14}$$

Define Squared Lasso Signal Approximation (**SLSA**) as follows:

$$\min_{\mathbf{v}} ||\mathbf{x} - \mathbf{v}||_2^2 + \lambda||\mathbf{v}||_1^2, \tag{3.15}$$

then its stationary equation is

$$\mathbf{v} + \lambda||\mathbf{v}||_1 \cdot \mathbf{sgn}(\mathbf{v}) = \mathbf{x}. \tag{3.16}$$

Therefore we can see that, by using (3.12), we are actually processing the standardized signal $\mathbf{X}^T\mathbf{u}/(\mathbf{u}^T\mathbf{u})$ using squared lasso signal approximation with tuning parameter λ.

Comparing two pairs of smoothers $(\mathbf{I}+\lambda_\mathbf{u}\mathbf{\Omega}_\mathbf{u})^{-1}$, $(\mathbf{I}+\lambda_\mathbf{v}\mathbf{\Omega}_\mathbf{v})^{-1}$, $(\mathbf{I}+\lambda_\mathbf{u}/(\mathbf{v}^T\mathbf{v})\mathbf{\Omega}_\mathbf{u})^{-1}$

and $(\mathbf{I} + \lambda_{\mathbf{v}}/(\mathbf{u}^T\mathbf{u})\mathbf{\Omega}_{\mathbf{v}})^{-1}$ in two-way functional data analysis problem, or comparing two "actual" tuning parameters $\lambda/(\mathbf{u}^T\mathbf{u})$ and λ in sparse PCA problem, we can see that when the scale-invariance and equivariance properties are not satisfied, the tuning parameter(s) and scale(s) of $\mathbf{u}$ (and $\mathbf{v}$) are confounded together. This **Confonding** effect may cause lots of problems in convergence, iteration, consistency and other aspects.

3.2 A Different Viewpoint: New Type of Penalty and an Equivalence Result

In [16], the authors investigated the sparse PCA in a different view point from scale properties and the confounding effect. They started from the "maximize-variance" approach for standard PCA problem:

$$\max_{\mathbf{v}} \mathbf{v}^T\mathbf{X}^T\mathbf{X}\mathbf{v}, \tag{3.17}$$

for which the usual way of inducing sparsity is to add two constraints:

$$||\mathbf{v}||_2 = 1, \qquad ||\mathbf{v}||_1 \leq t. \tag{3.18}$$

This constraint set is convex, but not strictly convex, which yields problem in both algorithmic convergence and sparsity (see section 2 in [16] and Theorem 4 or 5 in [21]). Then they developed another constraint set by introducing a new penalty (see the new designed norm (2.9) in section 2) , and has the following optimization problem (equation 2.10):

$$\max_{\mathbf{v}} \mathbf{v}^T\mathbf{X}^T\mathbf{X}\mathbf{v}, \text{ sub to } ||\mathbf{v}||_\alpha \leq 1.$$

One can verify that the constraint set $||\mathbf{v}||_\alpha \leq 1$ is a strictly convex set. In

addition, they also developed algorithm and showed consistency results for their method.

Besides, if we substitute $||\mathbf{v}||_1$ by $||\mathbf{v}||_\alpha$ in (3.12), we would have the third formulation:

$$\min_{\mathbf{u},\mathbf{v}} ||\mathbf{X} - \mathbf{u}\mathbf{v}^T||_F^2 + \lambda \mathbf{u}^T\mathbf{u} \cdot ||\mathbf{v}||_\alpha^2 \tag{3.19}$$

Now we have three formulations: first one designed from the viewpoint of scale-invariance and equivariance; second one designed from a new penalty and strictly convex constraint set; the third one is designed from using a "stronger" penalty to enhance the formulation. However, all these three formulations are actually equivalent due to some tuning parameter transformation, as shown in following theorem:

Theorem 1. *Formulations (3.12), (2.10), and (3.19) are equivalent up to tuning parameter transformation, or in other words, they has the same full solution path.*

Proof. **(1)** First we compare (3.12) and (2.10).

Note that the stationary equation of $\mathbf{u}$ for (3.12) is

$$\mathbf{u} = \frac{\mathbf{X}\mathbf{v}}{\lambda||\mathbf{v}||_1^2 + ||\mathbf{v}||_2^2}, \tag{3.20}$$

plug this result in (3.13) (equivalent form of (3.12)), we get the marginal optimization problem:

$$\max_{\mathbf{v}} \frac{\mathbf{v}^T\mathbf{X}^T\mathbf{X}\mathbf{v}}{\lambda||\mathbf{v}||_1^2 + ||\mathbf{v}||_2^2}, \tag{3.21}$$

Denote its solution by $\mathbf{v}_0$, define $\tilde{\mathbf{v}}_0 = \mathbf{v}_0/(\lambda||\mathbf{v}_0||_1^2 + ||\mathbf{v}_0||_2^2)$, then we could see that $\tilde{\mathbf{v}}_0$ is the solution of (2.10) with tuning parameter $\alpha = 1/(1+\lambda)$. Therefore

these two formulations are equivalent.

(2) Now we consider (3.19), transform this formulation as Rayleigh-quotient form, we have:

$$\min_{\mathbf{u},\mathbf{v}} -2\mathbf{u}^T\mathbf{X}\mathbf{v} + \mathbf{u}^T\mathbf{u} \cdot [(1+\lambda-\lambda\alpha)||\mathbf{v}||_2^2 + \lambda\alpha||\mathbf{v}||_1^2]. \tag{3.22}$$

Therefore, (3.12) is equivalent to (3.19), with tuning parameter correspondence

$$\lambda \longleftrightarrow \frac{\lambda\alpha}{1+\lambda-\lambda\alpha}. \tag{3.23}$$

□

3.3 Consistency Results in High Dimensions

In [16], the authors showed that under some mild conditions, the stationary point they obtained via their algorithm convergences to the true value of the PC loading vector. Since we have built the equivalence result to their method, we could have the consistency result under the same condition (single component SVD model). We recast the result as following theorem.

Theorem 2. *Suppose the data point* $\mathbf{x}_i$ *comes from the following model*

$$\mathbf{x}_i^{(n)} = \omega_i^{(n)}\mathbf{v}^{(n)} + \sigma\mathbf{z}_i^{(n)}, i = 1, ..., n, \tag{3.24}$$

where $\mathbf{v}^{(n)} = (v_1^{(n)}, ..., v_{p(n)}^{(n)})$*, the single component, is the true signal.* $\omega_i^{(n)}, i = 1, ..., n$ *is a set of i.i.d standard normal variables and* $\mathbf{z}_i^{(n)}$ *are standard normally distributed noise vectors.*

Suppose the true signal has the ℓ_q decay property, i.e.,

$$|\mathbf{v}^{(n)}|_{(\nu)} \leq C\nu^{-1/q}, \nu = 1, ..., p(n), \tag{3.25}$$

where $|\mathbf{v}^{(n)}|_{(\nu)}$ is the ν-th largest components of $|\mathbf{v}^{(n)}|$.

Suppose an extra technique condition is satisfied:

$$\limsup_{m\to\infty} \sup_{n\geq 1} \frac{S_{2m}^{(n)} - S_m^{(n)}}{S_m^{(n)}} < 1, \tag{3.26}$$

where $S_i^{(n)} = \sum_{\nu=1}^{i} |\mathbf{v}^n|_{(\nu)}$.

Then given that $||\mathbf{v}^{(n)}||_2 = 1$, $p(n)/n \to c$ as $n \to \infty$ and $4\sigma\sqrt{c} + 4\sigma^2\sqrt{c} + 6\sigma^2 c < 1$, also, there exists $0 < \alpha < 1/3$ such that $\liminf_{n\to\infty} \lambda^{(n)} n^\alpha > 0$ and $\lim_{n\to\infty} \lambda^{(n)} = 0$, we have

$$\liminf_{n\to\infty} ||\hat{\mathbf{v}}^{(n)} - \mathbf{v}^{(n)}||_2 = 0, \tag{3.27}$$

where $\hat{\mathbf{v}}^{(n)}$ is the optimal point for the sparse PCA problem.

Proof. See section 3 in [16]. □

4. ALGORITHM FOR MAIN FORMULATION

4.1 Alternating-Direction Strategy

In (3.12), there are two variables $\mathbf{u}$ and $\mathbf{v}$, and when one variable is fixed, the formulation becomes a penalized regression problem for the other. Therefore, we use alternating-direction strategy to solve the problem. Note that this is a commonly used strategy, which could be seen in [20], [26], [5], etc.

When $\mathbf{v}$ is fixed, problem (3.12) becomes:

$$\min_{\mathbf{u}} -2\mathbf{u}^T\mathbf{X}\mathbf{v} + \mathbf{u}^T\mathbf{u} \cdot (\lambda||\mathbf{v}||_1^2 + ||\mathbf{v}||_2^2), \tag{4.1}$$

by taking derivatives w.r.t. $\mathbf{u}$, we can see the solution for (4.1) is

$$\mathbf{u} = \frac{\mathbf{X}\mathbf{v}}{\lambda||\mathbf{v}||_1^2 + ||\mathbf{v}||_2^2}, \tag{4.2}$$

which only involves basic arithmetic and matrix-vector multiplication.

When $\mathbf{u}$ is fixed, the problem becomes:

$$\min_{\mathbf{v}} -2\mathbf{u}^T\mathbf{X}\mathbf{v} + \mathbf{u}^T\mathbf{u} \cdot (\lambda||\mathbf{v}||_1^2 + ||\mathbf{v}||_2^2), \tag{4.3}$$

which is equivalent to

$$\min_{\mathbf{v}} ||\mathbf{v} - \frac{\mathbf{X}^T\mathbf{u}}{\mathbf{u}^T\mathbf{u}}||_2^2 + \lambda||\mathbf{v}||_1^2. \tag{4.4}$$

Therefore, to solve this problem, we need to develop an algorithm for the SLSA

problem defined in (3.15):

$$\min_{\mathbf{x}\in\mathbb{R}^n} ||\mathbf{y}-\mathbf{x}||_2^2 + \lambda||\mathbf{x}||_1^2,$$

whose solution is denoted as $\text{SLSA}(\mathbf{y},\lambda)$.

The algorithm to solve $\text{SLSA}(\mathbf{y},\lambda)$ is provided in next subsection. Before doing that, we summarize our alternating-direction algorithm for problem (3.12) as follows.

Alternating-Direction-Algorithm to solve (3.12):
Input: $\mathbf{X}\in\mathbb{R}^{n\times p}$ and $\lambda>0$.
Output: $\mathbf{u}\in\mathbb{R}^n$, $\mathbf{v}\in\mathbb{R}^p$;
Algortihm:

1. Set initial value $\mathbf{v}^0$ for $\mathbf{v}$;
2. For $m=0,1,2,...$, repeat the following steps until convergence:

$$\begin{aligned}\mathbf{u}^{m+1} =& \frac{\mathbf{X}\mathbf{v}^m}{\lambda||\mathbf{v}^m||_1^2 + ||\mathbf{v}^m||_2^2}\\ \mathbf{v}^{m+1} =& \text{SLSA}\big(\frac{\mathbf{X}^T\mathbf{u}^{m+1}}{(\mathbf{u}^{m+1})^T\mathbf{u}^{m+1}},\lambda\big),\end{aligned} \tag{4.5}$$

where $\text{SLSA}(\cdot,\cdot)$ can be solved via **SLSA-Algorithm** in next section.

4.2 SLSA-Algorithm

First we investigate some properties for $\text{SLSA}(\cdot,\cdot)$ function and have the following theorem:

Theorem 3. *Suppose* $\mathbf{y}=(y_1,...,y_n)^T$ *is the source signal,* $\mathbf{t}=(t_1,...,t_n)$ *is the signal obtained after thresholding (or called estimator of* $\mathbf{y}$*),* $\mathbf{t}=SLSA(\mathbf{y},\lambda)$*, then we have*

(a) ***(sign preservation)*** *If $y_i > 0$, then $t_i \geq 0$; if $y_i = 0$, then $t_i = 0$; if $y_i < 0$, then $t_i \leq 0$.*

(b) ***(order preservation)*** *If $|y_i| > |y_j|$, then $|t_i| \geq |t_j|$, and the equality holds only when $t_i = t_j = 0$.*

(c) ***(mass preservation)*** *If $|y_i| = |y_j|$, then $|t_i| = |t_j|$.*

(d) ***(compression)*** $|t_i| \leq |y_i|$, $1 \leq i \leq n$.

We can see most commonly used threshold functions, such as soft-threshold, hard-threshold and SCAD-threshold, satisfy the above properties. These properties also justify that our squared lasso penalty is a meaningful penalty.

Proof. We call a set of points $\mathbf{S}$ a **complete class** for optimization problem $min_{\mathbf{x}} f(\mathbf{x}) = ||\mathbf{y} - \mathbf{x}||_2^2 + \lambda||\mathbf{x}||_1^2$, if $\forall \mathbf{x}$, there exists $\mathbf{x}_0 \in \mathbf{S}$ such that $f(\mathbf{x}) \leq f(\mathbf{x}_0)$. We can see the optimal point is always in the complete class.

(a) For any source signal $\mathbf{y}$ with $y_i > 0$, and any estimator $\mathbf{x}$ with $x_i < 0$, let $\tilde{\mathbf{x}}$ take the same value as $\mathbf{x}$ except that $\tilde{x}_i = 0$, then $||\mathbf{x}||_1 = ||\tilde{\mathbf{x}}||_1 + |x_i| > ||\tilde{\mathbf{x}}||_1$ and

$$f(\mathbf{x}) - f(\tilde{\mathbf{x}}) = x_i^2 + \lambda(||\mathbf{x}||_1 - ||\tilde{\mathbf{x}}||_1)^2 > 0. \tag{4.6}$$

For any estimator $\mathbf{x}$ with $x_i \geq 0$, just take $\tilde{\mathbf{x}} = \mathbf{x}$, thus $\mathbf{S} = \{\mathbf{x} : x_i \geq 0\}$ is a complete class for this given $\mathbf{y}$, and therefore $(\text{SLSA}(\mathbf{y}))_i \geq 0$.

For cases of $y_i = 0$ and $y_i < 0$, just note that their complete classes are $\mathbf{S} = \{\mathbf{x} : x_i = 0\}$ and $\mathbf{S} = \{\mathbf{x} : x_i \leq 0\}$, respectively. All others are similar.

(b) For any source signal $\mathbf{y}$ with $|y_i| > |y_j|$, and any estimator $\mathbf{x}$ with $|x_i| < |x_j|$, let $\tilde{\mathbf{x}}$ take the same value as $\mathbf{x}$ except that $\tilde{x}_i = sgn(y_i)|x_j|$ and $\tilde{x}_j = sgn(y_j)|x_i|$,

then $|\tilde{x}_i| > |\tilde{x}_j|$, $||\mathbf{x}||_1 = ||\tilde{\mathbf{x}}||_1$ and

$$\begin{aligned}
&f(\mathbf{x}) - f(\tilde{\mathbf{x}}) \\
=&(y_i - x_i)^2 + (y_j - x_j)^2 - (y_i - sgn(y_i)|x_j|)^2 - (y_j - sgn(y_j)|x_i|)^2 \\
=&2(|y_i||x_j| + |y_j||x_i| - x_i y_i - x_j y_j) \\
\geq&2(|y_i||x_j| + |y_j||x_i| - |x_i||y_i| - |x_j||y_j|) \\
=&2(|y_i| - |y_j|)(|x_j| - |x_i|) \\
>&0
\end{aligned} \tag{4.7}$$

For another estimator $\mathbf{z}$ with $|z_i| = |z_j| \neq 0$, let $\tilde{\mathbf{z}}$ take the same value as $\mathbf{z}$ except that $\tilde{z}_i = sgn(y_i)(|z_i| + \epsilon)$ and $\tilde{z}_j = sgn(y_j)(|z_j| - \epsilon)$, where ϵ is a enough small number, then $|\tilde{z}_i| > |\tilde{z}_j|$, $||\mathbf{z}||_1 = ||\tilde{\mathbf{z}}||_1$ (note that if $|z_i| = |z_j| = 0$, we cannot have this equality), and

$$\begin{aligned}
&f(\mathbf{z}) - f(\tilde{\mathbf{z}}) \\
=&(y_i - z_i)^2 + (y_j - z_j)^2 - \\
&(y_i - sgn(y_i)(|z_i| + \epsilon))^2 - (y_j - sgn(y_j)|z_j| - \epsilon))^2 \\
=&-2(y_i z_i + y_j z_j) + 2(|y_i||z_i| + |y_j||z_j|) - 2\epsilon^2 + \\
&2\epsilon[(y_i - sgn(y_i)|z_i|)sgn(y_i) - (y_j - sgn(y_j)|z_j|)sgn(y_j)] \\
=&-2(y_i z_i + y_j z_j) + 2(|y_i||z_i| + |y_j||z_j|) - 2\epsilon^2 + 2\epsilon(|y_i| - |y_j|) \\
>&0 \text{ (when } \epsilon \text{ is small enough)}
\end{aligned} \tag{4.8}$$

Therefore $\mathbf{S} = \{\mathbf{x} : |x_i| > |x_j| \text{ or } |x_i| = |x_j| = 0\}$ is a complete class for this given $\mathbf{y}$, thus we finish the proof of **(b)**.

(c) For any source signal $\mathbf{y}$ with $|y_i| = |y_j|$ and any estimator $\mathbf{x}$ with $|x_i| > |x_j|$, let $\tilde{\mathbf{x}}$ take the same value as $\mathbf{x}$ except that $\tilde{x}_i = sgn(y_i)(|x_i| + |x_j|)/2$ and $\tilde{x}_j =$

$sgn(y_j)(|x_i| + |x_j|)/2$, then $|\tilde{x}_i| = |\tilde{x}_j|$, $||\mathbf{x}||_1 = ||\tilde{\mathbf{x}}||_1$ and

$$\begin{aligned}
&f(\mathbf{x}) - f(\tilde{\mathbf{x}}) \\
=&(y_i - x_i)^2 + (y_j - x_j)^2 - \\
&(y_i - sgn(y_i)(|x_i| + |x_j|)/2)^2 - (y_j - sgn(y_j)(|x_i| + |x_j|)/2)^2 \\
=&(x_i^2 + x_j^2)/2 - |x_i||x_j| - 2x_i y_i - 2x_j y_j + (|y_i| + |y_j|)(|x_i| + |x_j|) \\
>&0.
\end{aligned} \tag{4.9}$$

Therefore $\mathbf{S} = \{\mathbf{x} : |x_i| = |x_j|\}$ is a complete class for this given $\mathbf{y}$, thus we finish the proof of **(c)**.

(d) For any source signal $\mathbf{y}$ with $|y_i| \geq 0$ and any estimator $\mathbf{x}$ with $|x_i| > |y_i|$, let $\tilde{\mathbf{x}}$ take the same value as $\mathbf{x}$ except that $\tilde{x}_i = y_i$, then $|\tilde{x}_i| \leq |y_i|$, $||\mathbf{x}||_1 > ||\tilde{\mathbf{x}}||_1$ and

$$f(\mathbf{x}) - f(\tilde{\mathbf{x}}) = (y_i - x_i)^2 + \lambda(||\mathbf{x}||_1 - ||\tilde{\mathbf{x}}||_1)^2 > 0. \tag{4.10}$$

Therefore $\mathbf{S} = \{\mathbf{x} : |x_i| \leq |y_i|, 1 \leq i \leq n\}$ is a complete class for this given $\mathbf{y}$, thus we finish the proof of **(d)**. □

According to those properties we obtained for SLSA problem, for a given signal $\mathbf{y}$, first we sort its coordinates by absolute values: $|y_{k_1}| \geq |y_{k_2}| \geq ... \geq |y_{k_n}|$, define $\mathbf{z} = \phi(\mathbf{y}) = (y_{k_1}, ..., y_{k_n})$, then $\text{SLSA}(\mathbf{y}) = \phi^{-1}(\text{SLSA}(\mathbf{z}))$, and $\text{SLSA}(\mathbf{z})$ should have the form $(\tilde{z}_1, ..., \tilde{z}_r, 0, ..., 0)^T$, where $\tilde{z}_i$ has the same sign as z_i and has smaller or same absolute value, $1 \leq i \leq r$. Therefore, to solve problem (3.15), we need to:

1. determine the value of r.

2. determine values of $\tilde{z}_i$, $1 \leq i \leq r$.

