2016

Weighted Low-Rank Approximation of Matrices:Some Analytical and Numerical Aspects

Aritra Dutta
University of Central Florida

WEIGHTED LOW-RANK APPROXIMATION OF MATRICES: SOME ANALYTICAL AND NUMERICAL ASPECTS

by

ARITRA DUTTA
B.S. Mathematics, Presidency College, University of Calcutta, 2006
M.S. Mathematics and Computing, Indian Institute of Technology, Dhanbad, 2008
M.S. Mathematical Sciences, University of Central Florida, 2011

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Mathematics
in the College of Sciences
at the University of Central Florida
Orlando, Florida

Fall Term
2016

Major Professors: Xin Li and Qiyu Sun

ABSTRACT

This dissertation addresses some analytical and numerical aspects of a problem of weighted low-rank approximation of matrices. We propose and solve two different versions of weighted low-rank approximation problems. We demonstrate, in addition, how these formulations can be efficiently used to solve some classic problems in computer vision. We also present the superior performance of our algorithms over the existing state-of-the-art unweighted and weighted low-rank approximation algorithms.

Classical principal component analysis (PCA) is constrained to have equal weighting on the elements of the matrix, which might lead to a degraded design in some problems. To address this fundamental flaw in PCA, Golub, Hoffman, and Stewart proposed and solved a problem of constrained low-rank approximation of matrices: For a given matrix $A = (A_1\ A_2)$, find a low rank matrix $X = (A_1\ X_2)$ such that $\text{rank}(X)$ is less than r, a prescribed bound, and $\|A - X\|$ is small. Motivated by the above formulation, we propose a weighted low-rank approximation problem that generalizes the constrained low-rank approximation problem of Golub, Hoffman and Stewart. We study a general framework obtained by pointwise multiplication with the weight matrix and consider the following problem: For a given matrix $A \in \mathbb{R}^{m\times n}$ solve:

$$\min_X \|\,(A - X) \odot W\|_F^2 \text{ subject to } \text{rank}(X) \leq r,$$

where $\odot$ denotes the pointwise multiplication and $\|\cdot\|_F$ is the Frobenius norm of matrices.

In the first part, we study a special version of the above general weighted low-rank approximation problem. Instead of using pointwise multiplication with the weight matrix, we use the regular matrix multiplication and replace the rank constraint by its convex surrogate, the nuclear norm, and consider the following problem:

$$\hat{X} \;=\; \arg\min_X \{\frac{1}{2}\|(A - X)W\|_F^2 + \tau\|X\|_*\},$$

where $\|\cdot\|_*$ denotes the nuclear norm of X. Considering its resemblance with the classic singular value thresholding problem we call it the weighted singular value thresholding (WSVT) problem. As expected, the WSVT problem has no closed form analytical solution in general, and a numerical procedure is needed to solve it. We introduce auxiliary variables and apply simple and fast alternating direction method to solve WSVT numerically. Moreover, we present a convergence analysis of the algorithm and propose a mechanism for estimating the weight from the data. We demonstrate the performance of WSVT on two computer vision applications: background estimation from video sequences and facial shadow removal. In both cases, WSVT shows superior performance to all other models traditionally used.

In the second part, we study the general framework of the proposed problem. For the special case of weight, we study the limiting behavior of the solution to our problem, both analytically and numerically. In the limiting case of weights, as $(W_1)_{ij} \to \infty, W_2 = \mathbb{1}$, a matrix of 1, we show the solutions to our weighted problem converge, and the limit is the solution to the constrained low-rank approximation problem of Golub et. al. Additionally, by asymptotic analysis of the solution to our problem, we propose a rate of convergence. By doing this, we make explicit connections between a vast genre of weighted and unweighted low-rank approximation problems. In addition to these, we devise a novel and efficient numerical algorithm based on the alternating direction method for the special case of weight and present a detailed convergence analysis. Our approach improves substantially over the existing weighted low-rank approximation algorithms proposed in the literature. Finally, we explore the use of our algorithm to real-world problems in a variety of domains, such as computer vision and machine learning.

Finally, for a special family of weights, we demonstrate an interesting property of the solution to the general weighted low-rank approximation problem. Additionally, we devise two accelerated algorithms by using this property and present their effectiveness compared

to the algorithm proposed in Chapter 4.

This thesis is dedicated to my parents Prodip and Bithika Dutta, my grandparents, and my advisers Professor Xin Li and Professor Qiyu Sun.

ACKNOWLEDGMENTS

I would like to express my profound appreciation to my advisers Prof. Xin Li and Prof. Qiyu Sun. I am very fortunate that they agreed to work with me, and since the first day, they took extreme care in my overall growth. It is their sheer genius and patience that they assured my success in completing a Ph.D. They are undoubtedly the greatest teachers I ever had. I would never be able to accumulate enough wealth in my life to ever repay my debt to them.

I would also like to convey a very special thanks and heartiest regards to Prof. Ram Narayan Mohapatra. Without his guidance, pursuing a graduate degree would have been an unfulfilled dream for me. He has been a tremendous mentor for me and his contributions in my life have been countless. Also, I would like to immensely thank my dissertation committee members. Prof. Mubarak Shah, for devoting his precious time to collaborate with me in my research and sharing insightful ideas, and Prof. M. Zuhair Nashed for his great advice and inspiration throughout my graduate life. I also sincerely thank Dr. Boqing Gong for his time and willingness to collaborate with me.

I would like to express my sincere and greatest regards to my parents. Without their constant motivation and inspiration, this work would have never come to fruition. My mother stayed awake for many long nights as I did in the past years. With their struggle, honesty, selflessness, and dedication they created a living example in my life. There are no words which can glorify their contribution in my life.

I want to give a special thanks to my dear brother Amitava, who has always been with me through trials and tribulations. In the past two years, his constant inspiration immensely helped me to keep my head straight and focused. In this scope, I would also like to thank my few very good friends, Dr. Aniruddha Dutta, Donald Porchia, Dr. Eugene Martinenko, Dr. Rizwan Arshad Ashraf, Dr. Bernd Losert, Dr. Shruba Gangopadhyay, Dr. Kamran

Sadiq, and Sanjit Kumar Roy. To be very specific, Aniruddha was the one who guided me through the process of pursuing a graduate degree, and he is the reason I decided to decline my offer from Auburn and accept my admission to UCF. All of these great people made my life complete with their wisdom and I learned a great deal from each of them in every aspect of my life, and the process is still ongoing. In this journey, I would also like to thank a special person in my life, Cintya Nirvana Larios, for her extreme kindness, patience, and love.

Last but not the least, I would like to thank two very dear friends of mine, Dr. Afshin Dehghan for his invaluable lessons in programming and Mr. Pawan Kumar Gupta for being a great companion in the past few years. At the end, I would like to thank some very special teachers from my high school and undergraduate career, for giving me free lessons day after day. Without their support, I probably would have discontinued studying. They are Late Mr. Mohanlal Sinha Roy, Mr. Rabindranath Ghatak, Mr. Subal Kumar Bose, Mr. Dwibedi, Mr. Gurudas Bajani, Mr. Biswanath Sengupta, and Mr. Dilip Shyamal. My life has always been influenced and inspired by them.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE: INTRODUCTION

In today's world, data generated from diverse scientific fields are high-volume and increasingly complex in nature. According to a report in 2004, the new data stored in digital media devices have increased to 92% in 2002, and the size of these new data is more than 5 exabytes [52]. This can be attributed by the fact that it is always easier to generate more data than finding useful information from the data. However, in many cases, the high-dimensional data points are constrained to a much lower dimensional subspace. Therefore, in the analysis and understanding of high-dimensional data, a major research challenge is to extract the most important features from the data by reducing its dimension. Dimension reduction techniques refer to the process of imposing structure to a data having large dimensions into a data with much lesser dimensions, while ensuring minimal information loss. The problem of dimensionality reduction arises in many applications, such as, image processing, machine learning, computer vision, bioinformatics data analysis, and web data ranking. In order to get storage and computation-efficient prediction models from a big data set, low rank approximation of matrices has become one of the most eminent tools. Low-rank matrix approximation is a multidisciplinary field involving mathematics, statistics, and optimization. It is widely applicable in high-dimensional data processing and analysis. In this study, we consider the given data points to be arranged in the columns of a matrix and there exists a much lower dimensional linear subspacial structure to represent it. The goal of dimensionality reduction is to find a low-rank matrix that guarantees a good approximation of the data matrix with high accuracy. Depending on the nature of the measurements of the discrepancy between the data matrix and its low rank approximation, there are several well known classical algorithms.

For an integer $r \leq \min\{m, n\}$ and a matrix $A \in \mathbb{R}^{m\times n}$, the standard low rank approximation problem can be defined as an approximation to A by a rank r matrix under the

Frobenius norm as follows:

$$\min_{\substack{X\in\mathbb{R}^{m\times n}\\ \mathrm{r}(X)\leq r}} \|A-X\|_F^2, \tag{1.1}$$

where $\mathrm{r}(X)$ denotes the rank of the matrix X and $\|\cdot\|_F$ denotes the Frobenius norm of matrices (see, more discussion in Section 1.1.2). This is also referred to as Eckart-Young-Mirsky's theorem [38] and is closely related to the principal component analysis (PCA) method in statistics [35]. Conventionally, if the given data are corrupted by the i.i.d. Gaussian noise, classical PCA is used. However, it is a well-known fact that the solution to the classical PCA problem is numerically sensitive to the presence of outliers in the matrix. In other words, if the matrix A is perturbed by one single large value at one entry, the explicit formula for its low-rank approximation would yield a much different solution than the unperturbed one. This phenomenon may be attributed to the use of the Frobenius norm. To address different nature of corrupted entries in the data matrix, different norms have been proposed to use. For example, ℓ_1 norm does encourage sparsity when the norm is made small. Therefore, to solve the problem of separating the sparse outliers added to a low-rank matrix, Candes et al. ([32]) argued to replace the Frobenius norm in the SVT problem by the ℓ_1 norm and formulated the following (see also [9]):

$$\min_{\substack{X\in\mathbb{R}^{m\times n}\\ \mathrm{r}(X)\leq r}} \|A-X\|_{\ell_1}, \tag{1.2}$$

which unlike PCA, does not assume the presence of uniformly distributed noise, rather it deals with sparse large errors or outliers in the data matrix. This is referred to as robust PCA (RPCA) [9]. Later (in Sections 1.2 and 1.3) we will discuss the motivation and formulation behind forming the unconstrained versions of (1.1) and (1.2), and their solutions in great detail.