Suppose the solution of equation (3.15) substituting $\mathbf{y}$ by $\mathbf{z}$ is $\text{SLSA}(\mathbf{z}) = \tilde{\mathbf{z}}$, take

subdifferential for the problem, then we have

$$\mathbf{0} \in 2(\tilde{\mathbf{z}} - \mathbf{z}) + 2\lambda||\tilde{\mathbf{z}}||_1 \cdot sgn(\tilde{\mathbf{z}}) \tag{4.11}$$

For $1 \leq i \leq r$, $sgn(\tilde{z}_i) = sgn(z_i) \in \{1, -1\}$, therefore we have

$$\begin{aligned} & \tilde{z}_i - z_i + \lambda||\tilde{\mathbf{z}}||_1 \cdot sgn(z_i) = 0 \\ \Leftrightarrow \quad & z_i = \tilde{z}_i + \lambda||\tilde{\mathbf{z}}||_1 \cdot sgn(z_i) \\ \Leftrightarrow \quad & |z_i| = |\tilde{z}_i| + \lambda||\tilde{\mathbf{z}}||_1, \end{aligned} \tag{4.12}$$

where the last equation is obtained by multiplying $sgn(z_i)$ on both sides. Note that $||\tilde{\mathbf{z}}||_1 = \Sigma_{i=1}^{n}|\tilde{z}_i| = \Sigma_{i=1}^{r}|\tilde{z}_i|$, therefore we have a system of linear equations about $|\tilde{\mathbf{z}}_0| = (|\tilde{z}_1|, ..., |\tilde{z}_r|)^T$ and $|\mathbf{z}_0| = (|z_1|, ..., |z_r|)^T$:

$$\begin{aligned} & (\mathbf{I}_r + \lambda \mathbf{1}_r \mathbf{1}_r^T)|\tilde{\mathbf{z}}_0| = |\mathbf{z}_0| \\ \Rightarrow \ & |\tilde{\mathbf{z}}_0| = (\mathbf{I}_r + \lambda \mathbf{1}_r \mathbf{1}_r^T)^{-1}|\mathbf{z}_0| = (\mathbf{I}_r - \frac{\lambda}{\lambda r + 1}\mathbf{1}_r \mathbf{1}_r^T)|\mathbf{z}_0|. \end{aligned} \tag{4.13}$$

This result implies that once we know the value of r, $\tilde{z}_i$'s can be determined easily.

To determine the value of r, first note that $\tilde{z}_i$ has the same sign as z_i $(1 \leq i \leq r)$, then $|z_i| - \frac{\lambda}{\lambda r+1}\Sigma_{j=1}^{r}|z_j| > 0$, $1 \leq i \leq r$.

Secondly, for $r < i \leq n$, we have

$$\begin{aligned} & 0 \in 2(\tilde{z}_i - z_i) + 2\lambda||\tilde{\mathbf{z}}||_1 \cdot sgn(\tilde{z}_i) \\ \Leftrightarrow & z_i/(\lambda||\tilde{\mathbf{z}}||_1) \in sgn(\tilde{z}_i) = [-1, 1] \\ \Rightarrow & |z_i| \leq \lambda||\tilde{\mathbf{z}}||_1 = \lambda\Sigma_{j=1}^{r}(|z_j| - \frac{\lambda}{\lambda r + 1}\Sigma_{k=1}^{r}|z_k|) \\ & = \frac{\lambda}{\lambda r + 1}\Sigma_{j=1}^{r}|z_j|. \end{aligned} \tag{4.14}$$

Therefore we can see the relationship between $|z_i|$ and $\frac{\lambda}{\lambda r+1}\Sigma_{j=1}^{r}|z_j|$ is a boundary between nonzero elements and zero elements. Most importantly, we have theoretical support on the existence and uniqueness of this boundary:

Lemma 1. *The value r such that*

$$|z_i| > \frac{\lambda}{\lambda r+1}\Sigma_{j=1}^{r}|z_j|, 1 \leq i \leq r \tag{4.15}$$

and

$$|z_i| \leq \frac{\lambda}{\lambda r+1}\Sigma_{j=1}^{r}|z_j|, r < i \leq n \tag{4.16}$$

exists and is unique. Furthermore, $(sgn(z_1)(|z_1| - \frac{\lambda}{\lambda r+1}\Sigma_{j=1}^{r}|z_j|), ..., (sgn(z_r)(|z_r| - \frac{\lambda}{\lambda r+1}\Sigma_{j=1}^{r}|z_j|), 0, ..., 0)^T$ is the unique solution of problem (3.15).

Proof. Because $|z_i| \geq |z_r|$ $(1 \leq i \leq r)$ and $|z_i| \leq |z_{r+1}|$ $(r < i \leq n)$, it is equivalent to show the uniqueness of r satisfying

$$|z_r| > \frac{\lambda}{\lambda r+1}\Sigma_{j=1}^{r}|z_j|, \quad |z_{r+1}| \leq \frac{\lambda}{\lambda r+1}\Sigma_{j=1}^{r}|z_j|. \tag{4.17}$$

Define $S_r = |z_r| - \Sigma_{j=1}^{r}|z_j|$ and $T_r = |z_{r+1}| - \frac{\lambda}{\lambda r+1}\Sigma_{j=1}^{r}|z_j|$ $(1 \leq r \leq n)$, then, define $P = \{i : 1 \leq i \leq n, S_r > 0\}$ and $Q = \{i : 1 \leq i \leq n, T_r \leq 0\}$.

Because $|z_1| > \frac{\lambda}{\lambda+1}\Sigma_{j=1}^{1}|z_j| = \frac{\lambda}{\lambda+1}|z_1|$ and $z_{n+1} = 0 \leq \frac{\lambda}{\lambda n+1}\Sigma_{j=1}^{n}|z_j|$ always hold, we have $1 \in P$ and $n \in Q$.

Note that

$$\begin{aligned} S_r &= |z_r| - \frac{\lambda}{\lambda r+1}\Sigma_{j=1}^{r}|z_j| \\ &= \frac{\lambda(r-1)+1}{\lambda r+1}|z_r| - \frac{\lambda}{\lambda r+1}\Sigma_{j=1}^{r-1}|z_j| \\ &= \frac{\lambda(r-1)+1}{\lambda r+1}(|z_r| - \frac{\lambda}{\lambda(r-1)+1}\Sigma_{j=1}^{r-1}|z_j|) \\ &\leq \frac{\lambda(r-1)+1}{\lambda r+1}(|z_{r-1}| - \frac{\lambda}{\lambda(r-1)+1}\Sigma_{j=1}^{r-1}|z_j|) \\ &= \frac{\lambda(r-1)+1}{\lambda r+1}S_{r-1}, \end{aligned} \tag{4.18}$$

thus $S_r > 0 \Rightarrow S_{r-1} > 0$.

Similarly,

$$\begin{aligned} T_r &= |z_{r+1}| - \frac{\lambda}{\lambda r+1}\Sigma_{j=1}^{r}|z_j| \\ &= \frac{\lambda(r+1)+1}{\lambda r+1}|z_{r+1}| - \frac{\lambda}{\lambda r+1}\Sigma_{j=1}^{r+1}|z_j| \\ &= \frac{\lambda(r+1)+1}{\lambda r+1}(|z_{r+1}| - \frac{\lambda}{\lambda(r+1)+1}\Sigma_{j=1}^{r+1}|z_j|) \\ &\geq \frac{\lambda(r+1)+1}{\lambda r+1}(|z_{r+2}| - \frac{\lambda}{\lambda(r+1)+1}\Sigma_{j=1}^{r+1}|z_j|) \\ &= \frac{\lambda(r+1)+1}{\lambda r+1}T_{r+1}, \end{aligned} \tag{4.19}$$

thus $T_r \leq 0 \Rightarrow T_{r+1} \leq 0$.

Therefore if define $p = \max P$, $q = \min Q$, then $P = \{1, ..., p\}$, $Q = \{q, ..., n\}$.

On the other hand, note that

$$\begin{aligned} S_r &= \frac{\lambda(r-1)+1}{\lambda r+1}(|z_r| - \frac{\lambda}{\lambda(r-1)+1}\Sigma_{j=1}^{r-1}|z_j|) \text{ (from (4.18))} \\ &= \frac{\lambda(r-1)+1}{\lambda r+1}T_{r-1}. \end{aligned} \tag{4.20}$$

Therefore

$$p+1 \notin P \Rightarrow S_{p+1} \leq 0 \Rightarrow T_p \leq 0 \Rightarrow p \in Q \Rightarrow p \geq q, \tag{4.21}$$

and

$$p \in P \Rightarrow S_p > 0 \Rightarrow T_{p-1} > 0 \Rightarrow p-1 \notin Q \Rightarrow p-1 < q. \tag{4.22}$$

Thus $p = q$ and $P \cap Q = \{p\}$, and we complete the proof. □

Given the results above, we could design an algorithm for SLSA problem as follows.

SLSA-Algorithm to solve (3.15):
Input: $\mathbf{y} \in \mathbb{R}^n$ and $\lambda > 0$.
Output: $\text{SLSA}(\mathbf{y}; \lambda) \in \mathbb{R}^n$;
Algorithm:

1. Sort the coordinates y_i's of $\mathbf{y}$ by their absolute values: $|y_{k_1}| \geq |y_{k_2}| \geq ... \geq |y_{k_n}|$. Let $z_i = y_{k_i}$, $1 \leq i \leq n$ and define $z_{n+1} = 0$. Then we say $\mathbf{z} = (z_1, ..., z_n)^T = \phi(\mathbf{y})$, with ϕ a transformation of permuting the coordinates.

2. Compute $S_i = \Sigma_{j=1}^{i} |z_j|$ and find the unique $r \in \{1, 2, ..., n\}$ satisfying

$$|z_{r+1}| \leq \frac{\lambda S_r}{1 + r\lambda} < |z_r|.$$

3. Compute $\tilde{z}_i = \begin{cases} \mathbf{sgn}(z_i) \cdot (|z_i| - \frac{\lambda S_r}{1+r\lambda}) & \text{if } i \leq r, \\ 0 & \text{if } i > r. \end{cases}$

4. Then $\text{SLSA}(\mathbf{y}; \lambda) = \phi^{-1}(\tilde{\mathbf{z}})$, with $\tilde{\mathbf{z}} = (\tilde{z}_1, ..., \tilde{z}_n)^T$.

5. MULTIPLE PCS: SPARSE PCA WITH ORTHOGONAL CONSTRAINT

5.1 Alternating-Direction Strategy and SLOCSA Problem

There are three ways to extract multiple layers of singular value decomposition for principle components: (1) extract multiple layers simultaneously; (2) use deflation method (there are many kinds of deflation methods, see [12]); and (3) extract higher-order layers with orthogonal constraint to previous layers. It is difficult to design proper regularization term to method (1), while different layers obtained from deflation method (2) may not necessarily be orthogonal to each other. Therefore we consider adding orthogonal constraint when solving the optimization problem for extracting higher-order layers.

Suppose we already extracted k-layers, and obtained $\{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_k\}$, denote the basis matrices of span$\{\mathbf{v}_1, ..., \mathbf{v}_k\}$ as $\mathbf{V}_k$, when $\{\mathbf{v}_1, ..., \mathbf{v}_k\}$ are mutually orthogonal and have unit length, $\mathbf{V}_k = (\mathbf{v}_1, ..., \mathbf{v}_k)$, otherwise we need to do QR decomposition, then the formulations for $(k+1)$-th layer regularized SVD with orthogonal constraint is:

$$\min_{\mathbf{u},\mathbf{v}} ||\mathbf{X} - \mathbf{u}\mathbf{v}^T||_F^2 + \lambda \mathbf{u}^T\mathbf{u} \cdot ||\mathbf{v}||_1^2, \text{ sub to } \mathbf{v} \perp \mathbf{V}_k, \tag{5.1}$$

We still use alternating direction strategy to solve the above problem above.

When we fix $\mathbf{v}$ and update $\mathbf{u}$, the problem becomes

$$\min_{\mathbf{u}} -2\mathbf{u}^T\mathbf{X}\mathbf{v} + \mathbf{u}^T\mathbf{u} \cdot (\lambda||\mathbf{v}||_1^2 + ||\mathbf{v}||_2^2), \tag{5.2}$$

taking derivative w.r.t. $\mathbf{u}$, we get its solution

$$\mathbf{u} = \frac{\mathbf{X}\mathbf{v}}{\lambda||\mathbf{v}||_1^2 + ||\mathbf{v}||_2^2}, \tag{5.3}$$

which is the same as the one without constraint.

When we fix $\mathbf{u}$ and update $\mathbf{v}$, the problem becomes

$$\min_{\mathbf{v}} -2\mathbf{u}^T\mathbf{X}\mathbf{v} + \mathbf{u}^T\mathbf{u} \cdot (\lambda||\mathbf{v}||_1^2 + ||\mathbf{v}||_2^2), \text{ sub to } \mathbf{v} \perp \mathbf{V}_k, \tag{5.4}$$

which is equivalent to

$$\min_{\mathbf{v}} ||\mathbf{v} - \frac{\mathbf{X}^T\mathbf{u}}{\mathbf{u}^T\mathbf{u}}||_2^2 + \lambda||\mathbf{v}||_1^2, \text{ sub to } \mathbf{v} \perp \mathbf{V}_k, \tag{5.5}$$

In other words, when $\mathbf{v}$ is fixed, problem (5.5) is a squared lasso penalized regression problem with identity design matrix and orthogonal constraint, which is also called squared lasso with orthogonal constraint signal approximation (**SLOCSA**) problem.

Therefore to utilize the alternating-direction strategy here, we need to design an algorithm for SLOCSA problem:

$$\min_{\mathbf{x}\in\mathbb{R}^n} ||\mathbf{y} - \mathbf{x}||_2^2 + \lambda||\mathbf{x}||_1^2, \text{ sub to } \mathbf{x} \perp \mathbf{V}. \tag{5.6}$$

This problem is investigated in next subsection. Before that, we summarize the alternating direction algorithm for regularized SVD with orthogonal constraint, which uses **SLOCSA-ADMM-Algorithm** or **SLOCSA-QP-Algorithm** in following subsections.

Alternating-Direction-Algorithm to solve (5.1):

Input: $\mathbf{X} \in \mathbb{R}^{n\times p}$ and $\lambda > 0$, $\mathbf{V}_k \in \mathbb{R}^{k\times p}$.

Output: $\mathbf{u} \in \mathbb{R}^n$, $\mathbf{v} \in \mathbb{R}^p$;

Algortihm:

1. Set initial value $\mathbf{v}^0$ for $\mathbf{v}$;

2. For $m = 0, 1, 2, ...$, repeat the following steps until convergence:

$$\begin{aligned} \mathbf{u}^{m+1} =& \frac{\mathbf{X}\mathbf{v}^m}{\lambda||\mathbf{v}^m||_1^2 + ||\mathbf{v}^m||_2^2} \\ \mathbf{v}^{m+1} =& \text{SLOCSA}\big(\frac{\mathbf{X}^T\mathbf{u}^{m+1}}{(\mathbf{u}^{m+1})^T\mathbf{u}^{m+1}}, \lambda, \mathbf{V}_k\big), \end{aligned} \tag{5.7}$$

   where $\text{SLOCSA}(\cdot,\cdot,\cdot)$ can be solved via **SLOCSA-ADMM-Algorithm** or **SLOCSA-QP-Algorithm** in following subsection.

5.2 SLOCSA Using ADMM Algorithm

In this section, we solve the SLOCSA problem using alternating direction method of multipliers (**ADMM**). ADMM is a popular and widely-used optimization method which can be considered as a version of method of multipliers where a single *Gauss-Seidel* pass is used. Consider the following optimization problem (see equation (3.1) in page 13 of [2]):

$$\min_{\mathbf{x}\in\mathbb{R}^n, \mathbf{z}\in\mathbb{R}^m} f(\mathbf{x}) + g(\mathbf{z}), \text{ sub to } \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{c}, \tag{5.8}$$

where $\mathbf{A} \in \mathbb{R}^{p\times n}$, $\mathbf{B} \in \mathbb{R}^{p\times m}$ and $\mathbf{c} \in \mathbb{R}^p$, with both $f : \mathbb{R}^n \mapsto \mathbb{R}$ and $g : \mathbb{R}^m \mapsto \mathbb{R}$ are convex.

The augmented Lagrangian of (5.8) is defined as

$$L_\rho(\mathbf{x}, \mathbf{z}, \boldsymbol{\phi}) = f(\mathbf{x}) + g(\mathbf{z}) + \boldsymbol{\phi}^T(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}) + \rho/2||\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}||_2^2. \tag{5.9}$$

ADMM consists of the following iteration step

$$\begin{aligned} \mathbf{x}^{k+1} :=& arg\min_{\mathbf{x}} L_\rho(\mathbf{x}, \mathbf{z}^k, \boldsymbol{\phi}^k) \\ \mathbf{z}^{k+1} :=& arg\min_{\mathbf{x}} L_\rho(\mathbf{x}^{k+1}, \mathbf{z}, \boldsymbol{\phi}^k) \\ \boldsymbol{\phi}^{k+1} :=& \boldsymbol{\phi}^k + \rho(\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z}^{k+1} - \mathbf{c}) \end{aligned} \tag{5.10}$$

Remark 1. *To facilitate the description, we change the notation of* $\mathbf{y}$ *in [2] into* $\boldsymbol{\phi}$ *here.*

To utilize ADMM method and solve SLOCSA problem, first define the basis matrix of orthogonal complement of $\mathbf{span}(\mathbf{V})$ as $\mathbf{V}^\perp : n \times (n-k)$, then $\mathbf{x} \perp \mathbf{V} \Leftrightarrow \mathbf{x} = \mathbf{V}^\perp\mathbf{z}$, with $\mathbf{z} \in \mathbb{R}^{n-k}$. Let $f(\mathbf{x}) = \lambda||\mathbf{x}||_1^2$, $g(\mathbf{z}) = ||\mathbf{y} - \mathbf{V}^\perp\mathbf{z}||_2^2$, $\mathbf{A} = \mathbf{I}_n$, $\mathbf{B} = -\mathbf{V}^\perp$, $\mathbf{c} = \mathbf{0}$ and $m = n-k$, $p = n$, then (5.8) becomes

$$\min_{\mathbf{x}\in\mathbb{R}^n, \mathbf{z}\in\mathbb{R}^{n-k}} \lambda||\mathbf{x}||_1^2 + ||\mathbf{y} - \mathbf{V}^\perp\mathbf{z}||_2^2, \text{ sub to } \mathbf{x} - \mathbf{V}^\perp\mathbf{z} = \mathbf{0}, \tag{5.11}$$

and we can see this is equivalent to SLOCSA problem (5.6). In addition, $f(\mathbf{x}) = \lambda||\mathbf{x}||_1^2$ and $g(\mathbf{z}) = ||\mathbf{y} - \mathbf{V}^\perp\mathbf{z}||_2^2$ are convex functions.

The augmented Lagrangian becomes

$$L_\rho(\mathbf{x}, \mathbf{z}, \boldsymbol{\phi}) = \lambda||\mathbf{x}||_1^2 + ||\mathbf{y} - \mathbf{V}^\perp\mathbf{z}||_2^2 + \boldsymbol{\phi}^T(\mathbf{x} - \mathbf{V}^\perp\mathbf{z}) + \rho/2||\mathbf{x} - \mathbf{V}^\perp\mathbf{z}||_2^2. \tag{5.12}$$

The iteration step (5.10) becomes:

$$\begin{aligned}
\mathbf{x}^{k+1} =& arg\min_{\mathbf{x}} L_\rho(\mathbf{x}, \mathbf{z}^k, \boldsymbol{\phi}^k) \\
=& arg\min_{\mathbf{x}} \lambda||\mathbf{x}||_1^2 + ||\mathbf{y} - \mathbf{V}^\perp\mathbf{z}^k||_2^2 + (\boldsymbol{\phi}^k)^T(\mathbf{x} - \mathbf{V}^\perp\mathbf{z}^k) + \rho/2||\mathbf{x} - \mathbf{V}^\perp\mathbf{z}^k||_2^2 \\
=& arg\min_{\mathbf{x}} \lambda||\mathbf{x}||_1^2 + (\boldsymbol{\phi}^k)^T\mathbf{x} + \rho/2||\mathbf{x} - \mathbf{V}^\perp\mathbf{z}^k||_2^2 \\
=& \text{SLSA}(\mathbf{V}^\perp\mathbf{z}^k - \boldsymbol{\phi}^k/\rho; 2\lambda/\rho).
\end{aligned}$$

$$\begin{aligned}
\mathbf{z}^{k+1} =& arg\min_{\mathbf{z}} L_\rho(\mathbf{x}^{k+1}, \mathbf{z}, \boldsymbol{\phi}^k) \\
=& arg\min_{\mathbf{z}} \lambda||\mathbf{x}^{k+1}||_1^2 + ||\mathbf{y} - \mathbf{V}^\perp\mathbf{z}||_2^2 + (\boldsymbol{\phi}^k)^T(\mathbf{x}^{k+1} - \mathbf{V}^\perp\mathbf{z}) + \rho/2||\mathbf{x}^{k+1} - \mathbf{V}^\perp\mathbf{z}||_2^2 \\
=& arg\min_{\mathbf{z}} ||\mathbf{y} - \mathbf{V}^\perp\mathbf{z}||_2^2 - (\boldsymbol{\phi}^k)^T\mathbf{V}^\perp\mathbf{z} + \rho/2||\mathbf{x}^{k+1} - \mathbf{V}^\perp\mathbf{z}||_2^2 \\
=& (\mathbf{V}^\perp)^T(2\mathbf{y} + \boldsymbol{\phi}^k + \rho\mathbf{x}^{k+1})/(2 + \rho).
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{\phi}^{k+1} =& \boldsymbol{\phi}^k + \rho(\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{z}^{k+1} - \mathbf{c}) \\
=& \boldsymbol{\phi}^k + \rho(\mathbf{x}^{k+1} - \mathbf{V}^\perp\mathbf{z}^{k+1})
\end{aligned} \tag{5.13}$$

Usually $k \ll n$, and it is difficult to calculate and store the matrix $\mathbf{V}^\perp$, therefore we need to modify our iteration step to avoid calculating $\mathbf{V}^\perp$.

Define $\mathbf{w} = \mathbf{V}^{\perp}\mathbf{z}$, we can see the iteration step can be rewritten as

$$\begin{aligned}
\mathbf{x}^{k+1} =& \text{SLSA}(\mathbf{V}^{\perp}\mathbf{z}^k - \boldsymbol{\phi}^k/\rho; 2\lambda/\rho) \\
=& \text{SLSA}(\mathbf{w}^k - \boldsymbol{\phi}^k/\rho; 2\lambda/\rho)
\end{aligned}$$

$$\begin{aligned}
\mathbf{w}^{k+1} =& \mathbf{V}^{\perp}\mathbf{z}^{k+1} \\
=& \mathbf{V}^{\perp}(\mathbf{V}^{\perp})^T(2\mathbf{y} + \boldsymbol{\phi}^k + \rho\mathbf{x}^{k+1})/(2+\rho) \\
=& (\mathbf{I}_n - \mathbf{V}\mathbf{V}^T)(2\mathbf{y} + \boldsymbol{\phi}^k + \rho\mathbf{x}^{k+1})/(2+\rho)
\end{aligned} \tag{5.14}$$

$$\begin{aligned}
\boldsymbol{\phi}^{k+1} =& \boldsymbol{\phi}^k + \rho(\mathbf{x}^{k+1} - \mathbf{V}^{\perp}\mathbf{z}^{k+1}) \\
=& \boldsymbol{\phi}^k + \rho(\mathbf{x}^{k+1} - \mathbf{w}^{k+1})
\end{aligned}$$

We can see iteration step for $\boldsymbol{\phi}$ only involves vector addition, iteration step for $\mathbf{w}$ only involves elementary matrix operations, and iteration step for $\mathbf{x}$ requires performing a standard SLSA algorithm, which only involves sorting of elements and elementwise thresholding, therefore all three steps are computationally efficient.

We summarize the results above as the following algorithm.

SLOCSA-ADMM-Algorithm to solve (5.6):

Input: $\mathbf{y} \in \mathbb{R}^n$, $\lambda > 0$, $\mathbf{V} \in \mathbb{R}^{n\times k}$, and $\rho > 0$.

Output: $\text{SLOCSA}(\mathbf{y}; \lambda, \mathbf{V}) \in \mathbb{R}^n$;

Algorithm:

1. Set initial values for $\mathbf{x}^0$, $\mathbf{w}^0$, and $\boldsymbol{\phi}^0$.

2. For $m = 0, 1, 2, ...$, repeat the following steps until convergence:

$$\begin{aligned}
\mathbf{x}^{m+1} =& \text{SLSA}(\mathbf{w}^m - \boldsymbol{\phi}^m/\rho; 2\lambda/\rho) \\
\mathbf{w}^{m+1} =& (\mathbf{I}_n - \mathbf{V}\mathbf{V}^T)(2\mathbf{y} + \boldsymbol{\phi}^m + \rho\mathbf{x}^{m+1})/(2+\rho) \\
=& \mathbf{V}^\perp(\mathbf{V}^\perp)^T(2\mathbf{y} + \boldsymbol{\phi}^k + \rho\mathbf{x}^{k+1})/(2+\rho) \\
\boldsymbol{\phi}^{m+1} =& \boldsymbol{\phi}^m + \rho(\mathbf{x}^{m+1} - \mathbf{w}^{m+1})
\end{aligned} \tag{5.15}$$

3. Set $\text{SLOCSA}(\mathbf{y}; \lambda, \mathbf{V}) = \mathbf{x}^*$, where $\mathbf{x}^*$ is the value of $\mathbf{x}$ at convergence.

5.2.1 Convergence of SLOCSA-ADMM-Algorithm

First we have the following lemma for SLSA problem and SLOCSA problem.

Lemma 2. *For any vector* $\mathbf{y}$ *and orthogonal constraint matrix* $\mathbf{V}$*, we have*

(a) The SLSA problem (3.15) is strictly convex, therefore the solution exists and is unique.

(b) The solution for SLOCSA problem (5.6) also exists and is unique.

(c) There exists $\mathbf{t}^* \in \mathbb{R}^k$ *such that* $SLOCSA(\mathbf{y}; \lambda, \mathbf{V}) = SLSA(\mathbf{y} + \mathbf{V}\mathbf{t}^*, \lambda)$*, and* $SLSA(\mathbf{y} + \mathbf{V}\mathbf{t}^*) \perp \mathbf{V}$*.*

Proof. **(a)** is trivial.

For **(b)**, denote the basis matrix of orthogonal complement of $\mathbf{V}$ as $\mathbf{V}^\perp$, then $\mathbf{x} \perp \mathbf{V} \Leftrightarrow \mathbf{x} = \mathbf{V}^\perp\mathbf{s}$, where $\mathbf{s} \in \mathbb{R}^{n-k}$, and the SLOCSA problem becomes:

$$\min_{\mathbf{s}\in\mathbb{R}^{n-k}} ||\mathbf{y} - \mathbf{V}^\perp\mathbf{s}||_2^2 + \lambda||\mathbf{V}^\perp\mathbf{s}||_1^2, \lambda > 0. \tag{5.16}$$

Since $\mathbf{V}^\perp$ has full column rank, we can see this problem is strictly convex, thus the solution exists and is unique, therefore the solution of (5.6) also exists and is unique.

For **(c)**, the KKT condition (note that our objective function is not differentiable, thus we are actually using the subdifferential version of KKT condition, see [18]) for problem (3.15) and (5.6) are

$$\mathbf{0} \in 2(\mathbf{x}^* - \mathbf{y}) + 2\lambda \cdot \mathbf{sgn}(\mathbf{x}^*), \tag{5.17}$$

and

$$\begin{aligned} \mathbf{0} \in & 2(\mathbf{x}^* - \mathbf{y}) + 2\lambda \cdot \mathbf{sgn}(\mathbf{x}^*) - 2\Sigma_{j=1}^{k} t_j^* \mathbf{m}_j \\ = & 2(\mathbf{x}^* - \mathbf{y} - \Sigma_{j=1}^{k} t_j^* \mathbf{m}_j) + 2\lambda \cdot \mathbf{sgn}(\mathbf{x}^*), \end{aligned} \tag{5.18}$$

respectively, where $\mathbf{V} = (\mathbf{m}_1, ..., \mathbf{m}_k)$. By comparing the KKT conditions, we can see $\mathbf{x}^*$ is also the solution of SLSA problem with $\mathbf{y} \leftarrow \mathbf{y} + \mathbf{V}\mathbf{t}^*$. Therefore we complete the proof. □

In [2], the authors listed two assumptions under which the convergence of ADMM can be guaranteed (see pages 16-17 of their paper).

Assumption 1. *The functions f and g are closed, proper, and convex. Or in other words, the epigraph* **epi***f and* **epi***g are closed nonempty convex sets.*

We can see our two functions $f(\mathbf{x}) = \lambda||\mathbf{x}||_1^2$ and $g(\mathbf{z}) = ||\mathbf{y} - \mathbf{V}^{\perp}\mathbf{z}||_2^2$ are continuous, proper and convex, thus this assumption can be easily satisfied.

Assumption 2. *The unaugmented Lagrangian L_0 has a saddle point. Or in other words, there exists $(\mathbf{x}^*, \mathbf{z}^*, \boldsymbol{\phi}^*)$, not necessarily unique, for which*

$$L_0(\mathbf{x}^*, \mathbf{z}^*, \boldsymbol{\phi}) \leq L_0(\mathbf{x}^*, \mathbf{z}^*, \boldsymbol{\phi}^*) \leq L_0(\mathbf{x}, \mathbf{z}, \boldsymbol{\phi}^*) \tag{5.19}$$

holds for all $\mathbf{x}$, $\mathbf{z}$, and $\boldsymbol{\phi}$.

One way to find a saddle point is to find a $\boldsymbol{\phi}^*$ such that there exists $(\mathbf{x}^*, \mathbf{z}^*)$ satisfying

$$(\mathbf{x}^*, \mathbf{z}^*) \in arg\min_{\mathbf{x},\mathbf{z}} L_0(\mathbf{x}, \mathbf{z}, \boldsymbol{\phi}^*),\ \mathbf{A}\mathbf{x}^* + \mathbf{B}\mathbf{z}^* = \mathbf{c}, \tag{5.20}$$

note that $arg\min_{\mathbf{x},\mathbf{z}} L_0(\mathbf{x}, \mathbf{z}, \boldsymbol{\phi}^*)$ may not be unique. Then $L_0(\mathbf{x}^*, \mathbf{z}^*, \boldsymbol{\phi}) = L_0(\mathbf{x}^*, \mathbf{z}^*, \boldsymbol{\phi}^*)$ because $\mathbf{A}\mathbf{x}^* + \mathbf{B}\mathbf{z}^* = \mathbf{c}$, and $L_0(\mathbf{x}^*, \mathbf{z}^*, \boldsymbol{\phi}^*) \leq L_0(\mathbf{x}, \mathbf{z}, \boldsymbol{\phi}^*)$, which is due to $(\mathbf{x}^*, \mathbf{z}^*) \in arg\min_{\mathbf{x},\mathbf{z}} L_0(\mathbf{x}, \mathbf{z}, \boldsymbol{\phi}^*)$.

Specified to our problem, we need to find $\boldsymbol{\phi}^*$ such that there exists $(\mathbf{x}^*, \mathbf{z}^*)$ satisfying

$$(\mathbf{x}^*, \mathbf{z}^*) \in arg\min_{\mathbf{x},\mathbf{z}} L_0(\mathbf{x}, \mathbf{z}, \boldsymbol{\phi}^*)\lambda||\mathbf{x}||_1^2 + ||\mathbf{y} - \mathbf{V}^\perp\mathbf{z}||_2^2 + (\boldsymbol{\phi}^*)^T(\mathbf{x} - \mathbf{V}^\perp\mathbf{z}) \tag{5.21}$$

and $\mathbf{x}^* = \mathbf{V}^\perp\mathbf{z}^*$.

Since

$$\begin{aligned} L_0(\mathbf{x}, \mathbf{z}, \boldsymbol{\phi}^*) =& \lambda||\mathbf{x}||_1^2 + ||\mathbf{y} - \mathbf{V}^\perp\mathbf{z}||_2^2 + (\boldsymbol{\phi}^*)^T(\mathbf{x} - \mathbf{V}^\perp\mathbf{z}) \\ =& (\lambda||\mathbf{x}||_1^2 + (\boldsymbol{\phi}^*)^T\mathbf{x}) + (||\mathbf{y} - \mathbf{V}^\perp\mathbf{z}||_2^2 - (\boldsymbol{\phi}^*)^T\mathbf{V}^\perp\mathbf{z}) \end{aligned} \tag{5.22}$$

and $arg\min_{\mathbf{z}} ||\mathbf{y} - \mathbf{V}^\perp\mathbf{z}||_2^2 - (\boldsymbol{\phi}^*)^T\mathbf{V}^\perp\mathbf{z}$ is unique and equals to $(\mathbf{V}^\perp)^T(\mathbf{y} + \boldsymbol{\phi}^*/2) = \mathbf{z}^*$, which implies $\mathbf{x}^* = \mathbf{V}^\perp\mathbf{z}^* = \mathbf{V}^\perp(\mathbf{V}^\perp)^T(\mathbf{y} + \boldsymbol{\phi}^*/2)$. Thus we need to look for $\boldsymbol{\phi}^* \in \mathbb{R}^n$ such that

$$\mathbf{V}^\perp(\mathbf{V}^\perp)^T(\mathbf{y} + \boldsymbol{\phi}^*/2) \in arg\min_{\mathbf{x}} \lambda||\mathbf{x}||_1^2 + (\boldsymbol{\phi}^*)^T\mathbf{x}. \tag{5.23}$$

Define $\boldsymbol{\xi} = \mathbf{y} + \boldsymbol{\phi}/2$, then $\boldsymbol{\phi} = -2(\mathbf{y} - \boldsymbol{\xi})$ and we need to look for $\boldsymbol{\xi}^* \in \mathbb{R}^n$ such that

$$\mathbf{V}^\perp(\mathbf{V}^\perp)^T\boldsymbol{\xi}^* \in arg\min_{\mathbf{x}} \lambda||\mathbf{x}||_1^2 - 2(\mathbf{y} - \boldsymbol{\xi}^*)^T\mathbf{x}. \tag{5.24}$$

Decompose $\boldsymbol{\xi} = \boldsymbol{\xi}_1 + \boldsymbol{\xi}_2$, with $\boldsymbol{\xi}_1 \in \mathbf{span}(\mathbf{V})$ and $\boldsymbol{\xi}_2 \in \mathbf{span}(\mathbf{V}^\perp)$, now it is equivalent to looking for $\boldsymbol{\xi}_1^* \in \mathbf{span}(\mathbf{V})$ and $\boldsymbol{\xi}_2^* \in \mathbf{span}(\mathbf{V}^\perp)$ such that

$$\mathbf{V}^\perp(\mathbf{V}^\perp)^T\boldsymbol{\xi}^* = \boldsymbol{\xi}_2^* \in arg\min_{\mathbf{x}} \lambda||\mathbf{x}||_1^2 - 2(\mathbf{y} - \boldsymbol{\xi}_1^* - \boldsymbol{\xi}_2^*)^T\mathbf{x}. \tag{5.25}$$

Note that the right hand side of (5.25) is a convex function, by looking at its subdifferential, we can see finding the saddle point is equivalent to looking for $\boldsymbol{\xi}_1^*$ and $\boldsymbol{\xi}_2^*$ such that

$$\mathbf{0} \in \left(2\lambda||\mathbf{x}||_1 \cdot \mathbf{sgn}(\mathbf{x}) - 2(\mathbf{y} - \boldsymbol{\xi}_1^* - \boldsymbol{\xi}_2^*)\right)\Big|_{\mathbf{x}=\boldsymbol{\xi}_2^*}, \tag{5.26}$$

or equivalently,

$$\mathbf{0} \in 2\lambda||\boldsymbol{\xi}_2^*||_1\mathbf{sgn}(\boldsymbol{\xi}_2^*) - 2(\mathbf{y} - \boldsymbol{\xi}_1^* - \boldsymbol{\xi}_2^*)\big). \tag{5.27}$$

Note that the right hand side is the subdifferential of $\lambda||\mathbf{x}||_1^2 + ||\mathbf{y} - \boldsymbol{\xi}_1^* - \mathbf{x}||_2^2$, where this function is strictly convex, thus finding the saddle point is equivalent to looking for $\boldsymbol{\xi}_1^* \in \mathbf{span}(\mathbf{V})$ and $\boldsymbol{\xi}_2^* \in \mathbf{span}(\mathbf{V}^\perp)$ such that

$$\boldsymbol{\xi}_2^* = arg\min_{\mathbf{x}} \lambda||\mathbf{x}||_1^2 + ||\mathbf{y} - \boldsymbol{\xi}_1^* - \mathbf{x}||_2^2, \tag{5.28}$$

According to result ($\mathbf{c}$) of Lemma 2, there exists $\mathbf{t}^* \in \mathbb{R}^k$ such that $\text{SLOCSA}(\mathbf{y}; \lambda, \mathbf{V}) = \text{SLSA}(\mathbf{y} + \mathbf{V}\mathbf{t}^*, \lambda)$ and $\text{SLSA}(\mathbf{y} + \mathbf{V}\mathbf{t}^*) \perp \mathbf{V}$. Note that $-\mathbf{V}\mathbf{t}^* \in \mathbf{span}(\mathbf{V})$ and

$\text{SLSA}(\mathbf{y}+\mathbf{V}\mathbf{t}^*) \perp \mathbf{V}$ thus belongs to $\mathbf{span}(\mathbf{V}^\perp)$, therefore the existence of $\boldsymbol{\xi}_1^*$ and $\boldsymbol{\xi}_2^*$ can be guaranteed, and so does the saddle point of $L_0(\mathbf{x}, \mathbf{z}, \boldsymbol{\phi})$.

When the two assumptions hold, the ADMM iterates satisfy $f(\mathbf{x}^k) + g(\mathbf{z}^k) \to p^*$ as $k \to \infty$, where $p^* = \inf\{f(\mathbf{x}) + g(\mathbf{z}) \mid \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{c}\}$. Therefore the convergence of our algorithm to the unique optima can be guaranteed.

We summarize the results above in the following theorem.

Theorem 4. *The **SLOCSA-ADMM-Algorithm** above converges to the unique global optima of SLOCSA problem (5.6).*

5.3 SLOCSA Using Quadratic Programming Algorithm

ADMM algorithm is parallel-computing friendly, however usually the convergence is slow. Therefore for small dimension cases, ADMM method is not competitive. In this section, we use quadratic programming method to solve SLOCSA problem.

First, by introducing augmented variables $\mathbf{x}_1$ and $\mathbf{x}_2$ as positive part and negative part of $\mathbf{x}$, we have $\mathbf{x} = \mathbf{x}_1 - \mathbf{x}_2$ and $||\mathbf{x}||_1 = \mathbf{1}_p^T|\mathbf{x}| = \mathbf{1}_p^T(\mathbf{x}_1 + \mathbf{x}_2)$, therefore we can transform the original SLOCSA problem (5.6) as follows:

$$\min_{\mathbf{x}_1,\, \mathbf{x}_2} ||\mathbf{x}_1 - \mathbf{x}_2 - \mathbf{y}||_2^2 + \lambda(\mathbf{1}_p^T(\mathbf{x}_1 + \mathbf{x}_2))^2, \tag{5.29}$$

sub to $\mathbf{x}_1 - \mathbf{x}_2 \perp \mathbf{V}$, $\mathbf{x}_1 \geq 0$, $\mathbf{x}_2 \geq 0$, and $\mathbf{x}_1$ has no common nonzero entry locations with $\mathbf{x}_2$.

For any coordinate i, if $(\mathbf{x}_1)_i > 0$ and $(\mathbf{x}_2)_i > 0$ (say $(\mathbf{x}_2)_i = c$ is smaller), then we could see that by subtracting c from both vectors, the objective function gets smaller value and the point is still feasible.

Therefore the solution of equation (5.29) must satisfy that "the nonzero entries for $\mathbf{x}_1$ and $\mathbf{x}_2$ have no common locations", therefore SLOCSA problem is also equivalent

to:

$$\begin{aligned} \min_{\mathbf{x}_1,\, \mathbf{x}_2} \; & ||\mathbf{x}_1 - \mathbf{x}_2 - \mathbf{y}||_2^2 + \lambda(\mathbf{1}_p^T(\mathbf{x}_1 + \mathbf{x}_2))^2 \\ & \text{sub to } \mathbf{x}_1 - \mathbf{x}_2 \perp \mathbf{V}, \mathbf{x}_1 \geq 0, \mathbf{x}_2 \geq 0. \end{aligned} \tag{5.30}$$

We could see that problem (5.30) is a positive semi-definite problem, to transfer it to a positive definite one, consider the following new problem:

$$\begin{aligned} \min_{\mathbf{x}_1,\, \mathbf{x}_2} \; & 2\mathbf{x}_1^T\mathbf{x}_2 + ||\mathbf{x}_1 - \mathbf{x}_2 - \mathbf{y}||_2^2 + \lambda(\mathbf{1}_p^T(\mathbf{x}_1 + \mathbf{x}_2))^2, \\ & \text{sub to } \mathbf{x}_1 - \mathbf{x}_2 \perp \mathbf{V}, \mathbf{x}_1 \geq 0, \mathbf{x}_2 \geq 0. \end{aligned} \tag{5.31}$$

Given that $\mathbf{x}_1 \geq 0, \mathbf{x}_2 \geq 0$, we know the solution of problem (5.30) minimizes $2\mathbf{x}_1^T\mathbf{x}_2$ (since its minimum is zero given the constraint, and it is zero at the solution of problem (5.30)), it also minimizes the rest part of the objective function for problem (5.31), so it is also the solution of problem (5.31). Therefore (5.6) $\Leftrightarrow$ (5.29) $\Leftrightarrow$ (5.30) $\Leftrightarrow$ (5.31).

To reform problem (5.31) as a standard positive definite programming problem, define $\mathbf{w} = (\mathbf{x}_1^T, \mathbf{x}_2^T)^T$, then we have the following problem:

$$\begin{aligned} \min_{\mathbf{w}} \; & \mathbf{w}^T(\mathbf{I}_{2p} + \lambda\mathbf{J}_{2p})\mathbf{w} - 2(\mathbf{y}^T, -\mathbf{y}^T)\mathbf{w}, \\ & \text{sub to } (\mathbf{V}^T, -\mathbf{V}^T)\mathbf{w} = \mathbf{0}, \mathbf{w} \geq 0. \end{aligned} \tag{5.32}$$

Then we can utilize standard algorithm for quadratic programming to solve SLOCSA. Those algorithms have been designed very efficient and the convergence can be guaranteed according to convergence results for quadratic programming (for example see [23]).