The idea of working with a weighted norm is very natural in solving many engineering problems. For example, if SVD is used in quadrantally-symmetric two-dimensional (2-D) filter design, as pointed out in ([37, 29, 30]), it might lead to a degraded construction in

some cases as it is not able to discriminate between the important and unimportant components of A. Similarly in many real world applications, one has good reasons to keep certain entries of A unchanged while looking for a low-rank approximation. To address this problem, a weighted least squares matrix decomposition (WLR) method was first proposed by Shpak [30]. Following his idea of assigning different weights to discriminate between important and unimportant components of the test matrix, Lu, Pei, and Wang ([29]) designed a numerical procedure to find the best rank r approximation of the matrix A in the *weighted* Frobenius norm sense:

$$\min_{\substack{X\in\mathbb{R}^{m\times n}\\ \mathrm{r}(X)\le r}} \|(A-X)\odot W\|_F^2, \tag{1.3}$$

where $W \in \mathbb{R}^{m\times n}$ is a weight matrix and $\odot$ denotes the element-wise matrix multiplication (Hadamard product). In 2003, Srebro and Jaakkola ([39]) proposed and solved a problem similar to (1.3) by using a matrix factorization technique: for a given matrix $A \in \mathbb{R}^{m\times n}$ find

$$\min_{\substack{U\in\mathbb{R}^{m\times r}, V\in\mathbb{R}^{n\times r}\\ X=UV^T\in\mathbb{R}^{m\times n}}} \|(A-X)\odot W\|_F^2, \tag{1.4}$$

where $W \in \mathbb{R}_+^{m\times n}$ is a weight matrix with positive entries. This is the weighted low rank approximation problem studied first when W is an indicator weight for dealing with the missing data case ([40, 41]) and then for more general weight in machine learning, collaborative filtering, 2-D filter design, and computer vision [39, 43, 45, 37, 29, 30]. At about the same time, Manton, Mahony and Hua ([37]) proposed a problem with a more generalized weighted norm:

$$\min_{\substack{X\in\mathbb{R}^{m\times n}\\ \mathrm{r}(X)\le r}} \|A-X\|_Q^2, \tag{1.5}$$

where $Q \in \mathbb{R}^{mn\times mn}$ is a symmetric and positive definite weight matrix, $\|A-X\|_Q^2 := \mathrm{vec}(A-X)^T Q \mathrm{vec}(A-X)$, which is more general than the norm $\|X\|_Q^2 = \mathrm{trace}(X^TQX)$, and $\mathrm{vec}(\cdot)$ is an operator which maps the entries of $\mathbb{R}^{m\times n}$ to $\mathbb{R}^{mn\times 1}$ by stacking the columns.

In computer vision shape and motion from image streams (SfM) [69], non-rigid SfM can be solved using a matrix factorization with missing components. The standard formulation of the problem as defined in [68, 67] is

$$\min_{X,Y} f(X,Y) := \min_{X,Y} \|A - XY\|_F^2, \tag{1.6}$$

where $A \in \mathbb{R}^{m\times n}$ is the given noiseless (or corrupted by Gaussian noise) matrix of rank r, to be factored in two matrices $X \in \mathbb{R}^{m\times r}, Y \in \mathbb{R}^{r\times n}$. The solution to (1.6) can be obtained using SVD. However, if some entries of A are missing then to minimize $f(X,Y)$ with respect to the existing components of A one has to minimize [68, 67]:

$$\min_{X,Y} f(X,Y) := \min_{X,Y} \|(A - XY) \odot W\|_F^2, \tag{1.7}$$

where $W \in \mathbb{R}^{m\times n}$ is a selector matrix such that

$$w_{ij} = \begin{cases} 1, & \text{if } m_{ij} \text{ exists} \\ 0, & \text{otherwise.} \end{cases}$$

Note that the problem (1.7) is equivalent to (1.4). Solving (1.7) requires iterative computation as defined in [45, 70, 71, 40, 68, 72] and by many others. In 2006, Okatani and Deguchi proposed a low-rank matrix approximation in the presence of missing data, which is also known as principal component analysis with missing data [40] and can be written using two equivalent formulations as follows:

$$\min_{X,Y} f(X,Y) := \min_{X,Y} \|(A - XY) \odot W\|_F^2, \tag{1.8}$$

and

$$\min_{X,Y,\mu} f'(X,Y,\mu) := \min_{X,Y,\mu} \|(A - XY - \mathbb{1}_m\mu^T) \odot W\|_F^2, \tag{1.9}$$

where $W \in \mathbb{R}^{m\times n}$ is the indicator matrix as in (1.7), $\mathbb{1} \in \mathbb{R}^m$ is a vector of 1, and $\mu \in \mathbb{R}^n$ is the mean-vector. The problems (1.7) and (1.8) are equivalent to (1.9) in the sense that with

slight modifications one can use the solutions to (1.7) and (1.8) for solving (1.9). Oaktani and Deguchi used the classical Wiberg algorithm [41] to solve (1.7).

So far we have presented some classic unweighted and weighted low-rank approximation problems and briefly mentioned their use in real world applications. We will explain their solutions later in Chapter 1. Starting from the next section of this chapter, we will discuss the background material and quote some useful classical results pertinent to the thesis. The rest of the thesis is organized as follows. In Chapter 2, we propose an elementary treatment (without using advanced tools of convex analysis) to the shrinkage function and show how naturally the shrinkage function can be used in solving more advanced problems. In Chapter 3, we propose and solve a weighted low-rank approximation problem motivated by the work of Golub, Hoffman, and Stewart on a problem of constrained low-rank approximation of matrices. We compare, in addition, the performance of our algorithm over other state-of-art rank minimization algorithms on some real world computer vision applications. In Chapter 4, we study a more generalized version of the problem as proposed in Chapter 3 and analytically discuss the convergence of its solution to that of Golub, Hoffman, and Stewart in the limiting case of weight. A numerical algorithm with detailed convergence analysis is also presented. Finally, in Chapter 5, an accelerated version of weighted low-rank approximation algorithm is discussed for a special family of weights.

1.1 Technical Background

In this section, we provide a detailed technical discussion of some classical results that are frequently used in this thesis. They had previously been proved and used in several different articles and journals. In order to have a better understanding, here we present them in great detail. Some results are rephrased in an elaborated manner so that the reader can understand the motivation behind them.

1.1.1 Notations

In this section we list some frequently used notations. Other less frequently used notations will be notations will be defined when they are used. We denote A as the given matrix and a_{ij} as (i,j)-th entry of A. The standard inner product of two matrices (vectors) is denoted by $\langle \cdot, \cdot \rangle$. A matrix norm is denoted by $\|\cdot\|$ unless specified and $\|\cdot\|^*$ is the corresponding dual norm. Using trace(A) or tr(A) we denote the sum of the diagonal entries of the matrix A. The inner product of two matrices X and Y is defined as $\langle X, Y \rangle = \text{trace}(X^T Y)$ and the Frobenius norm by $\|X\|_F = \sqrt{\text{trace}(X^T X)}$. The regular ℓ_1-norm is denoted by $\|X\|_{\ell_1} = \sum_{i,j} |x_{ij}|$. The Euclidean norm on $\mathbb{R}^m$ is denoted by $\|\cdot\|_{\mathbb{R}^m}$. Note that if $A \in \mathbb{R}^{m \times n}$ then the matrix operator norm can be defined as $\|A\| = \max_{\|x\|_{\mathbb{R}^n} \leq 1} \|Ax\|_{\mathbb{R}^m}$. By conv$\{A\}$ we denote the convex hull of the set A. We adopt the notation $a = \arg\min_{x \in A} f(x)$ to mean that $a \in A$ is a solution of the minimization problem $\min_{x \in A} f(x)$ and by dom$f$ we denote the domain of the function f. We use ∇f to denote the gradient of the function f.

1.1.2 Definitions

In this section we will quote some useful definitions.

Dual Norm [46] The dual norm of a matrix norm $\|\cdot\|$ induced on a matrix $A \in \mathbb{R}^{m \times n}$ is defined as

$$\|A\|^* = \max_{\substack{B \in \mathbb{R}^{m \times n} \\ \|B\| \leq 1}} (\text{trace}(B^T A)).$$

Subdifferential of a Matrix Norm [46] The subdifferential (or the set of subgradients) of a matrix $A \in \mathbb{R}^{m \times n}$ is defined as

$$\partial \|A\| = \{G \in \mathbb{R}^{m \times n} : \|B\| \geq \|A\| + \text{trace}((B - A)^T G), \text{ for all } B \in \mathbb{R}^{m \times n}\}. \tag{1.10}$$

The above definition is equivalent to

$$\partial \|A\| = \{G \in \mathbb{R}^{m \times n} : \|A\| = \text{trace}(G^T A) \text{ and } \|G\|^* \leq 1\}, \tag{1.11}$$

and can be proved using the following argument: In (1.10), since the choice of $B \in \mathbb{R}^{m\times n}$ is arbitrary consider $B = 2A$ in (1.10) and we find

$$\begin{aligned} \|2A\| &\geq \|A\| + \text{trace}((2A - A)^T G), \\ \text{which implies, } \|A\| &\geq \text{trace}(A^T G). \end{aligned} \tag{1.12}$$

Next, substituting $B = 0$ in (1.10) yields

$$\|A\| \leq \text{trace}(A^T G). \tag{1.13}$$

Combining (1.12) and (1.13) we have

$$\|A\| = \text{trace}(G^T A).$$

Using $\|A\| = \text{trace}(G^T A)$ in (1.10) we find

$$\begin{aligned} \|B\| &\geq \|A\| + \text{trace}((B - A)^T G) \\ &= \text{trace}(A^T G) + \text{trace}(B^T G) - \text{trace}(A^T G) \\ &= \text{trace}(B^T G). \end{aligned} \tag{1.14}$$

If $\|B\| \leq 1$ then $\text{trace}(B^T G) \leq \|B\| \leq 1$ and that implies $\|G\|^* \leq 1$ (using (1.11)). Therefore $\partial\|A\| \subset \{G \in \mathbb{R}^{m\times n} : \|A\| = \text{trace}(G^T A) \text{ and } \|G\|^* \leq 1\}$. On the other hand $\|G\|^* \leq 1$ implies $\text{trace}(B^T G) \leq 1$ as $\|B\| \leq 1$. Therefore for all $B \in \mathbb{R}^{m\times n}$, $\text{trace}(\frac{B^T}{\|B\|} G) \leq 1$, which implies $\text{trace}(B^T G) \leq \|B\|$. Finally we have

$$\begin{aligned} &\|B\| - \|A\| \\ &= \|B\| - \text{trace}(A^T G) \\ &= \|B\| - \text{trace}((A - B + B)^T G) \\ &= \|B\| + \text{trace}((B - A)^T G) - \text{trace}(B^T G) \\ &\geq \text{trace}((B - A)^T G). \end{aligned}$$

Therefore, $\{G \in \mathbb{R}^{m\times n} : \|A\| = \text{trace}(G^T A) \text{ and } \|G\|^* \leq 1\} \subset \partial\|A\|$. Hence the sets defined in (1.10) and (1.11) are equal and we proved the expressions (1.10) and (1.11) are equivalent.

Some Basic Properties of Subdifferential. Let ∂f be a subdifferential of a convex function f at $x \in \text{dom} f$. Then ∂f posses the following properties:

1. $f(x) + \langle \partial f, y - x \rangle$ is a global lower bound on $f(y)$ for all $y \in \text{dom} f$.

2. ∂f is a closed convex set.

3. If $x \in \text{int}(\text{dom} f)$ then ∂f is nonempty and bounded.

4. $\partial f(x) = \{\nabla f(x)\}$ if f is differentiable at x.

5. If $h(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x)$ with $\alpha_1, \alpha_2 \geq 0$, then $\partial h(x) = \alpha_1 \partial f_1(x) + \alpha_2 \partial f_2(x)$.

6. Let $h(x) = f(Ax + b)$ be an affine transform of f. Then $\partial h(x) = -A^T \partial f(Ax + b)$.

Operator Norm [46, 55, 56] The operator norm of a matrix $A \in \mathbb{R}^{m \times n}$ is defined as

$$\|A\| = \max_{\|x\|_{\mathbb{R}^n} \leq 1} \|Ax\|_{\mathbb{R}^m}.$$

N.B. [46] We can choose two vectors $v \in \mathbb{R}^n$ and $w \in \mathbb{R}^m$ and define $u := \frac{Av}{\|A\|}$, $u \in \mathbb{R}^m$, with $\|u\| = 1$. Thus, $\{v, w\}$ are the member of the set $\Phi(A)$, where

$$\Phi(A) = \{v \in \mathbb{R}^n, w \in \mathbb{R}^m : \|v\|_{\mathbb{R}^n} = 1, \frac{Av}{\|A\|} = u, \|u\|_{\mathbb{R}^m} = 1, w \in \partial \|u\|_{\mathbb{R}^m}\}.$$

Singular Value Decompositions and Matrix Norms [55, 56] Let $A \in \mathbb{R}^{m \times n}$ and $A = U\tilde{A}V^T$ be a singular value decomposition (SVD) of A with $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ being two orthogonal matrices (that is, $U^{-1} = U^T$ and $V^{-1} = V^T$) and $\tilde{A} = \text{diag}(\sigma_1(A)\ \sigma_2(A) \cdots \sigma_{\min\{m,n\}}(A))$ being a diagonal matrix with $\sigma_1(A) \geq \sigma_2(A) \geq \cdots \geq \sigma_{\min\{m,n\}}(A) \geq 0$. The $\sigma_i(A)$'s are called the singular values of A. It is known ([56]) that every matrix in $\mathbb{R}^{m \times n}$ has a SVD and that SVD of a matrix is not unique. Then the nuclear norm of A is given by

$$\|A\|_* = \sum_{i=1}^{\min\{m,n\}} \sigma_i(A),$$

and we can also define the Frobenius norm of A as

$$\|A\|_F = \left(\sum_{i=1}^{\min\{m,n\}} (\sigma_i(A))^2 \right)^{1/2}.$$

This norm turns out to be the same as the ℓ_2 norm of A, treated as a vector in $\mathbb{R}^{mn\times 1}$. Since the nonzero singular values $\sigma_i(A)$'s are exactly the square root of the nonzero eigenvalues of AA^T or A^TA. So,

$$\begin{aligned}
\|A\|_{l_2}^2 = \sum_{i=1}^{m}\sum_{j=1}^{n}(a_{ij})^2 &= \text{trace}(AA^T) \\
&= \text{trace}((U\tilde{A}V^T)(V\tilde{A}^TU)) \\
&= \text{trace}(U\tilde{A}\tilde{A}^TU^T) \\
&= \text{trace}(\tilde{A}\tilde{A}^T) \\
&= \sum_{i=1}^{\min\{m,n\}} (\sigma_i(A))^2,
\end{aligned}$$

and we have also used the fact that $\text{trace}(AB) = \text{trace}(BA)$ for any square matrices A and B. Finally we define the spectral norm of A as the square root of the maximum eigenvalue of the matrix AA^T or AA^T and write

$$\|A\|_2 = \sqrt{\max_{1\le i\le\{m,n\}} \lambda_i(AA^T)},$$

where λ_i's are the eigenvalues of AA^T. The spectral norm can also be viewed as the maximum singular value of A and can be written using the notation defined above as

$$\|A\|_2 = \sigma_1(A).$$

We can state the following simple fact about the nuclear norms of a matrix and that of its diagonal: Let $D(A)$ denote the diagonal matrix using the diagonal of A. We have

$$\|D(A)\|_* \le \|A\|_*. \tag{1.15}$$

This inequality can be verified by using a SVD of $A = U\tilde{A}V^T$ as follows. Write $U = (u_{ij})$, $V = (v_{ij})$, and $t = \min\{m, n\}$. Then

$$\|D(A)\|_* = \|D(U\tilde{A}V^T)\|_* = \sum_{i=1}^{t} \left| \sum_{j=1}^{t} \sigma_j(A) u_{ij} v_{ij} \right| \leq \sum_{j=1}^{t} \sigma_j(A) \sum_{i=1}^{t} |u_{ij} v_{ij}|$$

$$\leq \sum_{j=1}^{t} \sigma_j(A) \cdot \left(\sum_{i=1}^{t} |u_{ij}|^2 \right)^{1/2} \left(\sum_{i=1}^{t} |v_{ij}|^2 \right)^{1/2} \leq \sum_{j=1}^{t} \sigma_j(A) = \|A\|_*,$$

where we have used the Cauchy-Schwarz inequality, and the orthogonality of U and V (so that $\sum_{i=1}^{t} |u_{ij}|^2 \leq 1$ and $\sum_{i=1}^{t} |v_{ij}|^2 \leq 1$) in the second inequality.

Symmetric Gauge Function [46] Using the notations used in the previous definition let $A = U\tilde{A}V^T$ be a SVD of A. Define $\|A\| := \phi(\sigma(A))$, where $\sigma(A)$ is a vector containing the singular values of A and $\phi : \mathbb{R}^{\min\{m,n\}} \to \mathbb{R}$ is known as a symmetric gauge function. By the property of symmetric gauge function [46] we have

$$\phi\left(\epsilon_1 x_{i_1}, \epsilon_2 x_{i_2}, \ \ldots\ , \epsilon_n x_{i_{\min\{m,n\}}}\right) = \phi(x),$$

where $\epsilon_i = \pm 1$, for all i, and $\{i_1, i_2, \ldots i_{\min\{m,n\}}\}$ is a permutation of the set $\{1, 2, \cdots, \min\{m, n\}\}$. One can define different symmetric gauge function to denote different matrix norms (associated with SVD of a matrix). For example, if $\phi(\sigma) := \|\sigma(A)\|_1$, then it is the nuclear norm of A, if $\phi(\sigma) := \|\sigma(A)\|_\infty$, then it denotes the spectral norm of A and, so on.

Shrinkage Function [57, 58] The shrinkage function $S_\lambda(\cdot)$, first introduced by Donoho and Johnstone in their landmark paper [57], (see also [58]) on function estimation using wavelets in the early 1990's. Recently, the shrinkage function has been heavily used in the solutions of several optimization and approximation problems of matrices (see, e.g., [9, 44, 48, 65]).

Let $\lambda > 0$ be fixed. For each $a \in \mathbb{R}$, the shrinkage function $S_\lambda(a)$, is defined as

$$S_\lambda(a) = \begin{cases} a - \lambda, & a > \lambda \\ 0, & |a| \leq \lambda \\ a + \lambda, & a < -\lambda \end{cases} .$$

Remark. The function $S_\lambda(\cdot)$ defined above is called the shrinkage function (also referred to as soft shrinkage or soft threshold, [57, 58]). One may imagine that $S_\lambda(a)$ "shrinks" a to zero when $|a| \leq \lambda$. A plot of $S_\lambda(\cdot)$ for $\lambda = 1$ is given in Figure 1.1.

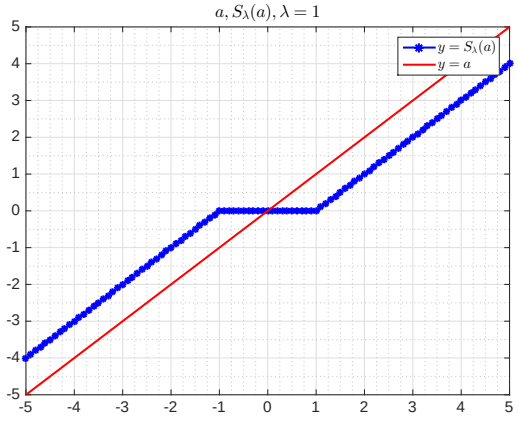

Figure 1.1: A plot of S_λ for $\lambda = 1$.

Elementwise Shrinkage Function [44, 60] For $\mu > 0$ and $X = (x_{ij}) \in \mathbb{R}^{m \times n}$ the element-wise shrinkage function can be defined as

$$(S_\mu(X))_{ij} := \max\{abs(X_{ij} - \mu), 0\}.\text{sign}(X_{ij})$$

where abs(.) and sign(.) are the absolute value and sign functions respectively.

Singular Value Thresholding [48] Let $X \in \mathbb{R}^{m \times n}$ be a matrix of rank $r \leq \min\{m, n\}$ and $X = U\Sigma V^T$ be a singular value decomposition of X. The soft-thresholding operator D_τ is defined as follows [48, 64]: for each $\tau \geq 0$,

$$D_\tau(X) := UD_\tau(\Sigma)V^T,$$

where $D_\tau(\Sigma) = diag\{(\sigma_i - \tau)_+\}$[1] and t_+ is defined as $t_+ = max\{0, t\}$. This is also referred to as *singular value shrinkage operator*. On the other hand, let $X = U_r \Sigma_r V_r^T$ be a rank r SVD of X such that $U_r \in \mathbb{R}^{m\times r}$ and $V_r \in \mathbb{R}^{n\times r}$ are column orthonormal matrices ($U_r^T U_r = I_r$ and $V_r^T V_r = I_r$) and $\Sigma_r \in \mathbb{R}^{r\times r}$ is a diagonal matrix containing the first r non zero singular values of X arranged in a non-increasing order along the diagonal. With the notations defined above one can define the *soft-thresholding operator* D_τ as following: For each $\tau \geq 0$,

$$D_\tau(X) := U_r D_\tau(\Sigma_r) V_r^T.$$

Unitarily Invariant Norms [46, 55, 56] Let $A \in \mathbb{R}^{m\times n}$ be a given matrix. The class of norms $\|\cdot\|$ are said to be unitarily invariant if

$$\|UAV\| = \|A\| \text{ for all orthogonal matrices } U \in \mathbb{R}^{m\times m}, V \in \mathbb{R}^{n\times n}.$$

Note that, the Frobenius norm, nuclear norm, and spectral norm are examples of unitary invariant matrix norms.

1.1.3 Lagrange Multiplier Method and Duality [19]

Consider the standard form of optimization problem (not necessarily convex) as:

$$\begin{aligned}
&\text{minimize} f_0(x)\\
&\text{subject to } f_i(x) \leq 0,\ i = 1, 2, \cdots, m,\\
&h_i(x) = 0,\ i = 1, 2, \cdots, p,
\end{aligned}$$

where $x \in \mathbb{R}^n, \mathcal{D}$ be the domain of f, and $p^* = \arg\min_x f_0(x)$. Note that the function $f_0(x)$ is the objective function and f_i, h_i's are the constriant functions. The Lagrange multiplier method is to form a function $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ which is a weighted sum of the objective and constriant functions such that $\text{dom}\mathcal{L} = \mathcal{D} \times \mathbb{R}^m \times \mathbb{R}^p$, and define $\mathcal{L}$ as

$$\mathcal{L}(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x),$$

[1] σ_i's are the singular values of X.

where $\lambda_i \geq 0$ is Lagrange multiplier associated with $f_i(x) \leq 0$ and ν_i is lagrange multiplier associated with $h_i(x) = 0$. Denote

$$\psi_P(x) = \sup_{\lambda \geq 0, \nu} \mathcal{L}(x, \lambda, \nu)$$

as the primal problem. If x violates any of the primal constraints, that is, $f_i(x) > 0$ or $h_i(x) \neq 0$ for any i then $\psi_P(x) = \infty$. On the other hand, if x satisfies primal constraints then $\psi_P(x) = f_0(x)$. Therefore,

$$\psi_P(x) = \begin{cases} f_o(x), & \text{if } x \text{ is primal feasible} \\ 0, & \text{otherwise} \end{cases}.$$

An equivalent unconstrianed minimization problem can be written as:

$$\inf_x \psi_P(x) = \inf_x \sup_{\lambda \geq 0, \nu} \mathcal{L}(x, \lambda, \nu).$$

Next define $\psi_D : \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$, where D stands for dual and denote

$$\begin{aligned} \psi_D(\lambda, \nu) &= \inf_{x \in \mathcal{D}} \mathcal{L}(x, \lambda, \nu) \\ &= \inf_{x \in \mathcal{D}} (f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x)). \end{aligned}$$

Note that, ψ_D is concave, because it is the point-wise infimum of a collection of affine functions in x. It is easy to see that, if $\lambda \geq 0$, then $\psi_D(\lambda, \nu) \leq p^*$.