6. CONVERGENCE FOR THE ALTERNATING-DIRECTION ALGORITHM

Alternating-direction algorithm is a useful strategy when there exists more than one variables, and it has been used in many problems. Alternating-direction and coordinate-descent algorithm (e.g., algorithm for lasso) both belong to Block Coordinate Descent Algorithm (or BCD-Algorithm for short) family.

BCD is a family of widely used algorithms with many successful examples (such as described above). BCD-algorithm also has some failure cases, such as it cannot solve fussed lasso problem. The key thing is whether the function to be optimized is **regular** or not. In [21], the author showed some useful results on regularity and convergence. In this section, we make analysis, arrangement, and, more importantly, some useful improvements on its results. Finally we provide a unified approach for proving the convergence of regularized SVD problems, using our enhanced results, which of course include our sparse PCA method as example.

6.1 Introduction and Definitions

Definition 1. *For any function $f : \mathbb{R}^n \mapsto \mathbb{R}$, any point in its domain $\mathbf{x} \in dom(f)$, and any direction $\mathbf{d} \in \mathbb{R}^n$, the **lower directional derivative** of f at $\mathbf{x}$ in the direction $\mathbf{d}$ is defined as:*

$$f'(\mathbf{x}; \mathbf{d}) = \liminf_{\lambda \downarrow 0} \frac{f(\mathbf{x} + \lambda \mathbf{d}) - f(\mathbf{x})}{\lambda}. \tag{6.1}$$

We could see for any function lower directional derivative always exists, since lower limit always exists; also for any function f differentiable at $\mathbf{x}$ in direction $\mathbf{d}$, the lower directional derivative is the directional derivative.

Definition 2. *For any n-dimension coordinate system, a **block coordinate par-***

tition *is a partition of the coordinate system into several coordinate blocks, and is denoted as* $\pi = \{b_1, b_2, ..., b_r\}$*, where* b_i*'s are disjoint blocks that consist a partition of* $\{1, 2, ..., n\}$*.*

We say that a partition π_1 is **stronger** than another partition π_2, if π_2 can be obtained by combining some blocks in π_1. The strongest partition is $\pi = \{b_1, ..., b_n\}$ with $b_i = \{i\}$, $1 \leq i \leq n$, and the weakest partition is $\pi = \{b\}$ with $b = \{1, 2, ..., n\}$.

Definition 3. *We say that* $\mathbf{x}$ *is a* ***stationary point*** *of function* f*, if for any direction* $\mathbf{d}$ *we have* $f'(\mathbf{x}; \mathbf{d}) \geq 0$*.*

Note that this is equivalent to the usual definition of stationary point: "a stationary point is such that there exists a neighbor and it takes minimum in this neighbor".

Definition 4. *We say that a direction* $\mathbf{d}$ *is a* ***block coordinate direction*** *(or BC-direction for short) under partition* $\pi = \{b_1, ..., b_r\}$*, if it takes nonzero values in at most one coordinate block of* π*.*

Under the weakest block coordinate partition, any direction is a block coordinate direction; while under the strongest partition, a direction is a block coordinate direction only if it is the direction along one coordinate axis.

Definition 5. *We say* $\mathbf{x}$ *is a* ***(block) coordinatewise minimum point*** *(or BCM point for short) of function* f *under partition* π*, if for any BC-direction* $\mathbf{d}$ *we have* $f'(\mathbf{x}; \mathbf{d}) \geq 0$*.*

Note that a BCM point under a weaker partition π_1 is also a BCM point under a stronger partition π_2. A stationary point is exactly a BCM point under weakest partition, and thus is BCM point under any partition. Also note that this definition

is slightly different from the one in [21], in which the author's definition is a global one (equation (4) in his paper), and thus not good enough, since, analytically, stationarity should be a local property. Therefore according to their definition, even a stationary point may not necessarily be a BCM point.

Usually when we perform a BCD-Algorithm, the converging point is a BCM point. Our goal is to make it a stationary point, i.e., we want to fill the gap between the two concepts (stationarity vs. BCM stationarity). The extra property filling this gap is **regularity**:

Definition 6. *We say a function f is **weak regular** at point $\mathbf{x}$ under partition π, if stationarity is equivalent to block coordinate minimality under π at this point, or mathematically,*

$$f'(\mathbf{x};\mathbf{d}) \geq 0, \forall \text{ direction } \mathbf{d} \Leftrightarrow f'(\mathbf{x};\mathbf{d}) \geq 0, \forall \text{ BC-direction } \mathbf{d} \text{ under } \pi. \tag{6.2}$$

Also, we say a function is weak regular, if it is weak regular at every point in its domain.

Note that the definition of regularity used by [21] is weaker than the one in A. Auslender (1976), in which the author first introduced decomposition of a direction:

Definition 7. *Any direction $\mathbf{d}$ can be decomposed uniquely as a summation of BC-directions under a partition π,*

$$\mathbf{d} = \Sigma_{i=1}^{r} \mathbf{d}_i, \tag{6.3}$$

*where $\mathbf{d}_i$ takes the same value as $\mathbf{d}$ in i-th coordinate block of π and takes zero values in all other blocks. $\Sigma_{i=1}^{r}\mathbf{d}_i$ is called **block coordinate direction decomposition** of direction $\mathbf{d}$ under π.*

Then the regularity in [21] can be defined as:

Definition 8. *A function f is **strong regular** at point $\mathbf{x}$ under partition π, if*

$$f'(\mathbf{x};\mathbf{d}) = \Sigma_{i=1}^{r} f'(\mathbf{x};\mathbf{d}_i), \forall \text{ direction } \mathbf{d}, \tag{6.4}$$

where $\Sigma_{i=1}^{r}\mathbf{d}_i$ is block coordinate direction decomposition of $\mathbf{d}$ under π.

Note that if f is strong regular at point $\mathbf{x}$ under π, then it is also weak regular at $\mathbf{x}$ (under π), that's why we use "strong" and "weak" for the two concepts.

However, further investigation on these two concepts shows that (i): strong regularity is "too strong to transmit" (for example in Lemma (5) we find that even summation of marginal functions are not strong regular, given that any marginal function is strong regular); (ii) weak regularity is "too weak to take summation" (for example in Lemma (3) we find that summation of a differentiable function and a weak regular function may not be a weak regular function). Based on this consideration, we propose a new definition on regularity:

Definition 9. *A function f is **standard regular** at point $\mathbf{x}$ under partition π, if*

$$f'(\mathbf{x};\mathbf{d}) \geq \Sigma_{i=1}^{r} f'(\mathbf{x};\mathbf{d}_i), \forall \text{ direction } \mathbf{d}, \tag{6.5}$$

where $\Sigma_{i=1}^{r}\mathbf{d}_i$ is block coordinate direction decomposition of $\mathbf{d}$ under π.

We could easily see that "weak regularity < standard regularity < strong regularity", thus all three property could guarantee a BCM point to be a stationary point. In addition, we have the following results on additivity and scalar multiplicity:

Lemma 3. *Suppose f_1 and f_2 are two functions defined on the same domain, π is a partition,*

(a) ***(weak additivity)*** *If f_1 is differentiable, f_2 is standard (strong) regular under π, then $f_1 + f_2$ is also standard (strong) regular under π.*

(b) ***(nonnegative scalar multiplicity)*** *If f is standard (strong, weak) regular under π, c is a nonnegative scalar, then $c \cdot f$ is also standard (strong, weak) regular under π.*

Proof. All results follow the property of lower limit. We only need to note that for any functions $L_1(\lambda)$, $L_2(\lambda)$, and nonnegative scalar c,

$$\liminf_{\lambda \downarrow 0} \big(L_1(\lambda) + L_2(\lambda)\big) \geq \liminf_{\lambda \downarrow 0} L_1(\lambda) + \liminf_{\lambda \downarrow 0} L_2(\lambda), \tag{6.6}$$

$$\liminf_{\lambda \downarrow 0} \big(L_1(\lambda) + L_2(\lambda)\big) = \lim_{\lambda \downarrow 0} L_1(\lambda) + \liminf_{\lambda \downarrow 0} L_2(\lambda), \text{ if } \lim_{\lambda \downarrow 0} L_1(\lambda) \text{ exists}, \tag{6.7}$$

and

$$\liminf_{\lambda \downarrow 0} c \cdot L_1(\lambda) = c \cdot \liminf_{\lambda \downarrow 0} L_1(\lambda), \tag{6.8}$$

(a):

Denote $g = f_1 + f_2$, then for any direction $\mathbf{d} = (\mathbf{d}_{(1)}^T, ..., \mathbf{d}_{(r)}^T)^T = \Sigma_{i=1}^r \mathbf{d}_i$,

$$\begin{aligned} g'(\mathbf{x}; \mathbf{d}) =& \liminf_{\lambda \downarrow 0} \frac{g(\mathbf{x} + \lambda \mathbf{d}) - g(\mathbf{x})}{\lambda} \\ =& \liminf_{\lambda \downarrow 0} \big[\frac{f_1(\mathbf{x} + \lambda \mathbf{d}) - f_1(\mathbf{x})}{\lambda} + \frac{f_2(\mathbf{x} + \lambda \mathbf{d}) - f_2(\mathbf{x})}{\lambda}\big] \\ =& \lim_{\lambda \downarrow 0} \frac{f_1(\mathbf{x} + \lambda \mathbf{d}) - f_1(\mathbf{x})}{\lambda} + \liminf_{\lambda \downarrow 0} \frac{f_2(\mathbf{x} + \lambda \mathbf{d}) - f_2(\mathbf{x})}{\lambda} (\text{due to (6.7)}) \\ =& f_1'(\mathbf{x}; \mathbf{d}) + f_2'(\mathbf{x}; \mathbf{d}). \end{aligned} \tag{6.9}$$

Then if f_2 is standard regular, $g'(\mathbf{x}; \mathbf{d}) = f_1'(\mathbf{x}; \mathbf{d}) + f_2'(\mathbf{x}; \mathbf{d}) \geq \Sigma_{i=1}^r f_1'(\mathbf{x}; \mathbf{d}_i) +$

$\Sigma_{i=1}^r f_2'(\mathbf{x};\mathbf{d}_i) = \Sigma_{i=1}^r g'(\mathbf{x};\mathbf{d}_i)$.

If f_2 is strong regular, $g'(\mathbf{x};\mathbf{d}) = f_1'(\mathbf{x};\mathbf{d}) + f_2'(\mathbf{x};\mathbf{d}) = \Sigma_{i=1}^r f_1'(\mathbf{x};\mathbf{d}_i) + \Sigma_{i=1}^r f_2'(\mathbf{x};\mathbf{d}_i) = \Sigma_{i=1}^r g'(\mathbf{x};\mathbf{d}_i)$.

(b):

We denote $g = c \cdot f$, if $c = 0$, then $g = 0$, which is differentiable thus strong regular, therefore we assume $c > 0$. For any direction $\mathbf{d} = (\mathbf{d}_{(1)}^T, ..., \mathbf{d}_{(r)}^T)^T = \Sigma_{i=1}^r \mathbf{d}_i$,

$$\begin{aligned} g'(\mathbf{x};\mathbf{d}) =& \liminf_{\lambda \downarrow 0} \frac{g(\mathbf{x}+\lambda\mathbf{d}) - g(\mathbf{x})}{\lambda} \\ =& \liminf_{\lambda \downarrow 0} c \cdot \frac{f(\mathbf{x}+\lambda\mathbf{d}) - f(\mathbf{x})}{\lambda} \\ =& c \cdot \liminf_{\lambda \downarrow 0} \frac{f(\mathbf{x}+\lambda\mathbf{d}) - f(\mathbf{x})}{\lambda} (\text{due to (6.8)}) \\ =& c \cdot f'(\mathbf{x};\mathbf{d}). \end{aligned} \tag{6.10}$$

Then if f is standard regular, $g'(\mathbf{x};\mathbf{d}) = c \cdot f'(\mathbf{x};\mathbf{d}) \geq c \cdot \Sigma_{i=1}^r f'(\mathbf{x};\mathbf{d}_i) = \Sigma_{i=1}^r g'(\mathbf{x};\mathbf{d}_i)$.

If f is strong regular, $g'(\mathbf{x};\mathbf{d}) = c \cdot f'(\mathbf{x};\mathbf{d}) = c \cdot \Sigma_{i=1}^r f'(\mathbf{x};\mathbf{d}_i) = \Sigma_{i=1}^r g'(\mathbf{x};\mathbf{d}_i)$.

If f is weak regular,

$$\begin{aligned} & g'(\mathbf{x};\mathbf{d}) \geq 0 (\forall \text{ BC-direction } \mathbf{d}) \\ \Rightarrow & f'(\mathbf{x};\mathbf{d}) = g'(\mathbf{x};\mathbf{d})/c \geq 0 (\forall \text{ BC-direction } \mathbf{d}) \\ \Rightarrow & f'(\mathbf{x};\mathbf{d}) \geq 0 (\forall \text{ direction } \mathbf{d}) \\ \Rightarrow & g'(\mathbf{x};\mathbf{d}) = f'(\mathbf{x};\mathbf{d}) \cdot c \geq 0 (\forall \text{ direction } \mathbf{d}). \end{aligned} \tag{6.11}$$

□

Remark 2. *We cannot obtain any additivity results on regularity, because if f_1 and f_2 are strong regular, $f_1 + f_2$ even may not be weak regular.*

For block coordinate minimality and partition, we have the following lemma:

Lemma 4. *Suppose π_1 and π_2 are two block coordinate partitions, π_1 is weaker than π_2,*

(a) If $\mathbf{x}$ is a BCM point under partition π_1, then it is also a BCM point under π_2.

(b) If function f is standard (strong, weak) regular at $\mathbf{x}$ under π_2, then it is also standard (strong, weak) regular at $\mathbf{x}$ under π_1.

One way to understand this result is that we always have

$$\text{BCM} + \text{Regularity} = \text{Stationarity}, \tag{6.12}$$

where "BCM" stands for block coordinatewise minimality here. When the partition gets stronger (↑), the regularity gets stronger (↑) and BCM gets weaker (↓), and makes the summation being constant. One extreme case is, for weakest partition, BCM becomes Stationarity, while Regularity becomes "zero" (thus any function is standard (strong, weak) regular under weakest partition). On the other hand, for lasso problem

$$\min_{\beta \in \mathbb{R}^p} ||\mathbf{y} - X\boldsymbol{\beta}||_2^2 + \lambda ||\boldsymbol{\beta}||_1, \tag{6.13}$$

people have shown that this objective function is weak regular under strongest partition, so any coordinatewise (not block coordinatewise) minimum point is stationary point, and we can use coordinate descent algorithm to solve it.

6.2 Results on Regularity

We propose a systematic results on (standard, strong, and weak) regularity in the following lemma.

Lemma 5. *Suppose π is a block coordinate partition of $\mathbb{R}^n$, then we have the following results:*

(a) Any differentiable function is strong regular (thus standard and weak regular) under strongest partition; any function is strong regular under weakest partition; any one-dimensional function is strong regular (only one possible partition).

(b) Any marginal function f of π is strong regular under π, where a marginal function f of partition π is a function of at most one coordinate block of π.

(c) If $g : \mathbb{R}^1 \mapsto \mathbb{R}^1$ is differentiable and has $g'(\cdot) \geq 0$ on its domain, $h : \mathbb{R}^n \mapsto \mathbb{R}^1$ is a continuous function and is standard (strong, weak) regular under π, then $f(\cdot) = g(h(\cdot))$ is also standard (strong, weak) regular under π.

(d) Suppose $\pi_0 = \{b_1, ..., b_r\}$ is a partition of $\mathbb{R}^n$ with b_i having s_i coordinates ($\Sigma_{i=1}^r s_i = n$), any point $\mathbf{x}$ can be written as $\mathbf{x} = (\mathbf{x}_{(1)}^T, ..., \mathbf{x}_{(r)}^T)^T$ under π_0. $\{h_i : \mathbb{R}^{s_i} \mapsto \mathbb{R}^1, 1 \leq i \leq r\}$ are r continuous functions, and are standard regular under partitions $\{\pi_i : 1 \leq i \leq r\}$, respectively. For consistency, π_i is partition of set $\{S_{i-1} + 1, ..., S_i\}$, where $S_i = \Sigma_{j=1}^i s_i$. If $g : \mathbb{R}^r \mapsto \mathbb{R}^1$ is differentiable, and all r partial derivative functions are continuous and nonnegative, then $f(\mathbf{x}) = f(\mathbf{x}_{(1)}, ..., \mathbf{x}_{(r)}) = g(h_1(\mathbf{x}_{(1)}), ..., h_r(\mathbf{x}_{(r)}))$ is standard regular under $\bar{\pi}$, where $\bar{\pi} = \bigcup_{i=1}^r \pi_i$ is a partition of $\mathbb{R}^n$ stronger than π_0.

Besides, if h_i's are strong regular, we get f is standard regular (not strong regular); if h_i's are weak regular, we get f is weak regular.

$\bar{\pi}$ is a stronger partition than π_0, e.g., when $n = 6$, $\pi_0 = \big\{\{1, 2, 3\}, \{4, 5, 6\}\big\}$, $\pi_1 = \big\{\{1, 2, 3\}\big\}$, $\pi_2 = \big\{\{4, 5\}, \{6\}\big\}$, then $\bar{\pi} = \big\{\{1, 2, 3\}, \{4, 5\}, \{6\}\big\}$.

Proof. The requirement of nonnegative derivatives in **(c)** and **(d)** is due to the fact

$$\liminf_{\lambda\downarrow 0} L_1(\lambda)\cdot L_2(\lambda) = \lim_{\lambda\downarrow 0} L_1(\lambda)\cdot \liminf_{\lambda\downarrow 0} L_2(\lambda),\ \text{if}\ \lim_{\lambda\downarrow 0} L_1(\lambda)\geq 0. \tag{6.14}$$

(a):

Suppose f is differentiable, then its lower directional derivatives are all directional derivatives, and we have the total derivative decomposition

$$f'(\mathbf{x};\mathbf{d}) = \Sigma_{i=1}^{n} f'(\mathbf{x};d_i), \tag{6.15}$$

where d_i is the i-th coordinate of $\mathbf{d}$. Therefore f is strong regular under strongest partition. The other two results are trivial.

(b):

Suppose f is a function of k-th block of π, i.e., $f(\mathbf{x}) = g(\mathbf{x}_{(k)})$. Then for any direction $\mathbf{d} = (\mathbf{d}_{(1)}^T,...,\mathbf{d}_{(r)}^T)^T = \Sigma_{i=1}^{r}\mathbf{d}_i$,

$$\begin{aligned} f'(\mathbf{x};\mathbf{d}) =& \liminf_{\lambda\downarrow 0} \frac{f(\mathbf{x}+\lambda\mathbf{d})-f(\mathbf{x})}{\lambda} \\ =& \liminf_{\lambda\downarrow 0} \frac{g(\mathbf{x}_{(k)}+\lambda\mathbf{d}_{(k)})-g(\mathbf{x}_{(k)})}{\lambda} \\ =& g'(\mathbf{x}_{(k)};\mathbf{d}_{(k)}), \end{aligned} \tag{6.16}$$

thus for $i\neq k$, $f'(\mathbf{x};\mathbf{d}_i)=0$, and for $i=k$, $f'(\mathbf{x};\mathbf{d}_i)=f'(\mathbf{x};\mathbf{d})$. Therefore we have $f'(\mathbf{x};\mathbf{d}) = \Sigma_{i=1}^{r} f'(\mathbf{x};\mathbf{d}_i)$, and thus strong regularity holds.

(c):

For any point $\mathbf{x} \in \mathrm{dom}(f)$ and any direction $\mathbf{d}$, we have

$$\begin{aligned}
&f'(\mathbf{x};\mathbf{d}) \\
=&\liminf_{\lambda\downarrow 0}\frac{f(\mathbf{x}+\lambda\mathbf{d})-f(\mathbf{x})}{\lambda} \\
=&\liminf_{\lambda\downarrow 0}\frac{g(h(\mathbf{x}+\lambda\mathbf{d}))-g(h(\mathbf{x}))}{\lambda} \\
=&\liminf_{\lambda\downarrow 0}\frac{g(h(\mathbf{x}+\lambda\mathbf{d}))-g(h(\mathbf{x}))}{h(\mathbf{x}+\lambda\mathbf{d})-h(\mathbf{x})}\cdot\frac{h(\mathbf{x}+\lambda\mathbf{d})-h(\mathbf{x})}{\lambda} \\
=&\lim_{\lambda\downarrow 0}\frac{g(h(\mathbf{x}+\lambda\mathbf{d}))-g(h(\mathbf{x}))}{h(\mathbf{x}+\lambda\mathbf{d})-h(\mathbf{x})}\cdot\liminf_{\lambda\downarrow 0}\frac{h(\mathbf{x}+\lambda\mathbf{d})-h(\mathbf{x})}{\lambda}(\text{ due to (6.14)}) \\
=&g'(h(\mathbf{x}))\cdot h'(\mathbf{x};\mathbf{d}),
\end{aligned} \tag{6.17}$$

where the last two steps hold, since $h(\cdot)$ is continuous and $g(\cdot)$ has nonnegative derivative.

For any point $\mathbf{x}$ and any direction $\mathbf{d} = (\mathbf{d}_{(1)}^T, ..., \mathbf{d}_{(r)}^T)^T = \Sigma_{i=1}^r \mathbf{d}_i$, if $g'(h(\mathbf{x})) = 0$, then $f'(\mathbf{x};\mathbf{d}) = 0$ always holds, thus f is strong regular at point $\mathbf{x}$, therefore we assume $g'(h(\mathbf{x})) > 0$.

(1) If h is standard regular, then $f'(\mathbf{x};\mathbf{d}) = g'(h(\mathbf{x}))\cdot h'(\mathbf{x};\mathbf{d}) \geq g'(h(\mathbf{x}))\cdot \Sigma_{i=1}^r h'(\mathbf{x};\mathbf{d}_i) = \Sigma_{i=1}^r g'(\mathbf{x};\mathbf{d}_i)$.

(2) If h is strong regular, then $f'(\mathbf{x};\mathbf{d}) = g'(h(\mathbf{x}))\cdot h'(\mathbf{x};\mathbf{d}) = g'(h(\mathbf{x}))\cdot\Sigma_{i=1}^r h'(\mathbf{x};\mathbf{d}_i) = \Sigma_{i=1}^r g'(\mathbf{x};\mathbf{d}_i)$.

(3) If h is weak regular, then

$$\begin{aligned}
&f'(\mathbf{x};\mathbf{d}) \geq 0, \forall \text{ BC-direction } \mathbf{d} \\
\Rightarrow &h'(\mathbf{x};\mathbf{d}) = f'(\mathbf{x};\mathbf{d})/g'(h(\mathbf{x})) \geq 0, \forall \text{ BC-direction } \mathbf{d} \\
\Rightarrow &h'(\mathbf{x};\mathbf{d}) \geq 0, \forall \text{ direction } \mathbf{d} \\
\Rightarrow &f'(\mathbf{x};\mathbf{d}) = g'(\mathbf{x};\mathbf{d})\cdot g'(h(\mathbf{x})) \geq 0, \forall \text{ direction } \mathbf{d}
\end{aligned} \tag{6.18}$$

(d):

We denote the partial derivative function of $g(\cdot)$ w.r.t. i-th coordinate as $g_i'(\cdot)$, $1 \leq i \leq r$.

For any point $\mathbf{x}$ and any direction $\mathbf{d} = (\mathbf{d}_{(1)}^T, ..., \mathbf{d}_{(r)}^T)^T = \Sigma_{i=1}^r \mathbf{d}_i$, we have:

$$
\begin{aligned}
& f'(\mathbf{x}; \mathbf{d}_i) = \liminf_{\lambda \downarrow 0} \frac{f(\mathbf{x} + \lambda \mathbf{d}_i) - f(\mathbf{x})}{\lambda} \\
=& \liminf_{\lambda \downarrow 0} \frac{g(h_1(\mathbf{x}_{(1)}), ..., h_i(\mathbf{x}_{(i)} + \lambda \mathbf{d}_{(i)}), ..., h_r(\mathbf{x}_{(r)})) - g(h_1(\mathbf{x}_{(1)}), ..., h_r(\mathbf{x}_{(r)}))}{\lambda} \\
=& \liminf_{\lambda \downarrow 0} \frac{g(h_1(\mathbf{x}_{(1)}), ..., h_i(\mathbf{x}_{(i)} + \lambda \mathbf{d}_{(i)}), ..., h_r(\mathbf{x}_{(r)})) - g(h_1(\mathbf{x}_{(1)}), ..., h_r(\mathbf{x}_{(r)}))}{h_i(\mathbf{x}_{(i)} + \lambda \mathbf{d}_{(i)}) - h_i(\mathbf{x}_{(i)})} \\
& \quad \cdot \frac{h_i(\mathbf{x}_{(i)} + \lambda \mathbf{d}_{(i)}) - h_i(\mathbf{x}_{(i)})}{\lambda} \\
=& \lim_{\lambda \downarrow 0} \frac{g(h_1(\mathbf{x}_{(1)}), ..., h_i(\mathbf{x}_{(i)} + \lambda \mathbf{d}_{(i)}), ..., h_r(\mathbf{x}_{(r)})) - g(h_1(\mathbf{x}_{(1)}), ..., h_r(\mathbf{x}_{(r)}))}{h_i(\mathbf{x}_{(i)} + \lambda \mathbf{d}_{(i)}) - h_i(\mathbf{x}_{(i)})} \\
& \quad \cdot \liminf_{\lambda \downarrow 0} \frac{h_i(\mathbf{x}_{(i)} + \lambda \mathbf{d}_{(i)}) - h_i(\mathbf{x}_{(i)})}{\lambda} \\
=& g_i'(h_1(\mathbf{x}_{(1)}), ..., h_r(\mathbf{x}_{(r)})) \cdot h_i'(\mathbf{x}_{(i)}; \mathbf{d}_{(i)}).
\end{aligned}
\tag{6.19}
$$

where the last two steps hold, since $h_i(\cdot)$ is continuous and $g_i'(\cdot)$ is nonnegative.

For direction $\mathbf{d}$, we have

$$
\begin{aligned}
&f'(\mathbf{x};\mathbf{d}) = \liminf_{\lambda\downarrow 0} \frac{f(\mathbf{x}+\lambda\mathbf{d}) - f(\mathbf{x})}{\lambda} \\
=&\liminf_{\lambda\downarrow 0} \frac{g(h_1(\mathbf{x}_{(1)}+\lambda\mathbf{d}_{(1)}),...,h_r(\mathbf{x}_{(r)}+\lambda\mathbf{d}_{(r)}))) - g(h_1(\mathbf{x}_{(1)}),...,h_r(\mathbf{x}_{(r)}))}{\lambda} \\
=&\liminf_{\lambda\downarrow 0} \\
&\Big[\frac{g(h_1(\mathbf{x}_{(1)}+\lambda\mathbf{d}_{(1)}),...,h_r(\mathbf{x}_{(r)}+\lambda\mathbf{d}_{(r)})) - g(h_1(\mathbf{x}_{(1)}),...,h_r(\mathbf{x}_{(r)}+\lambda\mathbf{d}_{(r)}))}{\lambda} + \\
&... + \frac{g(h_1(\mathbf{x}_{(1)}),...,h_{r-1}(\mathbf{x}_{(r-1)}),h_r(\mathbf{x}_{(r)}+\lambda\mathbf{d}_{(r)})) - g(h_1(\mathbf{x}_{(1)}),...,h_r(\mathbf{x}_{(r)}))}{\lambda}\Big] \\
\doteq&\liminf_{\lambda\downarrow 0} \big(T_1(\lambda) + ... + T_r(\lambda)\big) \\
\geq&\liminf_{\lambda\downarrow 0} T_1(\lambda) + ... + \liminf_{\lambda\downarrow 0} T_r(\lambda)(\text{ due to (6.6)}).
\end{aligned}
\tag{6.20}
$$

For the i-th term $\liminf_{\lambda\downarrow 0} T_i(\lambda)$, we have

$$
\begin{aligned}
&\liminf_{\lambda\downarrow 0} T_i(\lambda) \\
=&\liminf_{\lambda\downarrow 0}[g(h_1(\mathbf{x}_{(1)}),...,h_{i-1}(\mathbf{x}_{(i-1)}),h_i(\mathbf{x}_{(i)}+\lambda\mathbf{d}_{(i)}),...,h_r(\mathbf{x}_{(r)}+\lambda\mathbf{d}_{(r)})) \\
&- g(h_1(\mathbf{x}_{(1)}),...,h_i(\mathbf{x}_{(i)}),h_{i+1}(\mathbf{x}_{(i+1)}+\lambda\mathbf{d}_{(i+1)}),...,h_r(\mathbf{x}_{(r)}+\lambda\mathbf{d}_{(r)}]/\lambda \\
=&\liminf_{\lambda\downarrow 0}[(g(h_1(\mathbf{x}_{(1)}),...,h_{i-1}(\mathbf{x}_{(i-1)}),h_i(\mathbf{x}_{(i)}+\lambda\mathbf{d}_{(i)}),...,h_r(\mathbf{x}_{(r)}+\lambda\mathbf{d}_{(r)})) \\
&- g(h_1(\mathbf{x}_{(1)}),...,h_i(\mathbf{x}_{(i)}),h_{i+1}(\mathbf{x}_{(i+1)}+\lambda\mathbf{d}_{(i+1)}),...,h_r(\mathbf{x}_{(r)}+\lambda\mathbf{d}_{(r)}) \\
&/(h_i(\mathbf{x}_{(i)}+\lambda\mathbf{d}_{(i)}) - h_i(\mathbf{x}_{(i)}))] \cdot [(h_i(\mathbf{x}_{(i)}+\lambda\mathbf{d}_{(i)}) - h_i(\mathbf{x}_{(i)}))/\lambda] \\
=&g_i'(h_1(\mathbf{x}_{(1)}),...,h_r(\mathbf{x}_{(r)})) \cdot h_i'(\mathbf{x}_{(i)};\mathbf{d}_{(i)}).
\end{aligned}
\tag{6.21}
$$

where the last step holds, since $h_i(\cdot)$ is continuous, $g_i'(\cdot)$ is continuous and nonnegative.

Therefore we have

$$f'(\mathbf{x};\mathbf{d}) \geq \Sigma_{i=1}^{r} g_i'(h_1(\mathbf{x}_{(1)}),...,h_r(\mathbf{x}_{(r)})) \cdot h_i'(\mathbf{x}_{(i)};\mathbf{d}_{(i)}). \tag{6.22}$$

Suppose $\mathbf{d}_i$ has decomposition $\mathbf{d}_i = \Sigma_{i=1}^{r_i} \mathbf{e}_{ij}$ under partition π_i, or equivalently, $\mathbf{d}_{(i)} = \Sigma_{i=1}^{r_i} \mathbf{e}_{(i)j}$, then $\mathbf{d} = \Sigma_{i=1}^{r} \Sigma_{j=1}^{r_i} \mathbf{e}_{ij}$.

(1) If h_i's are standard regular under π_i, then

$$\begin{aligned} & f'(\mathbf{x};\mathbf{d}) \\ \geq & \Sigma_{i=1}^{r} g_i'(h_1(\mathbf{x}_{(1)}),...,h_r(\mathbf{x}_{(r)})) \cdot h_i'(\mathbf{x}_{(i)};\mathbf{d}_{(i)}) \\ \geq & \Sigma_{i=1}^{r} g_i'(h_1(\mathbf{x}_{(1)}),...,h_r(\mathbf{x}_{(r)})) \cdot \left(\Sigma_{j=1}^{r_i} h_i'(\mathbf{x}_{(i)};\mathbf{e}_{(i)j})\right) \\ = & \Sigma_{i=1}^{r} \Sigma_{j=1}^{r_i} g_i'(h_1(\mathbf{x}_{(1)}),...,h_r(\mathbf{x}_{(r)})) \cdot h_i'(\mathbf{x}_{(i)};\mathbf{e}_{(i)j}) \end{aligned} \tag{6.23}$$

Note that $\mathbf{e}_{ij}$ is also a BC-direction of π_0, thus according to (6.19), we have

$$f'(\mathbf{x};\mathbf{e}_{ij}) = g_i'(h_1(\mathbf{x}_{(1)}),...,h_r(\mathbf{x}_{(r)})) \cdot h_i'(\mathbf{x}_{(i)};\mathbf{e}_{(i)j}). \tag{6.24}$$

Plug this result into (6.23), we get $f'(\mathbf{x};\mathbf{d}) \geq \Sigma_{i=1}^{r} \Sigma_{j=1}^{r_i} f'(\mathbf{x};\mathbf{e}_{ij})$, thus f is standard regular under $\bar{\pi}$.

(2) If h_i's are weak regular under π_i, then suppose $\mathbf{x}$ is a BCM point under $\bar{\pi}$, i.e., $f'(\mathbf{x};\mathbf{e}_{ij}) \geq 0$, $\forall$ BC-direction $\mathbf{e}_{ij}$ under $\bar{\pi}$.

(i) If $g_i'(h_1(\mathbf{x}_{(1)}),...,h_r(\mathbf{x}_{(r)})) = 0$, then $g_i'(h_1(\mathbf{x}_{(1)}),...,h_r(\mathbf{x}_{(r)})) \cdot h_i'(\mathbf{x}_{(i)};\mathbf{d}_{(i)}) = 0 \geq 0$.

(ii) If $g_i'(h_1(\mathbf{x}_{(1)}), ..., h_r(\mathbf{x}_{(r)})) > 0$, then

$$\begin{aligned}
&f'(\mathbf{x}; \mathbf{e}_{ij}) = g_i'(h_1(\mathbf{x}_{(1)}), ..., h_r(\mathbf{x}_{(r)})) \cdot h_i'(\mathbf{x}_{(i)}; \mathbf{e}_{(i)j}) \geq 0, \\
&(\forall \text{ BC-direction } \mathbf{e}_{ij} \text{ under } \bar{\pi}, \text{ or } \forall \text{ BC-direction } \mathbf{e}_{(i)j} \text{ under } \pi_i) \\
\Rightarrow &h_i'(\mathbf{x}_{(i)}; \mathbf{e}_{(i)j}) \geq 0, \forall \text{ BC-direction } \mathbf{e}_{(i)j} \text{ under } \pi_i \\
\Rightarrow &h_i'(\mathbf{x}_{(i)}; \mathbf{d}_{(i)}) \geq 0, \forall \text{ direction } \mathbf{d}_{(i)} \text{ of } \mathbb{R}^{s_i} \text{ (due to weak regularity of } h_i) \\
\Rightarrow &g_i'(h_1(\mathbf{x}_{(1)}), ..., h_r(\mathbf{x}_{(r)})) \cdot h_i'(\mathbf{x}_{(i)}; \mathbf{d}_{(i)}) \geq 0, \forall \text{ direction } \mathbf{d}_{(i)} \text{ of } \mathbb{R}^{s_i}.
\end{aligned} \tag{6.25}$$

Under either (i) or (ii), we have $g_i'(h_1(\mathbf{x}_{(1)}), ..., h_r(\mathbf{x}_{(r)})) \cdot h_i'(\mathbf{x}_{(i)}; \mathbf{d}_{(i)}) \geq 0$, thus $f'(\mathbf{x}; \mathbf{d}) \geq \Sigma_{i=1}^r g_i'(h_1(\mathbf{x}_{(1)}), ..., h_r(\mathbf{x}_{(r)})) \cdot h_i'(\mathbf{x}_{(i)}; \mathbf{d}_{(i)}) \geq 0$, therefore f is weak regular at any point $\mathbf{e}$ under $\bar{\pi}$, and we complete the proof of **(d)**.

□

Remark 3. *In both **(c)** and **(d)**, we assume function(s) $h(\cdot)$ (or $h_i(\cdot)$) to be continuous, this is needed in term*

$$\lim_{\lambda \downarrow 0} \frac{g(h(\mathbf{x} + \lambda \mathbf{d})) - g(h(\mathbf{x}))}{h(\mathbf{x} + \lambda \mathbf{d}) - h(\mathbf{x})},$$

*besides, **(d)** also assumes $g_i'(\cdot)$ to be continuous, this is needed in term*

$$\begin{aligned}
&\liminf_{\lambda \downarrow 0} [(g(h_1(\mathbf{x}_{(1)}), ..., h_{i-1}(\mathbf{x}_{(i-1)}), h_i(\mathbf{x}_{(i)} + \lambda \mathbf{d}_{(i)}), ..., h_r(\mathbf{x}_{(r)} + \lambda \mathbf{d}_{(r)})) \\
&- g(h_1(\mathbf{x}_{(1)}), ..., h_i(\mathbf{x}_{(i)}), h_{i+1}(\mathbf{x}_{(i+1)} + \lambda \mathbf{d}_{(i+1)}), ..., h_r(\mathbf{x}_{(r)} + \lambda \mathbf{d}_{(r)}) \\
&/(h_i(\mathbf{x}_{(i)} + \lambda \mathbf{d}_{(i)}) - h_i(\mathbf{x}_{(i)}))],
\end{aligned}$$

*while **(b)** does not need this assumption, because in one-dimensional case, it does not have this decomposition.*

Remark 4. *We cannot obtain strong regularity of function f in **(d)**, because in*

equation (6.22) there occurs inequality, further, this is due to the property of lower limit (6.6).

Remark 5. *"The micro essence of differentiability is linearity", therefore from (6.17) we can see* ***(c)*** *actually corresponds to "nonnegative scalar multiplicity", thus there is nothing strange that all standard, strong and weak regularity can be transmitted; from (6.22) we can see* ***(d)*** *actually corresponds to "additivity of marginal functions with disjoint blocks", or equivalently we take* $g(t_1, ..., t_r) = t_1 + ... + t_r$*, in which strong regularity still cannot be transmitted if we look at its details.*

Note that any function is strong regular under weakest partition, thus by letting π_i in **(d)** be the weakest partition (then $\bar{\pi} = \pi_0$), we have the following results:

Corollary 1. *For any functions* h_i*'s in* ***(d)*** *of Lemma (5) (not necessarily regular), we have* $f(\mathbf{x}) = f(\mathbf{x}_{(1)}, ..., \mathbf{x}_{(r)}) = g(h_1(\mathbf{x}_{(1)}), ..., h_r(\mathbf{x}_{(r)}))$ *is standard regular under* π_0*.*

The results in Lemma (3) and Lemma (5) consist our main results on regularity, and in following sections, we use these results to show the convergence for our algorithm.

6.3 Convergence of Alternating-Direction Algorithms for Regularized SVD

Now go back to our formulation for regularized SVD problem:

$$-2\mathbf{u}^T\mathbf{X}\mathbf{v} + \mathbf{u}^T\mathbf{u} \cdot (\mathbf{v}^T\mathbf{v} + \lambda||\mathbf{v}||_1^2).$$

Since we use an alternating direction scheme, which is a special case of block coordinate descent, our proof is based on the regularity results in previous section and convergence results in [21] (Theorem 4.1 and theorem 5.1).

It is easy to check that the formulation above is strictly convex w.r.t. $\mathbf{u}$ when $\mathbf{v}$ is fixed, and vice versa. As we know strictly convex function has at most one minimum point, therefore this satisfies the condition of theorem 4.1 (c) in [21]: "If $f(x_1, ..., x_N)$ has at most one minimum in x_k for $k = 2, ..., N-1$" (here we only have two blocks, x_1 is $\mathbf{u}$, x_2 is $\mathbf{v}$). Thus, our alternating-direction algorithm will converge to a BCM point under partition $\big\{\{1, ..., n\}, \{n+1, ..., n+p\}\big\}$.

To show it converges to a stationary point, we need to show the objective function is regular under the above partition.

We use Corollary 1 to prove the regularity, now $\pi_0 = \big\{\{1, ..., n\}, \{n+1, ..., n+p\}\big\}$, $r = 2$, $s_1 = n$, $s_2 = p$, $\pi_1 = \big\{\{1, ..., n\}\big\}$, $\pi_2 = \big\{\{1, ..., p\}\big\}$, $h_1(\mathbf{u}) = \mathbf{u}^T\mathbf{u}$, $h_2(\mathbf{v}) = \mathbf{v}^T\mathbf{v} + \lambda||\mathbf{v}||_1^2$, $g(t_1, t_2) = t_1 \cdot t_2$, with $t_1 \geq 0$ and $t_2 \geq 0$.

$g(\cdot)$ is continuously differentiable, and its partial derivatives are nonnegative given $t_1 \geq 0$ and $t_2 \geq 0$, thus $\mathbf{u}^T\mathbf{u} \cdot (\mathbf{v}^T\mathbf{v} + \lambda||\mathbf{v}||_1^2)$ is standard regular under π_0.

In addition, $-2\mathbf{u}^T\mathbf{X}\mathbf{v}$ is a differentiable function, thus using result **(a)** in Lemma (3), the objective function in sparse-smooth problem is standard regular under π_0.

Therefore we showed that the alternating-direction algorithm we used converges to a stationary point for our regularized SVD problem (3.12).