1.1.4 Smooth Minimization of Non-Smooth Functions [73]

Consider the following optimization problem [73]:

$$f^* = \arg\min_x \{f(x) : x \in Q_1\}, \tag{1.16}$$

where Q_1 is a closed and bounded convex set in a finite dimensional real vector space E_1 and $f(x)$ is a continuous (not necessarily smooth) convex function on Q_1. Note that $f(x)$ is not necessarily differentiable everywhere on Q_1. The problem (1.16) can be modified as:

$$f(x) = \hat{f}(x) + \max_u \{\langle Ax, u \rangle - \hat{\phi}(u) : u \in Q_2\},$$

where $\hat{f}(x)$ is continuous and convex on Q_1, Q_2 is a closed and bounded convex set in a finite dimensional real vector space E_2, $\hat{\phi}(u)$ is a continuous convex function on Q_2, and $A : E_1 \to E_2$ is a linear operator. Therefore,

$$\begin{aligned} \min_x f(x) &= \min_{x\in Q_1}\{\hat{f}(x) + \max_{u\in Q_2}\{\langle Ax, u\rangle - \hat{\phi}(u)\} \\ &= \max_{u\in Q_2}\{-\hat{\phi}(u) + \min_{x\in Q_1}\{\langle Ax, u\rangle + \hat{f}(x)\}\}. \end{aligned}$$

If $\phi(u) = -\hat{\phi}(u) + \min_{x\in Q_1}\{\langle Ax, u\rangle + \hat{f}(x)\}$ the dual of (1.16) is

$$\max_u\{\phi(u) : u \in Q_2\}. \tag{1.17}$$

An Inequality [73] For a positive parameter μ let the function $f_\mu(x)$ be

$$f_\mu(x) := \max_u\{\langle Ax, u\rangle - \hat{\phi}(u) - \mu d_2(u) : u \in Q_2\},$$

where $d_2(u)$ is a continuous and strongly convex function on Q_2. Denote

$$u_0 := \arg\min_u\{d_2(u) : u \in Q_2\}.$$

Note that if $A(x)$ and $B(x)$ are two functions defined on a set X then

$$\max_{x\in X}\{A(x) + B(x)\} \le \max_{x\in X} A(x) + \max_{x\in X} B(x).$$

[2] Therefore,

$$\begin{aligned} f_\mu(x) &= \max_u\{\langle Ax, u\rangle - \hat{\phi}(u) - \mu d_2(u)\} \\ &\ge \max_u\{\langle Ax, u\rangle - \hat{\phi}(u)\} - \mu \max_u d_2(u). \end{aligned} \tag{1.18}$$

Denote $D_2 := \max_u\{d_2(u) : u \in Q_2\}$ and $f_0(x) := \max_u\{\langle Ax, u\rangle - \hat{\phi}(u)\}$. So, (1.18) yields

$$f_\mu(x) + \mu D_2 \ge f_0(x). \tag{1.19}$$

[2] $\sup_{x\in X}(A(x) + B(x)) \ge \sup_{x\in X}(A(x) + \sup_{y\in X} B(y)) = \sup_{x\in X} A(x) + \sup_{y\in X} B(y)$

Since $d_2(u) \geq 0$ we have

$$\begin{aligned} \langle Ax, u\rangle - \hat{\phi}(u) - \mu d_2(u) &\leq \langle Ax, u\rangle - \hat{\phi}(u), \\ \text{which implies, } \max_u\{\langle Ax, u\rangle - \hat{\phi}(u) - \mu d_2(u)\} &\leq \max_u\{\langle Ax, u\rangle - \hat{\phi}(u)\}, \\ \text{and finally, } f_\mu(x) &\leq f_0(x). \end{aligned} \tag{1.20}$$

Combining (1.19) and (1.20) together we have

$$f_\mu(x) \leq f_0(x) \leq f_\mu(x) + \mu D_2.$$

Let $f(A) = \|A\|_*$ be a non smooth function. By adopting Nesterov's smoothing technique, Ayabat et.al. [74] defined a smooth $C^{1,1}$ variant $f_\mu(A)$ of the original function $f(A)$, denoted by $f_\mu(A) := \max_{W\in\mathbb{R}^{m\times n}, \|W\|\leq 1}\{\langle A, W\rangle - \frac{\mu}{2}\|W\|_F^2\}$. By using the smoothing technique it can be shown that,

$$f_\mu(A) + \frac{\mu}{2}\max_{\substack{W\in\mathbb{R}^{m\times n}\\ \|W\|\leq 1}} \|W\|_F^2 \geq \max_{\substack{W\in\mathbb{R}^{m\times n}\\ \|W\|\leq 1}} \langle A, W\rangle = \|A\|_*.$$

And finally,

$$f_\mu(A) \leq \|A\|_* \leq f_\mu(A) + \frac{\mu}{2}\max_{\substack{W\in\mathbb{R}^{m\times n}\\ \|W\|\leq 1}} \|W\|_F^2.$$

1.1.5 Classic Results on Subdifferentials of Matrix Norm

In this section, we will discuss some useful results and theorems. The first theorem is due to G.A. Watson [46] which gives an expression for directional derivative of any unitary invariant matrix norm $\|\cdot\|$, in terms of the singular value decomposition (SVD) of the matrix. The second theorem is also due to Watson [46], which helps us to obtain a more general representation of the subdifferential of a matrix norm in terms of its SVD. The next two theorems are due to operator norms. Additionally, we present some of useful examples, which indeed explain the use of the main results in this section.

Theorem 1. *[46] Let $U\Sigma V^T$ be a SVD of $A \in \mathbb{R}^{m\times n}$. Without loss of generality consider $m \geq n$. The columns of $U(V)$ are denoted as $u_i(v_i)$ and σ_i be the ith singular value of the matrix A. If $R \in \mathbb{R}^{m\times n}$, then*

$$\lim_{\gamma\to 0+} \frac{\|A+\gamma R\| - \|A\|}{\gamma} = \max_{d\in\partial\phi(\sigma)} \sum_{i=1}^{n} d_i {u_i}^T R v_i \tag{1.21}$$

Proof. Let A depend smoothly on the parameter γ and denote it as $A(\gamma)$. We will show how the change in γ influences the change of the singular values and the singular vectors of $A(\gamma)$. Write,

$$A(\gamma)v_i(\gamma) = \sigma_i(\gamma)u_i(\gamma), \tag{1.22}$$

which on differentiating with respect to γ and then premultiplying by $u_i^T(\gamma)$ yields

$$u_i^T(\gamma)\frac{\partial A(\gamma)}{\partial\gamma}v_i(\gamma) + u_i^T(\gamma)A(\gamma)\frac{\partial v_i(\gamma)}{\partial\gamma} = u_i^T(\gamma)\frac{\partial\sigma_i(\gamma)}{\partial\gamma}u_i(\gamma) + \sigma_i(\gamma)u_i^T(\gamma)\frac{\partial u_i(\gamma)}{\partial\gamma} \tag{1.23}$$

Since U is an orthogonal, we have $u_i^T(\gamma)u_i(\gamma) = 1$, that is, $\sum_{j=1}^{n}(u_i^j)^2(\gamma) = 1$, which on differentiating with respect to γ gives

$$u_i^T(\gamma)\frac{\partial u_i(\gamma)}{\partial\gamma} = 0. \tag{1.24}$$

This together with (1.23) yields

$$u_i^T(\gamma)\frac{\partial A(\gamma)}{\partial\gamma}v_i(\gamma) + u_i^T(\gamma)A(\gamma)\frac{\partial v_i(\gamma)}{\partial\gamma} = \frac{\partial\sigma_i(\gamma)}{\partial\gamma}. \tag{1.25}$$

Pre-multiplying (1.22) by $A^T(\gamma)$ and writing $A^T(\gamma)A(\gamma)v_i(\gamma) = \sigma_i(\gamma)^2 v_i(\gamma)$ we have

$$\begin{aligned} \sigma_i^2 v_i(\gamma) &= \sigma_i(\gamma)A^T(\gamma)u_i(\gamma), \\ \text{which is, } (\sigma_i^2 v_i(\gamma))^T &= (\sigma_i(\gamma)A^T(\gamma)u_i(\gamma))^T, \\ \text{and finally, } u_i^T(\gamma)A(\gamma) &= \sigma_i(\gamma)v_i^T(\gamma). \end{aligned} \tag{1.26}$$

Using the orthogonality of the columns of V, that is, $v_i^T(\gamma)v_i(\gamma) = 1$ and differentiating it with respect to γ we have, for each i,

$$v_i^T(\gamma)\frac{\partial v_i(\gamma)}{\partial\gamma} = 0, \tag{1.27}$$

which together with (1.26) gives

$$u_i^T(\gamma)A(\gamma)\frac{\partial v_i(\gamma)}{\partial \gamma} = \sigma_i(\gamma)v_i^T(\gamma)\frac{\partial v_i(\gamma)}{\partial \gamma} = 0. \tag{1.28}$$

Using (1.25) we find

$$u_i^T(\gamma)\frac{\partial A(\gamma)}{\partial \gamma}v_i(\gamma) = \frac{\partial \sigma_i(\gamma)}{\partial \gamma}. \tag{1.29}$$

Note that the orthogonal left and right singular vectors play an important role in finding the relation (1.29), which is a generic expression for $A(\gamma)$, any matrix which depends smoothly on the parameter γ. For given A and R denote $A(\gamma) := A + \gamma R$. Define the singular values of $A(\gamma)$ as $\sigma_i(\gamma)$ for $i = 1, 2...n$. We can write a Taylor series expansion for $\sigma_i(\gamma)$ at $\gamma = 0$ as

$$\sigma_i(\gamma) = \sigma_i(0) + (\gamma - 0)\frac{\partial \sigma_i(\gamma)}{\partial \gamma} \mid_{\gamma=0} + o(\gamma). \tag{1.30}$$

Substituting $A(\gamma) = A + \gamma R$ in (1.29) we find

$$u_i^T(\gamma)\frac{\partial (A + \gamma R)}{\partial \gamma}v_i(\gamma) = \frac{\partial \sigma_i(\gamma)}{\partial \gamma}. \tag{1.31}$$

Since $\frac{\partial A}{\partial \gamma} = 0$ and $\frac{\partial R}{\partial \gamma} = 0$, (1.31) gives

$$u_i^T(\gamma)Rv_i(\gamma) = \frac{\partial \sigma_i(\gamma)}{\partial \gamma}. \tag{1.32}$$

Note that, at $\gamma = 0$, $\sigma_i(0) = \sigma_i$, and $u_i^T(0) = u_i^T$, are the singular values and singular vectors of the matrix A, respectively. So finally we have,

$$u_i^T R v_i = \frac{\partial \sigma_i}{\partial \gamma} \mid_{\gamma=0} . \tag{1.33}$$

Using (1.33) in (1.30) gives

$$\begin{aligned}\sigma_i(\gamma) &= \sigma_i(0) + \gamma\frac{\partial \sigma_i(\gamma)}{\partial \gamma} \mid_{\gamma=0} + o(\gamma) \\ &= \sigma_i + \gamma u_i^T R v_i + o(\gamma).\end{aligned} \tag{1.34}$$

Denote $\|A\| := \phi(\vec{\sigma})$, where $\vec{\sigma} = (\sigma_1\ \sigma_2\ ...\ \sigma_n)^T$ and ϕ is a symmetric gauge function [46]. If $d(\gamma) \in \partial\phi(\sigma(\gamma))$, then by the definition of subdiffrential, for all $\hat{\sigma}(\gamma) \in \mathbb{R}^n$ we have

$$\phi(\hat{\sigma}(\gamma)) - \phi(\sigma(\gamma)) \geq (\hat{\sigma}(\gamma) - \sigma(\gamma))^T d(\gamma). \tag{1.35}$$

Applying triangle inequality on $\phi(\hat{\sigma}(\gamma) - \sigma(\gamma))$ we find

$$\phi(\hat{\sigma}(\gamma) - \sigma(\gamma)) \;\; \geq \;\; \phi(\hat{\sigma}(\gamma)) - \phi(\sigma(\gamma)),$$

which together with (1.35) gives

$$\phi(\hat{\sigma}(\gamma) - \sigma(\gamma)) \geq \phi(\hat{\sigma}(\gamma) - \sigma(\gamma)) \geq (\hat{\sigma}(\gamma) - \sigma(\gamma))^T d(\gamma). \tag{1.36}$$

Since the choice of $\hat{\sigma}(\gamma) \in \mathbb{R}^n$ is arbitrary choose $\hat{\sigma}(\gamma) - \sigma(\gamma) = \sigma(0)$ and we have

$$\|A\| \;\; = \;\; \phi(\vec{\sigma}) = \phi(\hat{\sigma}(0)) \geq \hat{\sigma}^T d(\gamma),$$

and using (1.34) we find

$$\|A\| \;\; \geq \;\; \hat{\sigma}^T d(\gamma) \geq (\sigma(\gamma) - \gamma u^T R v - o(\gamma))^T d(\gamma).$$

That is,

$$\|A\| \;\; \geq \;\; \sum_{i=1}^{n} \sigma_i(\gamma) d_i(\gamma) - \gamma \sum_{i=1}^{n} d_i(\gamma) u_i^T R v_i - \sum_{i=1}^{n} o(\gamma) d_i(\gamma). \tag{1.37}$$

Using the fact that $d(\gamma) \in \phi(\sigma(\gamma))$, if and only if, $\phi(\sigma(\gamma)) = \sigma(\gamma)^T d(\gamma)$ and $\phi^*(\sigma(\gamma)) \leq 1$ we have,

$$\|A + \gamma R\| \;\; = \;\; \phi(\sigma(\gamma)) = \sigma(\gamma)^T d(\gamma) = \sum_{i=1}^{n} \sigma_i(\gamma) d_i(\gamma). \tag{1.38}$$

Using (1.37) and (1.38) together yields,

$$\|A\| \;\; \geq \;\; \|A + \gamma R\| - \gamma \sum_{i=1}^{n} d_i(\gamma) u_i^T R v_i - \sum_{i=1}^{n} o(\gamma) d_i(\gamma) \tag{1.39}$$

On the other hand, if $d(0) \in \partial\phi(\sigma(0))$, then by the definition of subdiffrential, for all $\sigma \in \mathbb{R}^n$ we have

$$\phi(\sigma) - \phi(\sigma(0)) \;\; \geq \;\; (\sigma - \sigma(0))^T d(0). \tag{1.40}$$

Applying triangle inequality on $\phi(\sigma - \phi(\sigma(0))$ and using (1.40) we find

$$\phi(\sigma - \sigma(0)) \;\; \geq \;\; \phi(\sigma) - \phi(\sigma(0)) \geq (\sigma - \sigma(0))^T d(0).$$

Since the choice of $\sigma \in \mathbb{R}^n$ is arbitrary considering $\sigma - \sigma(0) = \sigma(\gamma)$ we obtain

$$\|A + \gamma R\| \;=\; \phi(\sigma(\gamma)) \geq \sigma(\gamma)^T d(0) = (\sigma(0) + \gamma u^T R v + o(\gamma))^T d(0).$$

The last equality is due to (1.34). Therefore we have

$$\begin{aligned} \|A + \gamma R\| &\geq \sum_{i=1}^{n} \sigma_i(0) d_i(0) + \gamma \sum_{i=1}^{n} d_i(0) u_i^T R v_i + \sum_{i=1}^{n} o(\gamma) d_i(0) \\ &= \|A\| + \gamma \sum_{i=1}^{n} d_i(0) u_i^T R v_i + \sum_{i=1}^{n} o(\gamma) d_i(0). \end{aligned} \tag{1.41}$$

Combining (1.39) and (1.41) together we obtain

$$\sum_{i=1}^{n} d_i(0) u_i^T R v_i \leq \frac{\|A + \gamma R\| - \|A\|}{\gamma} \leq \sum_{i=1}^{n} d_i(\gamma) u_i^T R v_i. \tag{1.42}$$

Considering $\gamma \to 0+$ we achieve the result as desired. □

The next theorem gives a general representation of the subdifferential of a matrix norm. In this theorem subdifferential of a matrix norm is represented as the convex combination of the elements of a set, obtained from the SVD of a matrix.

Theorem 2. *[46] Let $A = U\Sigma V^T$ be a SVD of A and $d \in \partial\phi(\sigma)$. Then for a unitary invariance matrix norm $\|\cdot\|$, we have*

$$\partial\|A\| = \text{conv}\{UDV^T : D \in R^{m\times n}; D = \text{diag}(d) \text{ and } d \in \partial\phi(\sigma)\}.$$

Proof. Denote $\text{conv}\{UDV^T : D \in R^{m\times n}; D = \text{diag}(d) \text{ and } d \in \partial\phi(\sigma)\}$ as $S(A)$. Let $G \in S(A)$ and write $G = \sum_{i=1}^{n} \lambda_i e_i$, where $e_i \in S(A)$ and $\lambda_i \geq 0$ such that $\sum_{i=1}^{n} \lambda_i = 1$. For each i, let $A = U_i \Sigma V_i^T$ be a SVD of A. If $d_i \in \partial\phi(\sigma)$ then we can write $e_i = U_i D_i V_i^T$ such that $G = \sum_{i=1}^{n} \lambda_i U_i D_i V_i^T$ where $D_i = \text{diag}(d_i)$ for each $d_i \in \partial\phi(\sigma)$. Our goal is to show if $G \in S(A)$ then (i) $\text{tr}(G^T A) = \|A\|$, and (ii) $\|G\|^* \leq 1$. To prove the first condition we use the linearity and some basic properties of trace [56] and find

$$\text{tr}(G^T A) = \text{tr}(A^T G) = \text{tr}(A^T \sum_{i=1}^{n} \lambda_i U_i D_i V_i^T) = \text{tr}(\sum_{i=1}^{n} \lambda_i A^T U_i D_i V_i^T) = \text{tr}(\sum_{i=1}^{n} \lambda_i V_i \Sigma^T U_i^T U_i D_i V_i^T),$$

which can be further reduced to

$$\mathrm{tr}(G^T A) = \mathrm{tr}(\sum_{i=1}^{n} \lambda_i \Sigma^T D_i V_i^T V_i) = \sum_{i=1}^{n} \lambda_i \mathrm{tr}(\Sigma^T D_i) = \sum_{i=1}^{n} \lambda_i \sigma^T d_i = \sum_{i=1}^{n} \lambda_i \phi(\sigma).$$

Therefore,

$$\mathrm{tr}(G^T A) = \phi(\sigma) \sum_{i=1}^{n} \lambda_i = \phi(\sigma) = \|A\|.$$

To prove the second condition recall that, $\|G\|^* = \max_{\|R\|\leq 1} tr(G^T R)$. Therefore,

$$\begin{aligned} \|G\|_* &= \max_{\|R\|\leq 1} \mathrm{tr}(G^T R) = \max_{\|R\|\leq 1} \mathrm{tr}(R^T G) = \max_{\|R\|\leq 1} \mathrm{tr}(R^T \sum_{i=1}^{n} \lambda_i U_i D_i V_i^T) \\ &= \max_{\|R\|\leq 1} \sum_{i=1}^{n} \lambda_i \mathrm{tr}(V_i^T R^T U_i D_i). \end{aligned} \tag{1.43}$$

Using the definition of unitary invariant norm we have $\|U_i R V_i\| = \|R\|$, for all orthogonal matrices U_i and V_i. For $R^T \in \mathbb{R}^{n\times m}$ we find $\|V_i^T R^T U_i\| = \|R^T\| = \|R\| \leq 1$. Denote $\hat{R}_i := U_i^T R V_i$. From (1.43) we have

$$\begin{aligned} \|G\|_* &= \max_{\|R\|\leq 1} \sum_{i=1}^{n} \lambda_i \mathrm{tr}(V_i^T R^T U_i D_i) \leq \max_{\|\hat{R}_i\|\leq 1} \sum_{i=1}^{n} \lambda_i \mathrm{tr}(\hat{R}_i^T D_i) = \sum_{i=1}^{n} \lambda_i \max_{\|\hat{R}_i\|\leq 1} \mathrm{tr}(\hat{R}_i^T D_i) \\ &= \sum_{i=1}^{n} \lambda_i \|D_i\|_*. \end{aligned} \tag{1.44}$$

In order to prove $\|G\|^* \leq 1$, first we show $\|D_i\|_* = \phi^*(d_i)$. By the characterization of subdiffrential we have

$$\partial(\phi(\sigma)) = \{d_i : \sigma^T d_i = \phi(\sigma), \phi^*(d_i) = \max_{\phi(y)\leq 1} d_i^T y \leq 1\}.$$