6.4 Convergence for Regularized SVD with Orthogonal Constraint

When adding orthogonal constraint on our regularized SVD problem, the formulation becomes:

$$-2\mathbf{u}^T\mathbf{X}\mathbf{v} + \mathbf{u}^T\mathbf{u} \cdot (\mathbf{v}^T\mathbf{v} + \lambda||\mathbf{v}||_1^2), \text{ sub to } \mathbf{v} \perp \mathbf{V}_k$$

By introducing basis matrices $\mathbf{V}_k$ and $\mathbf{V}_k^\perp$, $\mathbf{v} \perp \mathbf{V}_k$ can be denoted as $\mathbf{v} = \mathbf{V}_k^\perp\mathbf{t}$

($\mathbf{t} \in \mathbb{R}^{p-k}$), and the formulation becomes:

$$\min_{\mathbf{u},\mathbf{t}} -2\mathbf{u}^T\mathbf{X}\mathbf{V}_k^{\perp}\mathbf{t} + \mathbf{u}^T\mathbf{u} \cdot (\mathbf{t}^T(\mathbf{V}_k^{\perp})^T\mathbf{V}_k\mathbf{t} + \lambda||(\mathbf{V}_k^{\perp})^T\mathbf{t}||_1^2).$$

We can see this formulation is marginally strictly convex w.r.t. $\mathbf{u}$ and $\mathbf{v}$, respectively thus it has unique minimum when updating $\mathbf{u}$ or $\mathbf{v}$. Thus again according to Theorem 4.1 (c) in [21], our algorithm (using formulation about $\mathbf{u}$ and $\mathbf{t}$) converges to a block coordinatewise minimum (BCM) point under partition $\big\{\{1, ..., n-k\}, \{n-k+1, ..., n+p-2k\}\big\}$.

To show it converges to a stationary point, we need to prove the regularity, again we use Corollary 1, now $\pi_0 = \big\{\{1, ..., n-k\}, \{n-k+1, ..., n+p-2k\}\big\}$, $r = 2$, $s_1 = n-k$, $s_2 = p-k$, $\pi_1 = \big\{\{1, ..., n-k\}\big\}$, $\pi_2 = \big\{\{1, ..., p-k\}\big\}$, $h_1(\mathbf{u}) = \mathbf{u}^T\mathbf{u}$, $h_2(\mathbf{t}) = \mathbf{t}^T(\mathbf{V}_k^{\perp})^T\mathbf{V}_k\mathbf{t} + \lambda||(\mathbf{V}_k^{\perp})^T\mathbf{t}||_1^2$, $g(t_1, t_2) = t_1 \cdot t_2$, with $t_1 \geq 0$ and $t_2 \geq 0$.

$g(\cdot)$ is continuously differentiable, and its partial derivatives are nonnegative given $t_1 \geq 0$ and $t_2 \geq 0$, thus $\mathbf{u}^T\mathbf{u} \cdot (\mathbf{t}^T(\mathbf{V}_k^{\perp})^T\mathbf{V}_k\mathbf{t} + \lambda||(\mathbf{V}_k^{\perp})^T\mathbf{t}||_1^2)$ is standard regular under π_0.

In addition, $-2\mathbf{u}^T\mathbf{X}\mathbf{V}_k^{\perp}\mathbf{t}$ is a differentiable function, thus using result **(a)** in Lemma (3), the objective function in sparse-smooth problem with orthogonal constraint is standard regular under π_0.

Therefore we showed that the alternating-direction algorithm we used converges to a stationary point for our regularized SVD problem (5.1).

7. FURTHER PROGRESS: CONVERGENCE TO GLOBAL OPTIMA

7.1 Regularized SVD Problem and Convergence for Power Iteration

We still starts from our main formulation (problem (3.12)):

$$\min_{\mathbf{u},\mathbf{v}} ||\mathbf{X} - \mathbf{u}\mathbf{v}^T||_F^2 + \lambda \mathbf{u}^T\mathbf{u} \cdot ||\mathbf{v}||_1^2,$$

which is equivalent to (Rayleigh-Quotient form) (problem (3.13))

$$\min_{\mathbf{u},\mathbf{v}} -2\mathbf{u}^T\mathbf{X}\mathbf{v} + \mathbf{u}^T\mathbf{u} \cdot (\lambda||\mathbf{v}||_1^2 + ||\mathbf{v}||_2^2).$$

In previous section, we have proved that our algorithm converges to a stationary point. Any stationary point of (3.13) must satisfies:

$$\begin{aligned} \mathbf{0} &\in -2\mathbf{X}^T\mathbf{u} + \mathbf{u}^T\mathbf{u} \cdot \big(2\mathbf{v} + 2\lambda_{\mathbf{v}}||\mathbf{v}||_1 \cdot \mathbf{sgn}(\mathbf{v})\big), \\ \mathbf{0} &= -2\mathbf{X}\mathbf{v} + 2(||\mathbf{v}||_2^2 + \lambda_{\mathbf{v}}||\mathbf{v}||_1^2)\mathbf{u}. \end{aligned} \tag{7.1}$$

In the aspect of algorithm, the updating rules for $\mathbf{u}$ and $\mathbf{v}$ are

$$\begin{aligned} \mathbf{u} &\leftarrow \frac{\mathbf{X}\mathbf{v}}{||\mathbf{v}||_2^2 + \lambda_{\mathbf{v}}||\mathbf{v}||_1^2} \propto \mathbf{X}\mathbf{v}, \\ \mathbf{v} &\leftarrow \mathrm{SLSA}(\mathbf{X}^T\mathbf{u}; \lambda_{\mathbf{v}})/\mathbf{u}^T\mathbf{u}. \end{aligned} \tag{7.2}$$

We plan to utilize the convergence result about standard power's iteration, therefore we need a matrix-vector multiplication form for iteration above.

Updating rule of $\mathbf{u}$ is already in the form of matrix-vector multiplication.

As for $\mathbf{v}$, since it is a squared lasso signal approximation (SLSA) problem, to develop a generally-used result, let's switch the scenario to standard SLSA problem

in next section.

7.2 Matrix-Vector Multiplication Form for SLSA Problem

Suppose the original signal is $\mathbf{x} = (x_1, x_2, ..., x_n)^T$ and the tuning parameter is λ. The main procedure for SLSA is

1. First order the entries of $\mathbf{x}$ by absolute value in an increasing order: $|x_{k_1}| \geq |x_{k_2}| \geq \cdots \geq |x_{k_n}|$.

2. Then denote $z_i = x_{k_i}$, $1 \leq i \leq n$.

3. Find the cutting point, i.e., find a unique integer r between 1 and n, such that

$$|z_i| > \frac{\lambda}{\lambda r + 1} \sum_{j=1}^{r} |z_j|, \ 1 \leq i \leq r, \quad |z_i| \leq \frac{\lambda}{\lambda r + 1} \sum_{j=1}^{r} |z_j|, \ r < i \leq n. \quad (7.3)$$

   Or,

$$|z_r| > \frac{\lambda}{\lambda r + 1} \sum_{j=1}^{r} |z_j|, \quad |z_{r+1}| \leq \frac{\lambda}{\lambda r + 1} \sum_{j=1}^{r} |z_j|. \quad (7.4)$$

4. Threshold to zero for z_i's, $r < i \leq n$.

5. Linearly shrinkage for z_i's, $1 \leq r \leq r$. Denote $\mathbf{z}_0 = (z_1, \ldots, z_r)$, the subvector to be shrank, and $\tilde{\mathbf{z}}_0 = (\tilde{z}_1, \ldots, \tilde{z}_r)$, which is the subvector of entries after shrinking, then we have $|\tilde{\mathbf{z}}_0| = (\mathbf{I}_r - \frac{\lambda}{\lambda r+1} \mathbf{1}_r \mathbf{1}_r^T)|\mathbf{z}_0|$.

To translate the whole procedure above, we need the following notations and manipulations:

1. For the step of transformation $\mathbf{x} \to \mathbf{z}$, define the order transformation matrix $\mathcal{T}_{(\mathbf{x})}$ such that $\mathcal{T}_{(\mathbf{x})}\mathbf{x} = \mathbf{z}$. We know $\mathcal{T}_{(\mathbf{x})}$ is an orthogonal matrix, also we have $\mathcal{T}_{(\mathbf{x})}\text{diag}(\mathbf{x})\mathcal{T}_{(\mathbf{x})}^T = \text{diag}(\mathbf{z})$.

   In addition, $\mathcal{T}_{(\tilde{\mathbf{x}})} = \mathcal{T}_{(\mathbf{x})}$, $\mathcal{T}_{(\mathbf{x})}\tilde{\mathbf{x}} = \tilde{\mathbf{z}}$, where $\tilde{\mathbf{x}} = \text{SLSA}(\mathbf{x}; \lambda)$ and $\tilde{\mathbf{z}} = \text{SLSA}(\mathbf{z}; \lambda)$

2. Define $\mathcal{S}_{(\mathbf{x})} = \text{diag}(\text{sgn}(\mathbf{x}))$, and same for $\mathcal{S}_{(\mathbf{z})}$. Then we have $\mathcal{S}_{(\mathbf{x})}\mathbf{x} = |\mathbf{x}|$, $\mathcal{S}_{(\mathbf{z})}\mathbf{z} = |\mathbf{z}|$, $\mathcal{T}_{(\mathbf{x})}\text{sgn}(\mathbf{x}) = \text{sgn}(\mathbf{z})$, $\mathcal{T}_{(\mathbf{x})}|\mathbf{x}| = |\mathbf{z}|$, and $\mathcal{T}_{(\mathbf{x})}\mathcal{S}_{(\mathbf{x})}\mathcal{T}_{(\mathbf{x})}^T = \mathcal{S}_{(\mathbf{z})}$.

3. For $\mathbf{z}$, we have $\mathbf{z} = (\mathbf{z}_0^T, \mathbf{z}_1^T)^T$, the partition of being thresholded and shrank, similarly we have $\tilde{\mathbf{z}} = (\tilde{\mathbf{z}}_0^T, \tilde{\mathbf{z}}_1^T)^T$. Then $|\tilde{\mathbf{z}}_0| = (\mathbf{I}_r - \frac{\lambda}{\lambda r+1}\mathbf{1}_r\mathbf{1}_r^T)|\mathbf{z}_0|$, by multiplying $\mathcal{S}_{(\tilde{\mathbf{z}}_0)}$ (note that $\mathcal{S}_{(\tilde{\mathbf{z}}_0)} = \mathcal{S}_{(\mathbf{z}_0)}$, since $\mathbf{z}_0$ is the shrank part, not thresholded part), we have $\tilde{\mathbf{z}}_0 = \mathcal{S}_{(\mathbf{z}_0)}(\mathbf{I}_r - \frac{\lambda}{\lambda r+1}\mathbf{1}_r\mathbf{1}_r^T)\mathcal{S}_{(\mathbf{z}_0)}\mathbf{z}_0$. As for $\tilde{\mathbf{z}}_1$, we have $\tilde{\mathbf{z}}_1 = \mathbf{0} \cdot \mathbf{z}_1$.

4. Therefore

$$\begin{aligned}\tilde{\mathbf{z}} = \begin{pmatrix} \tilde{\mathbf{z}}_0 \\ \tilde{\mathbf{z}}_1 \end{pmatrix} &= \begin{pmatrix} \mathcal{S}_{(\mathbf{z}_0)}(\mathbf{I}_r - \frac{\lambda}{\lambda r+1}\mathbf{1}_r\mathbf{1}_r^T)\mathcal{S}_{(\mathbf{z}_0)}\mathbf{z}_0 \\ \mathbf{0} \cdot \mathbf{z}_1 \end{pmatrix} \\ &= \mathcal{S}_{(\mathbf{z})} \cdot \begin{pmatrix} \mathbf{I}_r - \frac{\lambda}{\lambda r+1}\mathbf{1}_r\mathbf{1}_r^T, & \mathbf{0} \\ \mathbf{0}, & \mathbf{0} \end{pmatrix} \cdot \mathcal{S}_{(\mathbf{z})} \cdot \mathbf{z}.\end{aligned} \tag{7.5}$$

5. On the other hand, $\mathcal{T}_{(\mathbf{x})}\tilde{\mathbf{x}} = \tilde{\mathbf{z}}$, because threshold is one-by-one, and does not affect the order.

6. Thus,

$$\tilde{\mathbf{x}} = \mathcal{T}_{(\mathbf{x})}^{-1}\tilde{\mathbf{z}} = \mathcal{T}_{(\mathbf{x})}^{-1} \cdot \mathcal{S}_{(\mathbf{z})} \cdot \begin{pmatrix} \mathbf{I}_r - \frac{\lambda}{\lambda r+1}\mathbf{1}_r\mathbf{1}_r^T, & \mathbf{0} \\ \mathbf{0}, & \mathbf{0} \end{pmatrix} \cdot \mathcal{S}_{(\mathbf{z})} \cdot \mathbf{z} \tag{7.6}$$

$$= (\mathcal{T}_{(\mathbf{x})}\mathcal{S}_{(\mathbf{x})})^T \cdot \begin{pmatrix} \mathbf{I}_r - \frac{\lambda}{\lambda r+1}\mathbf{1}_r\mathbf{1}_r^T, & \mathbf{0} \\ \mathbf{0}, & \mathbf{0} \end{pmatrix} \cdot \mathcal{T}_{(\mathbf{x})}\mathcal{S}_{(\mathbf{x})} \cdot \mathbf{x}. \tag{7.7}$$

7. If we denote

$$\mathbf{M} = (\mathcal{T}_{(\mathbf{x})}\mathcal{S}_{(\mathbf{x})})^T \cdot \begin{pmatrix} \mathbf{I}_r - \frac{\lambda}{\lambda r+1}\mathbf{1}_r\mathbf{1}_r^T, & \mathbf{0} \\ \mathbf{0}, & \mathbf{0} \end{pmatrix} \cdot \mathcal{T}_{(\mathbf{x})}\mathcal{S}_{(\mathbf{x})}, \tag{7.8}$$

then we have

$$\tilde{\mathbf{x}} = \mathrm{SLSA}(\mathbf{x}; \lambda) = \mathbf{M}\mathbf{x}. \tag{7.9}$$

For regularized SVD problem, the signal $\mathbf{x}$ (which is $\mathbf{X}^T\mathbf{u}/(\mathbf{u}^T\mathbf{u})$) to be processed changes during iterations (since $\mathbf{u}$ and $\mathbf{v}$ changes), we need to investigate the stability and different possibilities for $\mathbf{M}$.

As we know, $\mathbf{u}$ changes (diverges at first and converges finally) during iterations, as function of $\mathbf{u}$, $\mathbf{M} = \mathbf{M}(\mathbf{u})$ needs to be stable after certain iteration step, so that our analysis above could make sense.

According to equation (7.8), the stability of $\mathbf{M}$ depends on $\mathcal{S}_{(\mathbf{x})}$, $\mathcal{T}_{(\mathbf{x})}$, and number r. The only nonzero part of

$$\begin{pmatrix} \mathbf{I}_r - \frac{\lambda}{\lambda r+1}\mathbf{1}_r\mathbf{1}_r^T, & \mathbf{0} \\ \mathbf{0}, & \mathbf{0} \end{pmatrix} \tag{7.10}$$

is its left-upper corner, which makes the corresponding part of $\mathcal{T}_{(\mathbf{x})}\mathcal{S}_{(\mathbf{x})}$ not important, or, $S_{(\mathbf{z}_1)}$ not important. $\mathbf{I}_r - \frac{\lambda}{\lambda r+1}\mathbf{1}_r\mathbf{1}_r^T$ is a sequentially symmetric matrix, which makes $\mathcal{S}_{(\mathbf{z}_0)}$ not important. The only thing matters is the boundary between shrank part and thresholded part, or equivalently, we need $|z_{r+1}| < |z_r|$ and $|z_{r+1}| < \frac{\lambda}{\lambda r+1}\sum_{j=1}^{r+1}|z_j| \leq z_r$. As [16] already analyzed, this requirement only excludes a zero measure set.

Suppose

$$|z_{r+1}| < \frac{\lambda}{\lambda r+1}\sum_{j=1}^{r+1}|z_j|$$

holds, then there exists a neighbor of the current point $\mathbf{x}$, say $o(\mathbf{x}, \delta_0)$, such that $\forall \mathbf{x}_0 \in o(\mathbf{x}, \delta_0)$, $\mathrm{SLSA}(\mathbf{x}_0, \lambda)$ has the same "pattern", in the sense that zero locations

and shrank locations are the same (not to mention the same r), or more meaningfully, $\mathrm{SLSA}(\mathbf{x}_0, \lambda) = \mathbf{M}\mathbf{x}_0$, with $\mathbf{M}$ being fixed.

To count how many different possibilies $\mathbf{M}$ has, again due to special form of

$$\begin{pmatrix} \mathbf{I}_r - \frac{\lambda}{\lambda r+1}\mathbf{1}_r\mathbf{1}_r^T, & \mathbf{0} \\ \mathbf{0}, & \mathbf{0} \end{pmatrix}, \tag{7.11}$$

there are only two things that matter, the number r and the r nonzero locations, therefore, the number of possibilities for $\mathbf{M}$ should be $\sum_{r=0}^{n} C_n^r 2^r = 3^n$, which is a huge yet finite number.

7.3 Convergence Result for Regularized SVD

Go back to problem (3.13), based on results in previous section, we have

$$\mathbf{u} \propto \mathbf{X}\mathbf{v}, \qquad \mathbf{v} \propto \mathrm{SLSA}(\mathbf{X}^T\mathbf{u}, \lambda) \propto \mathbf{M}\mathbf{X}^T\mathbf{u}, \tag{7.12}$$

which is equivalent to

$$\mathbf{u} \propto \mathbf{X}\mathbf{M}^{1/2}(\mathbf{M}^{-1/2}\mathbf{v}), \qquad \mathbf{M}^{-1/2}\mathbf{v} \propto \mathbf{M}^{1/2}\mathbf{X}^T\mathbf{u}, \tag{7.13}$$

where

$$\mathbf{M}^{-1/2} = (\mathcal{T}_{(\mathbf{x})}\mathcal{S}_{(\mathbf{x})})^T \cdot \begin{pmatrix} (\mathbf{I}_r - \frac{\lambda}{\lambda r+1}\mathbf{1}_r\mathbf{1}_r^T)^{-1/2}, & \mathbf{0} \\ \mathbf{0}, & \mathbf{0} \end{pmatrix} \cdot \mathcal{T}_{(\mathbf{x})}\mathcal{S}_{(\mathbf{x})}, \tag{7.14}$$

Note that $\mathbf{M}$ is not invertible, however when $\mathbf{v} \propto \mathbf{M}^{1/2}\mathbf{X}^T\mathbf{u}$, we can easily verify that $\mathbf{X}\mathbf{M}^{1/2}(\mathbf{M}^{-1/2}\mathbf{v}) = \mathbf{X}\mathbf{v}$.

If we denote $\boldsymbol{\xi} = \mathbf{u}$, $\boldsymbol{\eta} = \mathbf{M}^{-1/2}\mathbf{v}$, and $\mathbf{P} = \mathbf{X}\mathbf{M}^{1/2}$, then we have

$$\boldsymbol{\xi} \propto \mathbf{P}\boldsymbol{\eta}, \qquad \boldsymbol{\eta} \propto \mathbf{P}^T\boldsymbol{\xi}. \tag{7.15}$$

We can see $(\boldsymbol{\xi}, \boldsymbol{\eta})$ are a pair of singular vectors of $\mathbf{P} = \mathbf{X}\mathbf{M}^{-1/2}$, and the iteration step

$$\begin{aligned} \mathbf{u} &\leftarrow \frac{\mathbf{X}\mathbf{v}}{||\mathbf{v}||_2^2 + \lambda_{\mathbf{v}}||\mathbf{v}||_1^2} \propto \mathbf{X}\mathbf{v}, \\ \mathbf{v} &\leftarrow \mathrm{SLSA}(\mathbf{X}^T\mathbf{u}; \lambda_{\mathbf{v}})/\mathbf{u}^T\mathbf{u}. \end{aligned} \tag{7.16}$$

is equivalent to

$$\boldsymbol{\xi} \leftarrow \frac{\mathbf{P}\boldsymbol{\eta}}{\boldsymbol{\eta}^T\boldsymbol{\eta}} \cdot \frac{\boldsymbol{\eta}^T\boldsymbol{\eta}}{||\mathbf{v}||_2^2 + \lambda_{\mathbf{v}}||\mathbf{v}||_1^2}, \qquad \boldsymbol{\eta} \leftarrow \frac{\mathbf{P}^T\boldsymbol{\xi}}{\boldsymbol{\xi}^T\boldsymbol{\xi}}, \tag{7.17}$$

which is the standard alternating direction algorithm for SVD (up to a small scale difference, and when $\mathbf{v}$ begins to converge, the scale difference converges to a constant), which differs from standard power iteration only at scale. Alternating direction algorithm and power iteration have the same convergence result, moreover, as long as the scale difference finally converges and does not make trouble, any variant method has same converging property as standard power iteration.

As we know, the global optima of (3.13) must be a stationary point (may not be the unique stationary point), therefore it must corresponds one $\mathbf{M}$ (since you can see we actually many possibilities for $\mathbf{M}$, which would be discussed in the end of this section), therefore if we could find the correct $\mathbf{M}$ (say $\mathbf{M}_0$), the global optima is actually the leading singular vector of $\mathbf{X}\mathbf{M}_0^{1/2}$.

As a summary, the structure of convergence to global/local optimal has been made clear (given that the correct $\mathbf{M}$ is identified), and we have same global optimality

result as in standard power iteration.