Using the definition of the dual norm of D_i we write

$$\|D_i\|_* = \max_{\|X\|\leq 1} tr(X^T D_i),$$

where $X \in \mathbb{R}^{m\times n}$. Recall that any unitary invariant matrix norm can be characterized by the symmetric gauge function of its singular values [46]. Therefore we have $\|A\| = \phi(\sigma(A))$,

where $\sigma(A)$ is a vector containing the singular values of A and $\phi : \mathbb{R}^n \to \mathbb{R}$ is a symmetric gauge function. Hence,

$$\|D_i\|_* = \max_{\phi(\sigma(X))\leq 1} tr(X^T D_i). \tag{1.45}$$

Let $X = U_1\Sigma_1 V_1^T$ be a SVD of X and write $V_1^T X^T U_1 = \Sigma^T$. We can right multiply both sides of the above relation by a permutation matrix E of size $m \times n$ which have diagonal elements as either $+1$ or -1, and everywhere else is zero, and obtain $V_1^T X^T U_1 E = \Sigma^T E$. By the property of symmetric gauge function [46] we have

$$\phi\left(\epsilon_1 x_{i_1}, \epsilon_2 x_{i_2}, \ ... \ \epsilon_n x_{i_n}\right) = \phi(x),$$

where $\epsilon_i = \pm 1$ for all i and $i_1, i_2, ...i_n$ is a permutation of of the set $\{1, 2, ...n\}$. Therefore we have

$$\|X\| = \phi(\sigma(X)) = \phi(\sigma(V_1^T X^T U_1)) = \phi(\sigma(\Sigma^T)) = \phi(\sigma(V_1^T X^T U_1 E)) = \phi(\sigma(\Sigma^T E)),$$

and using (1.45) we find[3]

$$\begin{aligned}
\|D_i\|_* &= \max_{\phi(\sigma(X))\leq 1} tr(X^T D_i) \\
&= \max_{\phi(\sigma(V_1^T X^T U_1))\leq 1} tr(V_1^T X^T U_1 D_i) \\
&= \max_{\phi(\sigma(\Sigma^T))\leq 1} tr(\Sigma^T D_i); \quad [\text{since } \Sigma, D_i \in \mathbb{R}^{m\times n}] \\
&= \max_{\phi(\sigma(\Sigma^T E))\leq 1} \langle [E\sigma(X)], d_i\rangle \\
&= \max_{\phi(\sigma(\Sigma^T E))\leq 1} \langle [E\sigma(X)], d_i\rangle \\
&= \max_{\phi(z)\leq 1} \langle z, d_i\rangle \\
&= \phi^*(d_i).
\end{aligned}$$

Therefore, $\|D_i\|_* = \phi^*(d_i) \leq 1$. Using (1.44) we have

$$\|G\|_* \leq \sum_{i=1}^{n} \lambda_i \|D_i\|_* \leq \sum_{i=1}^{n} \lambda_i = 1.$$

[3]In this derivation we denote the coordinate of a vector v as $[v]$

Hence the second condition is proved. In summary we conclude, if $G \in S(A)$ then $G \in \partial\|A\|$. So $S(A) \subseteq \partial\|A\|$. On the contrary let us assume there exists a $G_0 \in \partial\|A\|$ but $G_0 \notin S(A)$. By separation theorem, for all $H \in S(A)$ there exists a $R \in \mathbb{R}^{m\times n}$ such that,

$$\mathrm{tr}(R^T H) \leq \mathrm{tr}(R^T G_0).$$

Let $H = UDV^T \in S(A)$, $D = \mathrm{diag}(d)$ and $d \in \partial(\phi(\sigma))$. Therefore,

$$\begin{aligned}
\mathrm{tr}(R^T H) &= \mathrm{tr}(R^T UDV^T) \\
&= \mathrm{tr}(UDV^T R^T); \quad [\mathrm{tr}(AB) = \mathrm{tr}(BA)] \\
&= \mathrm{tr}(D^T U^T RV); \quad [\mathrm{tr}(A^T B) = \mathrm{tr}(B^T A)] \\
&= \sum_{i=1}^{n} d_i u_i^T R v_i.
\end{aligned}$$

And finally,

$$\begin{aligned}
\max_{\substack{D=diag(d)\\ d\in\partial(\phi(\sigma))}} tr(R^T H) &< tr(R^T G_0) \leq \max_{G\in\partial\|A\|} tr(R^T G), \\
\text{which implies, } \max_{\substack{D=diag(d)\\ d\in\partial(\phi(\sigma))}} \sum_{i=1}^{n} d_i u_i^T R v_i &< \lim_{\gamma\to 0+} \frac{\|A+\gamma R\| - \|A\|}{\gamma}.
\end{aligned}$$

If $G \in \partial\|A\|$ then $\|A + \gamma R\| \geq \|A\| + tr(\gamma R^T G)$, for all $A + \gamma R \in \mathbb{R}^{m\times n}$. So, using Theorem 1, we have $\lim_{\gamma\to 0+} \frac{\|A+\gamma R\| - \|A\|}{\gamma} = \max_{d\in\partial\phi(\sigma)} \sum_{i=1}^{n} d_i u_i^T R v_i$ and arrive at a contradiction. Therefore, our assumption was wrong and $\partial\|A\| \subseteq S(A)$ and we obtain the desired result $S(A) = \partial\|A\|$. □

Example 3. *[46] Let $A = U\Sigma V^T$ be a singular value decomposition of A and denote $\phi(\sigma) := \|\sigma\|_\infty$ as the spectral norm of A. Then $\partial\|\sigma\|_\infty = \mathrm{conv}\{e_i, : \sigma_i = \sigma_1\}$ and if the algebraic multiplicity of σ_1 be t then we have $\partial\|A\| = \{U^{(1)} H V^{(1)} \ : H \in \mathbb{R}^{t\times t}, H \geq 0, \mathrm{tr}(H) = 1\}$.*

Proof. As mentioned above let $A = U\Sigma V^T$ be a SVD of A, and let the multiplicity of σ_1 be t, with $U = [U^{(1)} \ \ U^{(2)}]$ and $V = [V^{(1)} \ \ V^{(2)}]$, where $U^{(1)}$ and $V^{(1)}$ have t columns. Before writing the singular value decomposition of A we would like to define the $(t+1)^{th}$

singular values of A as σ_{t+1} and the preceding singular values as σ_1, since σ_1 has multiplicity t. Therefore,

$$A = U\Sigma V^T = [U^{(1)}\ U^{(2)}]\text{diag}(\sigma_1\ \sigma_1\ \cdots\sigma_1\ \sigma_{t+1}\ \cdots\sigma_n)[V^{(1)}\ V^{(2)}]^T,$$

$$\text{which implies, } A = U^{(1)}\begin{pmatrix} \sigma_1 & 0 & ... & 0 \\ 0 & \sigma_1 & ... & 0 \\ 0 & 0 & \sigma_1 & 0 \\ . & & & \\ 0 & 0 & ... & \sigma_1 \end{pmatrix} V^{(1)T} + U^{(2)}\Sigma^{(2)}V^{(2)T}$$

$$= U^{(1)}\sigma_1 I_t V^{(1)T} + U^{(2)}\Sigma^{(2)}V^{(2)T}$$

$$= \sigma_1 U^{(1)}V^{(1)T} + U^{(2)}\Sigma^{(2)}V^{(2)T}, \tag{1.46}$$

where $\Sigma^{(2)}$ is a diagonal matrix containing the remaining singular values of A. According to Theorem 2, $G \in \partial\|A\|$ can be written as $G = \sum_{i=1}^n \mu_i U_i^{(1)} D_i^{(1)} V_i^{(1)T}$, with $\mu_i \geq 0$, and $\sum_{i=1}^n \mu_i = 1$. Also note that for each i, $A = U_i\Sigma V_i^T$ be a SVD of A, and $d_i \in \partial\|\sigma\|_\infty$. Now we will prove the following statement: *If $\phi(\sigma) = \|\sigma\|_\infty$ which is the spectral norm of A then $\partial\|\sigma\|_\infty = \text{conv}\{e_i, : \sigma_i = \sigma_1\}$.* We use the following argument: We can write the subdiffrential of $\|\sigma\|_\infty$ as $\partial\|\sigma\|_\infty = \{y :< y, \sigma >= \|\sigma\|_\infty \text{ and } \|y\|_1 \leq 1\}$. Therefore,

$$\begin{aligned}
\|\sigma\|_\infty = \sigma_1 &= \sigma^T y \\
&= \sigma_1 y_1 + \sigma_1 y_2 + \cdots \sigma_1 y_t + \sigma_{t+1} y_{t+1} + \cdots \sigma_n y_n \\
&\leq \sigma_1(|y_1| + |y_2| + \cdots + |y_t|) + \sigma_{t+1}|y_{t+1}| + \cdots \sigma_n|y_n| \qquad [\text{Since, } \sigma_i \geq 0 \text{ for all } i] \\
&\leq \sigma_1(|y_1| + |y_2| + \cdots + |y_t| + |y_{t+1}| + \cdots + |y_n|) \\
&\leq \sigma_1\|y\|_1 \\
&\leq \sigma_1.
\end{aligned}$$

To achieve the equality we must have

$$\begin{aligned}\sigma_1|y_i| &= \sigma_i y_i; \quad i = 1, 2, ...t \\ &= \sigma_i 0; \quad i = t+1, ...n.\end{aligned}$$

Thus $y_1, y_2, ...y_t \geq 0$ and $y_i = 0,\ i = t+1, t+2, ...n$. Now from $\sigma_1\|y\|_1 = \sigma_1$ we have $\|y\|_1 = 1$ Therefore we find $\sum_{i=1}^{t} y_i = 1$ and $y = y_1e_1 + y_2e_2 + ...y_te_t \in \text{conv}\{e_i, : \sigma_i = \sigma_1\}$ and we proved the statement. Since, $G = \sum_{i=1}^{n} \mu_i U_i^{(1)} D_i^{(1)} V_i^{(1)T}$, we can express $U_i^{(1)}$ and $V_i^{(1)}$, in terms of $U^{(1)}$ and $V^{(1)}$ using the transformation $U_i^{(1)} = U^{(1)}X_i$ and $V_i^{(1)} = V^{(1)}Y_i$, where each X_i and Y_i is a $t \times t$ orthogonal matrix. Since $V_i^T = \frac{1}{\sigma_i} u_i^T A$ we have

$$V_i^{(1)}X_i = \frac{1}{\sigma_i} A^T u_i X_i = V^{(1)}X_i.$$

Hence $X_i = Y_i$. Therefore we can write G as

$$\begin{aligned}G &= \sum_{i=1}^{n} \mu_i U^{(1)} X_i D_i^{(1)} X_i^T V^{(1)T} \\ &= U^{(1)} \left(\sum_{i=1}^{n} \mu_i X_i D_i^{(1)} X_i^T \right) V^{(1)T}.\end{aligned}$$

Defining $H = \sum_{i=1}^{n} \mu_i X_i D_i^{(1)} X_i^T$ and using the linearity of trace we can show

$$\begin{aligned}tr(H) &= tr\left(\sum_{i=1}^{n} \mu_i X_i D_i^{(1)} X_i^T \right) \\ &= \left(\sum_{i=1}^{n} \mu_i tr(X_i D_i^{(1)} X_i^T) \right) \\ &= \left(\sum_{i=1}^{n} \mu_i tr(D_i^{(1)} X_i^T X_i) \right) \quad [\text{since } tr(AB) = tr(BA)] \\ &= \left(\sum_{i=1}^{n} \mu_i tr(D_i^{(1)}) \right).\end{aligned}$$

Recall from Theorem 2, we have $\partial\|A\|$=conv$\{UDV^T \in R^{m\times n}; D = diag(d)$ and $d \in \partial\phi(\sigma)\}$ and we have already proved for $y \in \partial\phi(\sigma)$, $\|y\|_1 = 1$ and $D_i^{(1)}$'s are constructed such that $D_i^{(1)} = \text{diag}(y), y \in \partial\phi(\sigma) = \partial\|\sigma\|_\infty$. Therefore $tr(D_i^{(1)}) = 1$, and $tr(H) =$

$\left(\sum_{i=1}^{n} \mu_i tr(D_i^{(1)})\right) = \sum_{i=1}^{n} \mu_i = 1$. To prove H is positive semidefinite we choose $x \in \mathbb{R}^t$ and find

$$\begin{aligned} x^T H x &= \sum_{i=1}^{n} \mu_i x^T X_i D_i^{(1)} X_i^T x \\ &= \sum_{i=1}^{n} \mu_i (X_i^T x)^T D_i^{(1)} (X_i^T x) \\ &= \sum_{i=1}^{n} \mu_i z^T D_i^{(1)} z. \quad (\text{Denote } z := X_i^T x) \end{aligned}$$

In summary we have, $D_i^{(1)}$ is positive semidefinite for all $y \in \mathbb{R}^t$ and $H = \sum_{i=1}^{n} \mu_i X_i D_i^{(1)} X_i^T$, is positive semidefinite as well. Therefore we can define the subdiffrential of A as

$$\partial \|A\| = \{U^{(1)} H V^{(1)} \text{ for all } H \in \mathbb{R}^{t \times t}, H \geq 0, \text{tr}(H) = 1\}.$$

Hence the result. □

Example 4. *[46] Let $A \in \mathbb{R}^{m \times n}$ (assume $m \geq n$) has a SVD $A = U \Sigma V^T$ with s zero singular values, such that $s < n$. Denote $\phi(\sigma) := \|\sigma\|_1$ then $\partial \|\sigma\|_1 = \{x \in \mathbb{R}^n : |x_i| \leq 1, x_i = 1, i = 1, 2, ...n - s\}$ and $\partial \|A\| = \{U^{(1)} V^{(1)T} + U^{(2)} T V^{(2)T}; \text{for all } T \in \mathbb{R}^{m-n+s \times s}, \sigma_1(T) \leq 1\}$, where $U = [U^{(1)} \ U^{(2)}]$ and $V = [V^{(1)} \ V^{(2)}]$, such that $U^{(1)}$ and $V^{(1)}$ have $(n - s)$ columns.*

Proof. Note that, $\partial \|\sigma\|_1 = \{y \in \mathbb{R}^n :< y, \sigma >= \|\sigma\|_1 \text{ and } \|y\|_\infty \leq 1\}$. Since there are s zero singular values, we have,

$$\begin{aligned} \|\sigma\|_1 &= \sigma_1 + \sigma_2 + ... + \sigma_n; \quad [\sigma_i \geq 0] \\ &= \sigma_1 + \sigma_2 + ... + \sigma_{n-s}. \end{aligned}$$

Furthermore,

$$\begin{aligned} \|\sigma\|_1 &= \sigma_1 + \sigma_2 + ... + \sigma_{n-s} \\ &= \sigma^T y \\ &= \sigma_1 y_1 + \sigma_2 y_2 + ... \sigma_{n-s} y_{n-s} \end{aligned} \tag{1.47}$$

$$\leq \quad \sigma_1|y_1| + \sigma_2|y_2| + ...\sigma_{n-s}|y_{n-s}| \tag{1.48}$$

$$\leq \quad \|y\|_\infty(\sigma_1 + \sigma_2 + ...\sigma_{n-s}); \quad (\text{Since; } \|y\|_\infty = \max_{1\leq i\leq n-s} |y_i|) \tag{1.49}$$

$$\leq \quad \|\sigma\|_1\|y\|_\infty \tag{1.50}$$

$$\leq \quad \|\sigma\|_1. \tag{1.51}$$

For (1.50) to become an equality $\|y\|_\infty(\sigma_1 + \sigma_2 + \sigma_3 + ...\sigma_{n-s}) = \|\sigma\|_1\|y\|_\infty$ we must have $\|y\|_\infty = 1$. For (1.49) to become an equality we need

$$\begin{aligned} \sigma_1|y_1| + \sigma_2|y_2| + \sigma_3|y_3| + ...\sigma_{n-s}|y_{n-s}| &= \|y\|_\infty(\sigma_1 + \sigma_2 + \sigma_3 + ...\sigma_{n-s}) \\ \text{which implies, } |y_i| &= \|y\|_\infty = 1, \quad (\text{for } i = 1, 2, ...n-s). \end{aligned} \tag{1.52}$$

For (1.48) to reduce to an equality, we need $\sigma_1 y_1 + \sigma_2 y_2 + \sigma_3 y_3 + ...\sigma_{n-s}y_{n-s} = \sigma_1|y_1| + \sigma_2|y_2| + \sigma_3|y_3| + ...\sigma_{n-s}|y_{n-s}|$, which together with (1.52) implies $y_i = |y_i| = 1; i = 1, 2, ...n-s$. Combining all these conditions together finally we have

$$\partial\|\sigma\|_1 = \{x \in \mathbb{R}^n : |x_i| \leq 1, x_i = 1, i = 1, 2, ...n - s\}.$$

From Theorem 2, an element G of the set $\partial\|A\|$ can be written as $G = \sum_{i=1}^n \mu_i U_i D_i V_i^T$ with $\mu_i \geq 0$ and $\sum_{i=1}^n \mu_i = 1$, where $d_i \in \partial\|\sigma\|_1$ and for each i, let $A = U_i\Sigma V_i^T$ be a SVD of A. Employing the partition $U = [U^{(1)} \;\; U^{(2)}]$ and $V = [V^{(1)} \;\; V^{(2)}]$, where $U^{(1)}$ and $V^{(1)}$ have $n - s$ columns, one can write $G = U^{(1)}V^{(1)T} + \sum_i \mu_i U_i^{(2)} W_i V_i^{(2)T}$, where W_i is an $(m-n+s) \times s$ diagonal matrix with the absolute value of each diagonal element less than 1. We can write $U_i^{(2)}$ and $V_i^{(2)}$, in terms of $U^{(2)}$ and $V^{(2)}$ using the transformation $U_i^{(2)} = U^{(2)}Y_i$ and $V_i^{(2)} = V^{(2)}Z_i$, where Y_i and Z_i are orthogonal matrices of size $(m-n+s) \times (m-n+s)$ and $s \times s$, respectively. Therefore, G can be written as

$$G = U^{(1)}V^{(1)T} + U^{(2)}TV^{(2)T},$$

where $T = \sum_i \mu_i Y_i W_i Z_i^T \in \mathbb{R}^{(m-n+s)\times s}$. Since Y_i and Z_i are orthogonal matrices of size $(m - n + s) \times (m - n + s)$ and $s \times s$, respectively, and W_i is an $(m - n + s) \times s$ diagonal

matrix $Y_i W_i Z_i^T$ is a singular value decomposition of W_i for each i. If $\sigma_1(T)$ denotes the largest singular value of the matrix T then

$$\sigma_1(T) = \sigma\left(\sum_i \mu_i Y_i W_i Z_i^T\right). \tag{1.53}$$

Since W_i is an $(m-n+s)\times s$ diagonal matrix with the absolute value of each diagonal element less than 1 we have $\sigma_1(W_i) \leq 1$. Hence (1.58) yields,

$$\begin{aligned}\sigma_1(T) &= \sigma\left(\sum_i \mu_i Y_i W_i Z_i^T\right)\\ &= \left(\sum_i \mu_i \sigma_1(W_i)\right)\\ &\leq \sum_i \mu_i = 1.\end{aligned}$$

Therefore, given any singular value decomposition of a matrix A the subdiffrential of the matrix norm can be written as

$$\partial\|A\| = \{U^{(1)}V^{(1)T} + U^{(2)}TV^{(2)T}; \text{for all } T \in \mathbb{R}^{m-n+s\times s}, \sigma_1(T) \leq 1\}.$$

Hence the result. □

Theorems on Operator Norms We present the next two theorems, which are an extension of Theorem 1 and 2 in case of operator norm. Since the proofs of these theorems follow closely to the proof of Theorem 1 and 2, we will just quote the theorems. The reader can find the proofs in [46].

Theorem 5. *[46] Let $A, R \in \mathbb{R}^{m\times n}$ be given matrices. Then*

$$\lim_{\gamma\to 0+} \frac{\|A+\gamma R\| - \|A\|}{\gamma} = \max_{(v,w)\in\Phi(A)} w^T R v,$$

where $\Phi(A) = \{v \in \mathbb{R}^n, w \in \mathbb{R}^m : \|v\|_{\mathbb{R}^n} = 1, \frac{Av}{\|A\|} = u, \|u\|_{\mathbb{R}^m} = 1, w \in \partial\|u\|_{\mathbb{R}^m}\}$.

Theorem 6. *[46] With the notations defined in the previous theorem,*

$$\partial\|A\| = conv\{wv^T : (v,w) \in \Phi(A)\}.$$

1.2 Constrained and Unconstrained Principal Component Analysis (PCA)

In this section, we will review constrained and unconstrained classical principal component analysis problems and their solutions. Recall the classical principal component analysis (PCA) problem ([35, 38]) can be defined as an approximation to a given matrix $A \in R^{m\times n}$ by a rank r matrix under the Frobenius norm:

$$\min_{\substack{X \\ \mathrm{r}(X)\leq k}} \|A - X\|_F, \tag{1.54}$$

where $\mathrm{r}(X)$ denotes the rank of the matrix X. If $U\Sigma V^T$ is a singular value decomposition (SVD) of X then the solutions to the above problem are given by thresholding on the singular values of A: $\hat{X} = U\mathbf{H}_r(\Sigma)V^T$, where $\mathbf{H}_r$ is the hard-thresholding operator that keeps the r largest singular values and replaces the others by 0. This is also referred to as Eckart-Young-Mirsky's theorem in the literature [35]. An unconstrained version of problem (1.54) is:

$$\min_X \{\|A - X\|_F + \tau \mathrm{r}(X)\},$$

where τ is some fixed positive parameter. A careful reader should note that the above problem is simply the "Lagrangian form" of the problem (1.54). This problem can be solved by assuming the rank of X from 0 to $\min\{m, n\}$, and for each rank, it admits a closed form analytical solution, given by the SVD of A, with the singular values being hard-thresholded with τ. This algorithm is solvable in polynomial time. But in a more general set up where only a subset of the entries of the data matrix is observable, for example, matrix completion problem under low-rank penalties [48, 64]:

$$\min_X \mathrm{rank}(X) \quad \text{subjcet to } A_{ij} = X_{ij},\ (i,j) \in \Omega,$$

where $\Omega \subseteq \{(i,j) : 1 \leq i \leq m, 1 \leq j \leq n\}$, is indeed NP-hard [34][4]. One common idea used in such a situation is to consider a convex relaxation of the above problem. As it turns out, the nuclear norm $\|X\|_*$, the sum of the singular values of X, is a good substitution for r(X) [33, 66] (see Section 1.1.2 for a detailed discussion on the nuclear norm and its properties). Cai et al. used this idea and formulated the following convex approximation problem ([48]):

$$\min_{X \in \mathbb{R}^{m \times n}} \left\{ \frac{1}{2}\|A - X\|_F^2 + \tau \|X\|_* \right\}, \tag{1.55}$$

which they refer as singular value thresholding (SVT). Problem (1.55) can be solved using an explicit formula ([48, 64]),which is derived by using advanced tools from convex analysis ("subdifferentials" to be more specific).