The convergence to global optima of power iteration is based on two conditions:

1. Leading singular value of $\mathbf{X}$ is strictly greater in magnitude than its other singular values;

2. The initialization vector $\mathbf{u}_0$ has a nonzero component in the direction of an left singular vector associated with the dominant singular value.

Due to the huge number of possibilities for $\mathbf{M}$, our result here has contribution in theoretic level, not practical level. However, one can follow this strategy and design some quick filtering method to reduce possible number of $\mathbf{M}$ (for example make local search based on standard SVD plus post threshold). Given that most SVD-based algorithm fails to show their convergence, not to mention convergence to global optima, our results here could be considered as a significant breakthrough.

8. MISSING VALUES AND ENTRYWISE CROSS-VALIDATION

8.1 Single-Layer Regularized SVD with Missing Values

Suppose the data matrix $\mathbf{X}$ has some missing values, we denote the missing locations and observed locations as m and o, respectively; also denote their corresponding indicator matrix as $\mathbf{I}_m$ and $\mathbf{I}_o$, respectively, then we have the following decomposition

$$\mathbf{X} = \mathbf{X} \odot (\mathbf{I}_m + \mathbf{I}_o) = \mathbf{X}_m + \mathbf{X}_o, \tag{8.1}$$

where $\odot$ stands for elementwise product.

Therefore under single-layer SVD model $\mathbf{X} = \mathbf{u}\mathbf{v}^T + \mathbf{E}$, $E = (e_{ij})$, e_{ij} iid. $\sim \mathcal{N}(0, \sigma^2)$, the observed log-likelihood function is proportional to:

$$||(\mathbf{X} - \mathbf{u}\mathbf{v}^T) \odot \mathbf{I}_o||_F^2. \tag{8.2}$$

We hope $\mathbf{v}$ to be sparse, no matter if the data entries are completely observed or not, thus we have the following penalized observed likelihood:

$$||(\mathbf{X} - \mathbf{u}\mathbf{v}^T) \odot \mathbf{I}_o||_F^2 + \lambda \mathbf{u}^T\mathbf{u} \cdot ||\mathbf{v}||_1^2. \tag{8.3}$$

We can see the solution $(\mathbf{u}^*, \mathbf{v}^*)$ of optimizing (8.3) is also the solution in

$$\min_{\mathbf{u},\mathbf{v},\mathbf{X}_m} ||\mathbf{X}_o + \mathbf{X}_m - \mathbf{u}\mathbf{v}^T||_F^2 + \lambda \mathbf{u}^T\mathbf{u} \cdot ||\mathbf{v}||_1^2. \tag{8.4}$$

The equivalence is due to the following fact: given any $(\mathbf{u}, \mathbf{v})$, the optima of $\mathbf{X}_m$

in (8.4) would be $\mathbf{u}\mathbf{v}^T \odot \mathbf{I}_m$, which results in

$$\begin{aligned}
&\min_{\mathbf{u},\mathbf{v},\mathbf{X}_m} ||\mathbf{X}_o + \mathbf{X}_m - \mathbf{u}\mathbf{v}^T||_F^2 + \lambda \mathbf{u}^T\mathbf{u} \cdot ||\mathbf{v}||_1^2, \\
\Leftrightarrow &\min_{\mathbf{u},\mathbf{v}} \Big\{ \min_{\mathbf{X}_m} ||\mathbf{X}_o + \mathbf{X}_m - \mathbf{u}\mathbf{v}^T||_F^2 + \lambda \mathbf{u}^T\mathbf{u} \cdot ||\mathbf{v}||_1^2 \Big\}, \\
\Leftrightarrow &\min_{\mathbf{u},\mathbf{v}} ||\mathbf{X}_o + \mathbf{u}\mathbf{v}^T \odot \mathbf{I}_m - \mathbf{u}\mathbf{v}^T||_F^2 + \lambda \mathbf{u}^T\mathbf{u} \cdot ||\mathbf{v}||_1^2, \\
\Leftrightarrow &\min_{\mathbf{u},\mathbf{v}} ||\mathbf{X}_o - \mathbf{u}\mathbf{v}^T \odot \mathbf{I}_o||_F^2 + \lambda \mathbf{u}^T\mathbf{u} \cdot ||\mathbf{v}||_1^2, \\
\Leftrightarrow &\min_{\mathbf{u},\mathbf{v}} ||(\mathbf{X} - \mathbf{u}\mathbf{v}^T) \odot \mathbf{I}_o||_F^2 + \lambda \mathbf{u}^T\mathbf{u} \cdot ||\mathbf{v}||_1^2.
\end{aligned} \tag{8.5}$$

To optimize (8.4), we could use alternating-direction strategy. In particular, we have three optimizing blocks: $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{X}_m$ (or, more precisely, $\mathfrak{X}_m$, which is a vector obtained by first stretching $\mathbf{X}_m$ then removing zero entries.), the updating rules all have closed form: $\mathbf{u}$ and $\mathbf{v}$ can be updated as in non-missing case, $\mathbf{X}_m$ can be updated by $\mathbf{u}\mathbf{v}^T \odot \mathbf{I}_m$.

We summarize above analysis as the following algorithm.

Alternating-Direction-Entry-Completion-Algorithm to solve (8.3):

Input: $\mathbf{X}_o \in \mathbb{R}^{n\times p}$ with missing values and $\lambda > 0$.

Output: $\mathbf{u} \in \mathbb{R}^n$, $\mathbf{v} \in \mathbb{R}^p$;

Algortihm:

1. Set initial value $\mathbf{u}^0$ and $\mathbf{v}^0$ for $\mathbf{u}$ and $\mathbf{v}$, respectively;
2. For $i = 0, 1, 2, ...$, repeat the following steps until convergence:

$$\begin{aligned}
\mathbf{X}_m^{i+1} =& \mathbf{u}^i(\mathbf{v}^i)^T \odot \mathbf{I}_m \\
\mathbf{u}^{i+1} =& \frac{(\mathbf{X}_o + \mathbf{X}_m^{i+1})\mathbf{v}^i}{\lambda||\mathbf{v}^i||_1^2 + ||\mathbf{v}^i||_2^2} \\
\mathbf{v}^{i+1} =& \text{SLSA}\Big(\frac{(\mathbf{X}_o + \mathbf{X}_m^{i+1})^T\mathbf{u}^{i+1}}{(\mathbf{u}^{i+1})^T\mathbf{u}^{i+1}}, \lambda\Big).
\end{aligned} \tag{8.6}$$

8.2 Convergence to Stationary Point

First, we have the following standard form of objective function:

$$\begin{aligned} f(\mathbf{u},\mathbf{v},\mathfrak{X}_m) =&\ f_0(\mathbf{u},\mathbf{v},\mathfrak{X}_m) + f_1(\mathbf{u},\mathbf{v}), \\ =&\ ||\mathbf{X}_o + \mathbf{X}_m - \mathbf{u}\mathbf{v}^T||_F^2 + \lambda \mathbf{u}^T\mathbf{u}\cdot||\mathbf{v}||_1^2, \\ \propto&\ ||\mathbf{X}_m||_F^2 - 2\mathbf{u}^T(\mathbf{X}_o+\mathbf{X}_m)\mathbf{v} + \mathbf{u}^T\mathbf{u}\cdot(\mathbf{v}^T\mathbf{v} + \lambda||\mathbf{v}||_1^2). \end{aligned} \tag{8.7}$$

Same as in non-missing case, we use the results we developed in section 6 and Theorem 3.1 and Theorem 4.1 in [21].

It is easy to check that (8.7) is marginally strictly convex w.r.t. $\mathfrak{X}_m$, $\mathbf{u}$, and $\mathbf{v}$. As we know strictly convex function has at most one minimum point, therefore this satisfies the condition of theorem 4.1 (c) in [21]: "If $f(\mathbf{x}_1,...,\mathbf{x}_N)$ has at most one minimum in $\mathbf{x}_k$ for $k = 2,...,N-1$" (here we have three blocks, $\mathbf{x}_1$ is $\mathbf{u}$, $\mathbf{x}_2$ is $\mathbf{v}$, and $\mathbf{x}_3$ is $\mathfrak{X}_m$). Thus, our **Alternating-Direction-Entry-Completion-Algorithm** will converge to a BCM (block coordinatewise minimum) point under partition $\big\{\{1,...,n\},\{n+1,...,n+p\},\{n+p+1,...,n+p+\tau\}\big\}$, where τ is length of $\mathfrak{X}_m$, i.e. number of missing entries.

To show it converges to a stationary point, we need to show the objective function is regular under the above partition.

We use Corollary 1 to prove the regularity, now $\pi_0 = \big\{\{1,...,n\},\{n+1,...,n+p\},\{n+p+1,...,n+p+\tau\}\big\}$, $r = 3$, $s_1 = n$, $s_2 = p$, and $s_3 = \tau$, $\pi_1 = \big\{\{1,...,n\}\big\}$, $\pi_2 = \big\{\{1,...,p\}\big\}$, $\pi_3 = \big\{\{n+p+1,...,n+p+\tau\}\big\}$, $h_1(\mathbf{u}) = \mathbf{u}^T\mathbf{u}$, $h_2(\mathbf{v}) = \mathbf{v}^T\mathbf{v}+\lambda||\mathbf{v}||_1^2$, $h_3(\mathfrak{X}_m) = 0$, $g(t_1,t_2,t_3) = t_1\cdot t_2$, with $t_1 \geq 0$, $t_2 \geq 0$, and $t_3 \in \mathbb{R}$.

$g(\cdot)$ is continuously differentiable, and its partial derivatives are nonnegative given $t_1 \geq 0$, $t_2 \geq 0$, and $t_3 \in \mathbb{R}$, thus $\mathbf{u}^T\mathbf{u}\cdot(\mathbf{v}^T\mathbf{v} + \lambda||\mathbf{v}||_1^2)$ is standard regular under π_0.

In addition, $||\mathbf{X}_m||_F^2 - 2\mathbf{u}^T(\mathbf{X}_o+\mathbf{X}_m)\mathbf{v}$ is a differentiable function, thus using

result **(a)** in Lemma (3), the objective function (8.7) is standard regular under π_0. Therefore we complete the proof to the convergence.

8.3 Multi-Layer Regularized SVD with Missing Values

The formulation regularized SVD with orthogonal constraint and missing values is

$$\min_{\mathbf{u},\mathbf{v},\mathbf{X}_m} ||\mathbf{X}_o + \mathbf{X}_m - \mathbf{u}\mathbf{v}^T||_F^2 + \mathbf{u}^T\mathbf{u} \cdot (\mathbf{v}^T\mathbf{v} + \lambda||\mathbf{v}||_1^2), \text{ sub to } \mathbf{v} \perp \mathbf{V}_k, \tag{8.8}$$

By introducing basis matrices $\mathbf{V}_k$ and $\mathbf{V}_k^\perp$, $\mathbf{v} \perp \mathbf{V}_k$ can be denoted as $\mathbf{v} = \mathbf{V}_k^\perp\mathbf{t}$ ($\mathbf{t} \in \mathbb{R}^{p-k}$), and the above formulation becomes:

$$\min_{\mathbf{u},\mathbf{t},\mathbf{X}_m} ||\mathbf{X}_m||_F^2 - 2\mathbf{u}^T(\mathbf{X}_o + \mathbf{X}_m)\mathbf{V}_k^\perp\mathbf{t} + \mathbf{u}^T\mathbf{u} \cdot ((\mathbf{t}^T\mathbf{V}_k^\perp)^T\mathbf{V}_k^\perp\mathbf{t} + \lambda||\mathbf{V}_k^\perp\mathbf{t}||_1^2). \tag{8.9}$$

To optimize (8.9), we could use alternating-direction strategy. In particular, we have three optimizing blocks: $\mathbf{u}$, $\mathbf{t}$, and $\mathfrak{X}_m$, the updating rules all have closed form: $\mathbf{u}$ and $\mathbf{t}$ can be updated as in non-missing case, $\mathbf{X}_m$ can be updated by $\mathbf{u}\mathbf{v}^T \odot \mathbf{I}_m$, or $\mathbf{u}\mathbf{t}^T(\mathbf{V}_k^\perp)^T \odot \mathbf{I}_m$.

8.4 Convergence to Stationary Point

It is easy to check that formulation (8.9) is marginally strictly convex w.r.t. $\mathbf{u}$, $\mathbf{t}$, and $\mathfrak{X}_m$. Again this satisfies the condition of theorem 4.1 (c) in [21]. Thus, our **Generalized Algorithm for Regularized SVD Problem with Orthogonal Constraint and Missing Values** converges to a BCM point under partition $\big\{\{1, ..., n-k\}, \{n-k+1, ..., n+p-2k\}, \{n+p-2k+1, ..., n+p-2k+\tau\}\big\}$, where τ is number of missing entries.

To show it converges to a stationary point, we need to prove its regularity, again

we use Corollary 1, now $\pi_0 = \big\{\{1, ..., n-k\}, \{n-k+1, ..., n+p-2k\}, \{n+p-2k+1, ..., n+p-2k+\tau\}\big\}$, $r = 3$, $s_1 = n-k$, $s_2 = p-k$, $s_3 = \tau$, $\pi_1 = \big\{\{1, ..., n-k\}\big\}$, $\pi_2 = \big\{\{n-k+1, ..., n+p-2k\}\big\}$, $\pi_3 = \big\{\{n+p-2k+1, ..., n+p-2k+\tau\}\big\}$, $h_1(\mathbf{u}) = \mathbf{u}^T\mathbf{u}$, $h_2(\mathbf{t}) = (\mathbf{t}^T\mathbf{V}_k^{\perp})^T\mathbf{V}_k^{\perp}\mathbf{t} + \lambda||\mathbf{V}_k^{\perp}\mathbf{t}||_1^2$, $h_3(\mathfrak{X}_m) = 0$, $g(t_1, t_2, t_3) = t_1 \cdot t_2$, with $t_1 \geq 0$, $t_2 \geq 0$, and $t_3 \in \mathbb{R}$.

$g(\cdot)$ is continuously differentiable, and its partial derivatives are nonnegative given $t_1 \geq 0$, $t_2 \geq 0$, and $t_3 \in \mathbb{R}$, thus $(\mathbf{u}^T\mathbf{u} \cdot ((\mathbf{t}^T\mathbf{V}_k^{\perp})^T\mathbf{V}_k^{\perp}\mathbf{t} + \lambda||\mathbf{V}_k^{\perp}\mathbf{t}||_1^2)$ is standard regular under π_0.

In addition, $||\mathbf{X}_m||_F^2 - 2\mathbf{u}^T(\mathbf{X}_o + \mathbf{X}_m)\mathbf{V}_k^{\perp}\mathbf{t}$ is a differentiable function, thus using result **(a)** in Lemma (3), the objective function of problem (8.9) is standard regular under π_0. Therefore we complete the proof to the convergence.

8.5 Cross Validation for Regularized SVD

In [20], the authors developed a row-based cross-validation methods. It cannot deal with missing values in $\mathbf{X}$, also, it can not be generalized to two-way regularized SVD. In this section, we propose an entrywise cross-validation, which can deal with missing values and can be generalized.

To perform cross-validation, one can randomly split the set of entries in data matrix $\mathbf{X}$ into training data and testing data (say, $\mathbf{X} \odot \mathbf{I}_{train}$ and $\mathbf{X} \odot \mathbf{I}_{test}$). Given the training data, we can get a regularized SVD model using algorithm above, say, the estimators are $(\hat{\mathbf{u}}, \hat{\mathbf{v}})$, then we have the following loss function:

$$||(\mathbf{X} - \hat{\mathbf{u}}\hat{\mathbf{v}}^T) \odot \mathbf{I}_{test}||_F^2. \tag{8.10}$$

To make a reasonable cross-validation, we can use following principles or strategies to make a systematic subsampling:

1. **Method 1**: We can partition the indices of rows and columns as $\{1, 2, \ldots, n\} =$

$\mathbf{R} = \Sigma_{i=1}^{r} s_i$ and $\{1, 2, \ldots, p\} = \mathbf{C} = \bigcup_{j=1}^{c} b_i$. By doing so, we partition the whole data set as $r \times c$ parts. We can use similar strategy as 5-fold cross-validation, by each time leaving one part out as testing set, and all rest as training part.

2. **Method 2**: We can randomly generate several sets of indices, and each time use one set as testing set, rest as training set.

3. Note that if a whole row (or column) is missing, say, i-th row (or j-th column), then we won't have reasonable estimation of u_i (or v_j), while the testing error is also not reasonable. Therefore, we have to make sure the set of indices for testing data in **Method 2** does not include any whole row (or whole column). While **Method 1** does not have this concern.

4. Besides, if $\mathbf{X}$ contains missing values, we can split $\mathbf{X}$ into three parts $\mathbf{X}_m$, $\mathbf{X}_{train}$, and $\mathbf{X}_{test}$, where $\mathbf{X}_{train}$ and $\mathbf{X}_{test}$ consists of $\mathbf{X}_o$.

9. SIMULATION AND REAL WORLD DATA ANALYSIS

9.1 Data Generation for Simulation

For simulation data analysis, we generate data sets whose covariance matrix actually has sparse leading eigenvectors. We describe here a general scheme to generate such data. Suppose we want to generate data from $\mathbb{R}^p$ such that the q ($q < p$) leading eigenvectors of the covariance matrix $\mathbf{\Sigma}$ are sparse. Denote the first q eigenvectors as $\mathbf{v}_1, ..., \mathbf{v}_q$, which are specified to be sparse and orthonormal. The remaining $p - q$ eigenvectors are not specified to be sparse. Denote the positive eigenvalues of $\mathbf{\Sigma}$ in decreasing order as $c_1, ..., c_p$.

We first generate the other $q-p$ orthonormal eigenvectors of $\mathbf{\Sigma}$. To this end, form a full-rank matrix $\mathbf{V} = [\mathbf{v}_1, ..., \mathbf{v}_q, \mathbf{v}^*_{q+1}, ..., \mathbf{v}^*_p]$, where $\mathbf{v}_1, ..., \mathbf{v}_q$ are the pre-specified sparse eigenvectors and $\mathbf{v}^*_{q+1}, ..., \mathbf{v}^*_p$ are arbitrary. For example, the vectors $\mathbf{v}^*_{q+1}, ..., \mathbf{v}^*_p$ can be randomly drawn from $U(0, 1)$; if $\mathbf{V}$ is not of full-rank for one random draw, we can draw another set of vectors. Then, we apply the GramSchmidt orthogonalization method to $\mathbf{V}$ to obtain an orthogonal matrix $\mathbf{V} = [\mathbf{v}_1, ..., \mathbf{v}_q, \mathbf{v}_{q+1}, ..., \mathbf{v}_p]$, which is actually the matrix $\mathbf{Q}$ from the QR decomposition of $\mathbf{V}^*$. Given the orthogonal matrix $\mathbf{V}$, we form the covariance matrix $\mathbf{\Sigma}$ using the following eigen decomposition expression,

$$\mathbf{\Sigma} = c_1\mathbf{v}_1\mathbf{v}_1^T + c_2\mathbf{v}_2\mathbf{v}_2^T + c_3\mathbf{v}_3\mathbf{v}_3^T + \ + c_p\mathbf{v}_p\mathbf{v}_p^T = \mathbf{V}\mathbf{C}\mathbf{V}^T,$$

where $\mathbf{C} = \text{diag}\{c_1, ..., c_p\}$ is the eigenvalue matrix. The first q eigenvectors of $\mathbf{\Sigma}$ are the pre-specified sparse vectors $\mathbf{v}_1, ..., \mathbf{v}_q$. To generate data from the covariance matrix $\mathbf{\Sigma}$, let $\mathbf{Z}$ be a random draw from $N(0, \mathbf{I}_p)$ and $\mathbf{X} = \mathbf{V}\mathbf{C}^{1/2}\mathbf{Z}$, then $\text{cov}(\mathbf{X}) = \mathbf{\Sigma}$,

as desired.

9.2 Simulation of Low-Dimension Case

Example 1. We considered a covariance matrix with two specified sparse leading eigenvectors. The data are in $\mathbb{R}^{10}$ and generated as $\mathbf{X} \sim N(0, \mathbf{\Sigma}_1)$. Let

$$\tilde{\mathbf{v}}_1 = (1,1,1,1,0,0,0,0,0.9,0.9)^T, \quad \tilde{\mathbf{v}}_2 = (0,0,0,0,1,1,1,1,-0.3,0.3)^T.$$

The first two eigenvectors of $\mathbf{\Sigma}_1$ are then chosen to be

$$\mathbf{v}_1 = \tilde{\mathbf{v}}_1/||\tilde{\mathbf{v}}_1|| = (0.422, 0.422, 0.422, 0.422, 0, 0, 0, 0, 0.380, 0.380)^T,$$
$$\mathbf{v}_2 = \tilde{\mathbf{v}}_2/||\tilde{\mathbf{v}}_2|| = (0, 0, 0, 0, 0.489, 0.489, 0.489, 0.489, -0.147, 0.147)^T,$$

both of which have a degree of sparsity of 4. The ten eigenvalues of $\mathbf{\Sigma}_1$ are respectively 200, 100, 50, 50, 6, 5, 4, 3, 2 and 1. The first two eigenvectors explain about 70% of the total variance.

We simulated 100 datasets of size $n = 30$ and $n = 300$ respectively with the covariance matrix being $\mathbf{\Sigma}_1$. For each simulated dataset, the first two sparse loading vectors are calculated using (1) standard PCA; (2)-(4) methods from [20] (called sPCA-rSVD) with the soft, hard and SCAD thresholding rules, the procedures are referred as sPCA-rSVD-soft, sPCA-rSVD-hard and sPCA-rSVD-SCAD respectively; (5) simple thresholding with true degree of sparsity; (6)-(7) methods from [26], which are referred as SPCA ($k = 2$) and SPCA ($k = 1$); (8)-(9): our sparse PCA methods, which are referred as sPCA-SL (sparse PCA using squared lasso penalty) and sPCA-SL-OC (with orthogonal constraint).

To facilitate comparison with simple thresholding and SPCA, for which there is no automatic way of selecting the degree of sparsity of the PC loading vectors,

the true degree of sparsity is used when applying the sPCA-rSVD, sPCA-SL, and sPCA-SLOC procedures (referred to as the oracle methods below).

Table (9.1) reports the medians of the angles between the extracted loading vectors and the corresponding truth for each procedure, as well as the percentages of correctly/incorrectly identified zero loadings for the loading vectors.

We can see our two methods appear to perform reasonably well and give comparable results with sPCA-rSVD family. Comparing with standard PCA, our methods result in smaller median angles, which suggests that sparsity does improve statistical efficiency. Comparing with SPCA family, our method outperforms in all three measurements.

Table (9.2) reports the comparison results for sparse PCA methods using cross validation. Since standard PCA, simple threshold method, and SPCA family cannot do cross validation, we only do comparison with sPCA-rSVD family. Similarly, we can see our results is comparable, and we can offer both rowwise method and entrywise method.

9.3 Simulation of High-Dimension Case

Example 2. We also considered a covariance matrix with two specified sparse leading eigenvectors. The data are in $\mathbb{R}^p$ with $p = 500$ and generated as $\mathbf{X} \sim N(0, \mathbf{\Sigma}_2)$. Let $\tilde{\mathbf{v}}_1$ and $\tilde{\mathbf{v}}_2$ be two 500-dimensional vectors such that $\mathbf{v}_{1k} = 1, k = 1, ..., 10$, and $\mathbf{v}_{1k} = 0, k = 11, ..., 500$; and $\mathbf{v}_{2k} = 0, k = 1, ..., 10, k = 21, ..., 500$. The first two eigenvectors of $\mathbf{\Sigma}_2$ are chosen to be $\mathbf{v}_1 = \tilde{\mathbf{v}}_1/||\tilde{\mathbf{v}}_1||$ and $\mathbf{v}_2 = \tilde{\mathbf{v}}_2/||\tilde{\mathbf{v}}_2||$. To make these two eigenvectors dominate, we let the eigenvalues be $c_1 = 400$, $c_2 = 300$ and $c_k = 1$ for $k = 3, ..., 500$. The simulation scheme of section 9.1 is used to generate data.

We simulated 100 data sets of size $n = 50$ with $\mathbf{\Sigma}_2$ being the covariance matrix.

Method	$\mathbf{v}_1$			$\mathbf{v}_2$		
	Median Angle	Correct (%)	Incorrect (%)	Median Angle	Correct (%)	Incorrect (%)
	$n = 30$					
PCA	15.05	0.17	0.00	28.83	0.00	1.00
sPCA-rSVD-soft	10.86	92.50	5.00	17.06	71.25	19.17
sPCA-rSVD-hard	7.50	90.50	6.33	17.14	70.50	19.67
sPCA-rSVD-SCAD	11.39	92.00	5.33	15.78	71.50	19.17
Simple	8.10	90.75	6.17	14.41	66.50	22.33
SPCA (k=2)	13.71	91.50	5.83	28.94	67.75	21.67
SPCA (k=1)	28.24	80.25	13.17			
sPCA-SL	13.73	90.25	6.50	14.41	75.00	16.67
sPCA-SL-OC	13.73	90.25	6.50	16.21	76.20	17.36
	$n = 300$					
PCA	4.80	1	0	8.21	0.75	0.00
sPCA-rSVD-soft	2.48	100	0	5.54	98.00	1.50
sPCA-rSVD-hard	2.19	100	0	4.20	98.25	1.17
sPCA-rSVD-SCAD	2.19	100	0	4.54	98.00	1.33
Simple	2.48	100	0	5.88	95.50	3.00
SPCA (k=2)	4.11	100	0	9.95	97.25	2.17
SPCA (k=1)	7.71	100	0			
sPCA-SL	2.76	100	0	4.92	98.50	1
sPCA-SL-OC	3.12	100	0	6.23	96.25	0.50

Table 9.1: Comparison of different methods in low-dimension case with oracle information

Method	$\mathbf{v}_1$			$\mathbf{v}_2$		
	Median Angle	Correct (%)	Incorrect (%)	Median Angle	Correct (%)	Incorrect (%)
	$n = 30$					
sPCA-rSVD-soft	11.91	45.00	2.33	23.28	46.50	12.50
sPCA-rSVD-hard	10.89	62.25	2.33	25.15	52.25	18.17
sPCA-rSVD-SCAD	10.68	45.25	2.50	22.40	43.25	12.83
sPCA-SL-rowwise	13.60	52.50	0.17	21.82	37.00	10.33
sPCA-SL-entrywise	11.72	43.50	0.00	18.52	54.00	15.5
sPCA-SL-OC-entrywise	14.87	43.75	2.50	36.91	70.25	42.33
	$n = 300$					
sPCA-rSVD-soft	2.95	69.00	0.00	6.09	44.00	1.17
sPCA-rSVD-hard	2.83	83.25	0.00	7.47	67.50	2.67
sPCA-rSVD-SCAD	2.83	74.75	0.00	5.90	57.25	1.33
sPCA-SL-rowwise	2.44	80.00	0.00	5.42	52.25	1.00
sPCA-SL-entrywise	2.72	83.25	0.00	5.94	92.00	2.00
sPCA-SL-OC-entrywise	2.58	81.25	0.00	7.33	91.75	5.17

Table 9.2: Comparison of different methods in low-dimension case using cross validation

Method	$\mathbf{v}_1$			$\mathbf{v}_2$		
	Median Angle	Correct (%)	Incorrect (%)	Median Angle	Correct (%)	Incorrect (%)
PCA	19.69	5.79	0.10	20.39	4.60	0.20
sPCA-rSVD-soft	1.36	99.59	20.00	1.66	99.59	20.00
sPCA-rSVD-hard	1.21	99.59	20.00	1.53	99.59	20.00
sPCA-rSVD-SCAD	1.21	99.59	20.00	1.53	99.59	20.00
sPCA-rSVD-soft-CV	1.82	98.97	12.20	1.95	98.89	13.00
sPCA-rSVD-hard-CV	1.98	98.98	11.70	2.14	98.95	11.40
sPCA-rSVD-SCAD-CV	2.05	98.85	10.30	1.85	98.88	11.90
SPCA (k=2)	4.95	99.63	18.00	6.21	99.63	18.00
SPCA (k=1)	44.21	99.43	28.00			
sPCA-SL	1.61	99.61	19.00	1.76	99.86	7.00
sPCA-SL-OC	2.17	99.61	19.00	2.79	99.92	10.80
sPCA-SL-CV-rowwise	2.05	90.41	12.00	2.17	98.05	11.50
sPCA-SL-CV-entrywise	3.21	93.92	3.90	4.78	82.11	3.70

Table 9.3: Comparison of different methods in high-dimension case

All methods used in last section are also performed to these high-dimension data sets with the degree of sparsity being specified as the truth (the oracle method) or by the five-fold CV (for sPCA-SL, sPCA-SL-OC and sPCA-rSVD family only). The results are summarized in Table 9.3.

We can see our two methods appear to perform reasonably well and give comparable results with sPCA-rSVD family. Comparing with standard PCA, our methods result in smaller median angles, which suggests that sparsity does improve statistical efficiency. Comparing with SPCA family, our method outperforms in all three measurements. As for cross-validation, we can see our results is comparable with sPCA-rSVD.

9.4 Simulation of Missing Values Case

In this section, we consider sparse PCA on simulated data matrix with missing values. Given a data matrix $\mathbf{X} : n \times p$ and missing ratio ρ, we randomly drawed $n \times p \times \rho$ location in matrix $\mathbf{X}$ and set these locations as missing. Then we ran the

Missing Ratio	$\mathbf{v}_1$			$\mathbf{v}_2$		
	Median Angle	Correct (%)	Incorrect (%)	Median Angle	Correct (%)	Incorrect (%)
0	3.21	93.92	3.90	4.78	82.11	3.70
0.01	9.59	93.26	5.30	8.57	80.89	6.80
0.05	10.69	92.89	4.60	9.40	80.88	5.10
0.10	9.06	93.65	5.20	8.00	80.92	6.00
0.20	9.35	93.50	4.60	7.72	79.89	5.10
0.40	8.66	95.41	6.30	7.04	79.39	7.10
0.60	14.81	95.61	9.60	11.03	77.23	6.80

Table 9.4: Sparse PCA with missing values for large p small q, tuning parameter is selected via cross validation

algorithm for sparse PCA with missing values and compared with true values.

Note that we did not do comparison with other methods, since there is no paper dealing with data analysis with missing values for sparse PCA.

In table 9.5, we use the same data generation model as in section 9.3. In table 9.5, the model setting is the same, except that $q = 100$, i.e., the first 100 elements for $\mathbf{v}_1$ and second 100 elements for $\mathbf{v}_2$ are nonzero.

From both table, we could see that the recovery is reasonably good when missing ratio is not too large (such as 20% or 40%), which suggests that, in reality, when data matrix is incomplete, we can use our method and can have good result. In addition, as missing ratio gets larger, the median angle gets worse and cross validation tends to choose larger tuning parameter, which yields larger correct rate and incorrect rate.

9.5 Pitprops Data Analysis

The pitprops data, with 180 observations and 13 measured variables, is a classic example showing the difficulty of interpreting PCs. To illustrate the performance of their sparse PCA methods, several authors have studied the pitprops data, such as [26], [20], and [16].

We first compared our method with [16], since both methods could achieve or-

Missing Ratio	$\mathbf{v}_1$			$\mathbf{v}_2$		
	Median Angle	Correct (%)	Incorrect (%)	Median Angle	Correct (%)	Incorrect (%)
0	6.67	70.44	2.90	7.67	64.55	3.51
0.01	6.76	70.58	2.94	7.61	64.62	3.52
0.05	7.02	71.01	3.02	7.82	65.50	3.76
0.10	7.29	71.79	2.96	8.04	66.52	4.06
0.20	7.96	73.37	3.14	8.33	68.88	4.53
0.40	9.40	77.03	4.20	9.56	73.54	6.86
0.60	12.94	81.40	5.55	13.56	79.39	8.53

Table 9.5: Sparse PCA with missing values for large p large q, tuning parameter is selected via cross validation

thogonality. The obtained sparse PC loadings are shown in table 9.6 and table 9.7. We can see our results are quite close to [16]. For explained variance, we can see both are close to the percentages achieved by the classic PCA: 32.4, 18.3, 14.4, 8.5, 7.0, 6.3, respectively.

The sparse PCs produced by SPCA and sPCA-rSVD are unorthogonal. The correlation matrices for [26] and [20] are shown below, from which we an see there exist many significant correlations:

$$\begin{pmatrix} 1 & -0.17 & -0.33 & -0.00 & -0.20 & 0.08 \\ -0.17 & 1 & 0.13 & -0.14 & -0.22 & 0.08 \\ -0.33 & 0.13 & 1 & 0.10 & 0.14 & -0.40 \\ -0.00 & -0.14 & 0.10 & 1 & 0.03 & -0.01 \\ -0.20 & -0.22 & 0.14 & 0.03 & 1 & -0.18 \\ 0.08 & 0.08 & -0.40 & -0.01 & -0.18 & 1 \end{pmatrix},$$

$$\begin{pmatrix} 1 & 0.20 & -0.46 & -0.33 & -0.20 & -0.04 \\ 0.20 & 1 & -0.11 & 0.27 & 0.13 & 0.05 \\ -0.46 & -0.11 & 1 & 0.26 & 0.16 & -0.10 \\ -0.33 & 0.27 & 0.26 & 1 & 0.20 & 0.07 \\ -0.20 & 0.13 & 0.16 & 0.20 & 1 & -0.05 \\ -0.04 & 0.05 & -0.10 & 0.07 & -0.05 & 1 \end{pmatrix}.$$

Variables	PC1	PC2	PC3	PC4	PC5	PC6
Topdiam	0.534	0.005	-0.061	-0.016	0	0.021
Length	0.548	0	-0.065	-0.016	0	0.025
Moist	0	0.346	-0.015	0.006	0	0
Testsg	0	0.349	0	0.017	0	0.001
Ovensg	0	0	0.197	0	-0.103	0
Ringtop	0.173	0.036	0.156	-0.012	0	0.013
Ringbut	0.444	0	0.092	0	0	0.008
Bowmax	0.297	0	0	0.065	0	-0.104
Bowdist	0.438	0	0	0	0	0
Whorls	0.469	-0.019	0	0	-0.383	0
Clear	0	0	0	0	0	0
Lnots	0	0.032	0	-0.238	0	-0.032
Diaknot	0	0	-0.220	0	-0.092	0
Variance (%)	0.302	0.152	0.146	0.085	0.061	0.047
Cum.var. (%)	0.302	0.454	0.600	0.686	0.746	0.794

Table 9.6: Results of sPCA-SL-OC method on pitprops data

Variables	PC1	PC2	PC3	PC4	PC5	PC6
Topdiam	-0.471	0	0.197	0	0	0
Length	-0.484	0	0.222	0	-0.045	0
Moist	0	-0.684	0	0.060	0.261	0
Testsg	0	-0.659	-0.072	0.063	0.189	-0.121
Ovensg	0	0	-0.745	0	0	-0.455
Ringtop	-0.134	0	-0.400	0	-0.137	0.345
Ringbut	-0.383	0	-0.110	0	-0.139	0.299
Bowmax	-0.254	0.137	0	-0.092	0	-0.679
Bowdist	-0.383	0	0	0	-0.080	0
Whorls	-0.410	0.163	0	0.035	0	0
Clear	0	0	0	-0.978	-0.040	-0.091
Lnots	0	-0.229	0	0	-0.921	-0.318
Diaknot	0	0	0.424	0.163	0	0
Variance (%)	0.301	0.156	0.146	0.078	0.065	0.046
Cum.var. (%)	0.301	0.457	0.600	0.666	0.731	0.778

Table 9.7: Results of method in Qi, Luo, and Zhao [16] on pitprops data

10. SUMMARY

In this dissertation we built a comprehensive approach for sparse PCA. Our approach can handle single-layer cases, multi-layer cases using deflation, multi-layer cases with orthogonal constraints, and cases with missing values. We showed methodology connections and comparisons with other methods, we proved convergence and consistency results for these different cases, and we evaluated our methods using simulated and real-world data sets.

In the first two sections we explained the challenges of sparse PCA compared with standard PCA. We investigated recent major papers and compared their methods and that of ours on several key metrics. The result showed that our approach is the most comprehensive one. In section 3 we proposed our main formulation, and justified this formulation via scale invariance property and choice of norms for penalty function. In section 4 and 5 we developed algorithms for all different cases. To solve the SLSA problem, we found its closed-form solution, to solve the SLOCSA problem, we introduced the ADMM and QP algorithms. In section 6 and 7 we proved convergence results for our methods. We developed some theoretical results to show regularity of functions which is the preliminary requirement for proving stationarity. We also proved the convergence of the estimator to the global optima by utilizing the convergence property of the power iteration algorithm. In section 8 we developed a method for cases with missing values, and we showed the convergence of this method using theoretical results developed above. We also proposed a cross-validation method for tuning parameter selection. In section 9 we evaluated our methods using simulated and real-world data sets, which covered both high-dimensional and low-dimensional scenarios.

Our first formulation is equivalent to the one in [16], this does not reduce the contribution of our methods. First of all, this result means the two methods are equivalent w.r.t. stationary equation (or whole solution path), does not mean that they are exactly the same. In fact we started from a totally different philosophy from theirs, therefore the equivalence result showed that our formulation is a good station to connect different approaches. Secondly, our formulation is more promising than theirs. For cases other than single-layer cases, we don't have equivalent results with theirs any more. For cases with orthogonal constraint, their algorithm is not as efficient as ours and they failed to show complete convergence results. For cases with missing values, it is impossible to develop methods based on their formulation.

To solve the SLOCSA problem, we developed two approaches: the ADMM approach and the QP approach. The first one is easy to parallelize and thus efficient for big-data scenarios, the second one is faster in small sample size cases. In simulated and real-world data analysis part, we followed this guideline and used different approachs for different settings.

Our results are specifically designed for sparse PCA problems, however it can be generalized to two-way regularized SVD problems, such as bi-clustering problems, two-way functional data analysis problems, and fMRI data analysis problems. This further showed that our approach is comprehensive and promising.

BIBLIOGRAPHY

[1] Genevera I Allen. Sparse and functional principal components analysis. *arXiv preprint arXiv:1309.2895*, 2013.

[2] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1): 1–122, 2011.

[3] Alexandre d' Aspremont, Francis R Bach, and Laurent El Ghaoui. Full regularization path for sparse principal component analysis. In *Proceedings of the 24th international conference on Machine learning*, pages 177–184. ACM, 2007.

[4] Zhaoping Hong and Heng Lian. Sparse-smooth regularized singular value decomposition. *Journal of Multivariate Analysis*, 117:163–174, 2013.

[5] Jianhua Z Huang, Haipeng Shen, and Andreas Buja. The analysis of two-way functional data using two-way regularized singular value decompositions. *Journal of the American Statistical Association*, 104(488), 2009.

[6] Iain M Johnstone and Arthur Yu Lu. Sparse principal components analysis. *Unpublished manuscript*, 2004.

[7] Iain M Johnstone and Arthur Yu Lu. On consistency and sparsity for principal components analysis in high dimensions. *Journal of the American Statistical Association*, 2012.

[8] Ian T Jolliffe, Nickolay T Trendafilov, and Mudassir Uddin. A modified principal component technique based on the lasso. *Journal of Computational and Graphical Statistics*, 12(3):531–547, 2003.

[9] Michel Journée, Yurii Nesterov, Peter Richtárik, and Rodolphe Sepulchre. Gen-

eralized power method for sparse principal component analysis. *The Journal of Machine Learning Research*, 11:517–553, 2010.

[10] Chenlei Leng and Hansheng Wang. On general adaptive sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 18(1), 2009.

[11] Ronny Luss and Marc Teboulle. Convex approximations to sparse pca via lagrangian duality. *Operations Research Letters*, 39(1):57–61, 2011.

[12] Lester Mackey. Deflation methods for sparse pca. In *NIPS*, volume 21, pages 1017–1024, 2008.

[13] Lester W Mackey. Deflation methods for sparse pca. In *Advances in neural information processing systems*, pages 1017–1024, 2009.

[14] Boaz Nadler. Finite sample approximation results for principal component analysis: A matrix perturbation approach. *The Annals of Statistics*, pages 2791–2817, 2008.

[15] Debashis Paul. Asymptotics of sample eigenstructure for a large dimensional spiked covariance model. *Statistica Sinica*, pages 1617–1642, 2007.

[16] Xin Qi, Ruiyan Luo, and Hongyu Zhao. Sparse principal component analysis by choice of norm. *Journal of multivariate analysis*, 114:127–160, 2013.

[17] Peter Richtarik, Martin Takac, and Selin Damla Ahipasaoglu. Alternating maximization: unifying framework for 8 sparse pca formulations and efficient parallel codes. *arXiv preprint arXiv:1212.4137*, 2012.

[18] Andrzej P Ruszczyński. *Nonlinear optimization*, volume 13. Princeton university press, 2006.

[19] Dan Shen, Haipeng Shen, and JS Marron. Consistency of sparse pca in high dimension, low sample size contexts. *Journal of Multivariate Analysis*, 115: 317–333, 2013.

[20] Haipeng Shen and Jianhua Z Huang. Sparse principal component analysis via

regularized low rank matrix approximation. *Journal of multivariate analysis*, 99 (6):1015–1034, 2008.

[21] Paul Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3): 475–494, 2001.

[22] Daniela M Witten, Robert Tibshirani, and Trevor Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, page kxp008, 2009.

[23] Stephen Wright and Jorge Nocedal. Numerical optimization. *Springer Science*, 35:67–68, 1999.

[24] Dan Yang, Zongming Ma, and Andreas Buja. A sparse svd method for high-dimensional data. *Journal of Computational and Graphical Statistics*, 2013.

[25] Youwei Zhang and Laurent E Ghaoui. Large-scale sparse principal component analysis with application to text data. In *Advances in Neural Information Processing Systems*, pages 532–539, 2011.

[26] Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286, 2006.