1.2.1 Singular Value Thresholding Theorem

In this section we will quote the celebrated theorem of Cai, Candes and Shen [48]. We will start with the following lemma.

Lemma 7. *[46] Let $\phi(\sigma) = \|\sigma\|_1$ and let $X = U\Sigma V^T$ be a singular value decomposition of $X \in \mathbb{R}^{m \times n}$ (we assume $m \geq n$). Let r denote the number of nonzero singular values of X, and $r < n$. Then we have*

$$\partial \|X\|_* = \{U^{(1)}V^{(1)T} + W; W \in \mathbb{R}^{m \times n}, U^{(1)T}W = 0, WV^{(1)} = 0, \sigma_1(W) = \|W\|_2 \leq 1\}$$

where $U^{(1)} \in \mathbb{R}^{m \times r}$ and $V^{(1)} \in \mathbb{R}^{n \times r}$ are column orthogonal matrices.

[4] A careful reader should note that the matrix completion problem is a special case of the affinely constrained matrix rank minimization problem [64]:

$$\min_X \operatorname{rank}(X) \quad \text{subjcet to } A(X) = b,$$

where $X \in \mathbb{R}^{m \times n}$ be the decision variable and $A : \mathbb{R}^{m \times n} \to \mathbb{R}^p$ be a linear map.

Proof. Recall from Example 2,

$$\partial\|\sigma\|_1 = \{x \in \mathbb{R}^n : |x_i| \leq 1, x_i = 1, i = 1, 2, ...r\}.$$

Note that, by Theorem 2, if $G \in \partial\|X\|_*$ then $G = \sum_{i=1}^{p} \mu_i U_i D_i V_i^T$ with $\mu_i \geq 0$ and $\sum_{i=1}^{p} \mu_i = 1$ and for each i, $X = U_i \Sigma V_i^T$ denotes a SVD of X and $d_i \in \partial\|\sigma\|_1$. Let $X = U\Sigma V^T$ be a singular value decomposition of X with r nonzero singular values. We partition the matrices U and V such that $U = [U^{(1)} \quad U^{(2)}]$ and $V = [V^{(1)} \quad V^{(2)}]$, where $U^{(1)}$ and $V^{(1)}$ have r columns. Write G as

$$G = U^{(1)}V^{(1)T} + \sum_i \mu_i U_i^{(2)} A_i V_i^{(2)T},$$

where A_i is an $(m-r) \times (n-r)$ diagonal matrix with each diagonal element having absolute value less than 1. We can express $U_i^{(2)} \in \mathbb{R}^{m \times m-r}$ and $V_i^{(2)} \in \mathbb{R}^{n \times n-r}$, in terms of $U^{(2)} \in \mathbb{R}^{m \times m-r}$ and $V^{(2)} \in \mathbb{R}^{n \times n-r}$ using the transformation $U_i^{(2)} = U^{(2)}Y_i$ and $V_i^{(2)} = V^{(2)}Z_i$ where the matrices Y_i and Z_i are orthogonal matrices of size $(m-r) \times (m-r)$ and $(n-r) \times (n-r)$ respectively. Therefore G can be written as

$$G = U^{(1)}V^{(1)T} + U^{(2)}TV^{(2)T},$$

where

$$T = \sum_i \mu_i Y_i A_i Z_i^T \in \mathbb{R}^{(m-r)\times(n-r)}.$$

We can further modify G as

$$G = U^{(1)}V^{(1)T} + W,$$

where

$$W = U^{(2)}TV^{(2)T} \ ; W \in \mathbb{R}^{m \times n},$$

such that

$$U^{(1)T}W = 0, WV^{(1)} = 0.$$

If $\sigma_1(W)$ denotes the largest singular value of the matrix W then using unitary invariant property of the matrix norm

$$\begin{aligned}\sigma_1(W) &= \sigma\left(\sum_i \mu_i U^{(2)} Y_i A_i Z_i^T V^{(2)T}\right) \\ &= \sigma\left(U^{(2)}[\sum_i \mu_i Y_i A_i Z_i^T] V^{(2)T}\right) \\ &= \sigma\left(\sum_i \mu_i Y_i A_i Z_i^T\right). \end{aligned} \tag{1.56}$$

Since A_i is an $(m-r)\times(n-r)$ diagonal matrix with each diagonal element having absolute value less than 1, we have $\sigma_1(A_i) \leq 1$. Further using the unitary invariant property of the matrix norm in (1.56) we find,

$$\sigma_1(W) = \sigma\left(\sum_i \mu_i Y_i A_i Z_i^T\right) = \left(\sum_i \mu_i \sigma_1(A_i)\right) \leq \sum_i \mu_i = 1.$$

Hence the result. □

Theorem 8. *[48] Let $A \in \mathbb{R}^{m\times n}$ be given. For each $\tau \geq 0$, the singular value shrinkage operator obeys*

$$D_\tau(A) = \arg\min_X \{\frac{1}{2}\|X - A\|_F^2 + \tau\|X\|_*\}. \tag{1.57}$$

Proof. Denote $h(X) := \frac{1}{2}\|A - X\|_F^2 + \tau\|X\|_*$. Since both Frobenius norm and nuclear norm are convex functions in X on $\mathbb{R}$, $h(X)$ is a strictly convex function in X. Therefore, $h(X)$ has a unique minimizer and the motivation behind this theorem is to show the minimizer is $D_\tau(A)$. Note that, $\hat{X}$ minimizes $h(X)$ if and only if $0 \in \partial h(\hat{X})$, that is,

$$0 \in \hat{X} - A + \tau\partial\|\hat{X}\|_*.$$

Let $X \in \mathbb{R}^{m\times n}$, be a matrix of rank r and $X = U\Sigma V^T$ be a SVD of X, with $U \in \mathbb{R}^{m\times r}$ and $V \in \mathbb{R}^{n\times r}$ being column orthonormal matrices. According to Lemma 7,

$$\partial\|X\|_* = \{UV^T + W; W \in \mathbb{R}^{m\times n}, U^T W = 0, WV = 0, \|W\|_2 \leq 1\}.$$

Write the SVD of A as

$$A = U_0 \Sigma_0 V_0^T + U_1 \Sigma_1 V_1^T, \tag{1.58}$$

where U_0, V_0 are the singular vectors corresponding to the singular values of A greater than τ, and U_1, V_1 are the singular vectors corresponding to the singular values of A less than τ, respectively. Denote $\hat{X} := D_\tau(A)$ then using (1.58) we have

$$\hat{X} = U_0(\Sigma_0 - \tau I)V_0^T,$$

and therefore,

$$A - \hat{X} \;=\; U_1 \Sigma_1 V_1^T + \tau U_0 V_0^T = \tau(U_0 V_0^T + \tau^{-1} U_1 \Sigma_1 V_1^T) = \tau(U_0 V_0^T + W),$$

where $W = \tau^{-1} U_1 \Sigma_1 V_1^T$ be such that $U_0^T W = 0$ and $W V_0 = 0$, since $U_0^T U_1 = 0$ and $V_1^T V_0 = 0$. Note that, from (1.58) we have the diagonal elements of Σ_1 are less than τ. If we denote $\sigma_1(W)$ as the largest singular value of W, then using the unitary invariance property of norm we find

$$\sigma_1(W) \;=\; \sigma_1\left(\tau^{-1} U_1 \Sigma_1 V_1^T\right) = \tau^{-1} \sigma_1\left(\Sigma_1\right) \leq 1.$$

Therefore, $\|W\|_2 \leq 1$ and finally we conclude $A - \hat{X} \in \tau \partial \|\hat{X}\|_*$, which implies $D_\tau(A) = \arg\min_X \{\frac{1}{2}\|X - A\|_F^2 + \tau \|X\|_*\}$. Hence the result. $\square$

1.3 Principal Component Pursuit Problems or Robust PCA

It is well-known that the solution to the classical PCA problem is numerically sensitive to the presence of outliers in the matrix. In other words, if the matrix A is perturbed by one single large value at one location, the explicit formula for its low-rank approximation would yield a much different solution than the unperturbed one. This phenomenon may be attributed to the use of the Frobenius norm in measuring the closeness to A by its approximation in the equivalent formulation of the classical PCA problem: the Frobenius norm would not

encourage zero entries while making the norm small. As long as the matrix $X - A$ is sufficiently sparse, one can recover the low-rank matrix X. This leads to the formulation of following rank-minimization problem:

$$\min_{X \in \mathbb{R}^{m \times n}} \{\mathrm{r}(X) + \lambda \|X - A\|_0\}, \tag{1.59}$$

where $\lambda > 0$ is a balancing parameter and $\|\cdot\|_0$ is the ℓ_0 norm which represents the number of non zero entries in amtrix. Solving (1.59) directly is infeasible. It is combinatorial and NP-hard [34]. On the other hand, we have learned recently (in particular during the last decade) that ℓ_1 norm does encourage vanishing entries when the norm is made small. Therefore, a good candidate to replace the ℓ_0 norm could be the ℓ_1 norm. Thus, to solve the problem of separating the sparse outliers added to a low-rank matrix, Candes, Li, Ma, and Wright argued to further replace the Frobenius norm in the SVT problem by the ℓ_1 norm ([32]; see also [9]) and introduced the *Robust PCA* (RPCA) formulation:

$$\min_X \{\frac{1}{2}\|X - A\|_{l_1} + \lambda \|X\|_*\}. \tag{1.60}$$

Unlike in the classical PCA and SVT problems, there is no explicit formula for the solution of the above problem. Various numerical procedures have been proposed to solve RPCA problem. In [9], using augmented Lagrange multiplier method, Lin, Chen, and Ma proposed two iterative methods: the exact Augmented Lagrange Method (EALM) and the inexact Augmented Lagrange Method (iEALM). The iEALM method turns out to be equivalent to the alternating direction method (ADM) later proposed by Tao and Yuan in [44]. In [49], Wright et. al. proposed an proximal gradient algorithm to solve the RPCA problems as well.

In many real world applications, it is possible that some entries of the matrix A is missing or only a portion of its entries is observable. In these situations one can think of an index set which represents the observable entries of the matrix A. Let Ω be such that $\Omega \subset \{1, 2, ..., m\} \times \{1, 2, ..., n\}$. One can also define a projection operator (which is self adjoint) $\pi_\Omega : \mathbb{R}^{m \times n} \to \mathbb{R}^{m \times n}$, such that, $(\pi_\Omega(A))_{ij} = A_{ij}$ if $(i, j) \in \Omega$, and $(\pi_\Omega(A))_{ij} = 0$ otherwise.

Therefore (1.60) can be written as

$$\min_{X,S\in\mathbb{R}^{m\times n}}\{\|X\|_* + \lambda\|S\|_{\ell_1}\},$$
$$\text{subject to } \pi_\Omega(X+S+E) = \pi_\Omega(A), \text{and for given } \delta > 0; \|\pi_\Omega(A-X-S)\|_F \leq \delta. \tag{1.61}$$

It is evident that the low-rank part of the matrix A is pretty rigid. In other words for problem (1.61), the sparse part of the decomposition can be restricted by using a projection operator and a feasible solution can still be achieved. But the projection operator can not be used on the low-rank part as it might bring huge discrepancies in X. It has already been shown that under certain randomness hypotheses the solution to the problem (1.61) can be achieved with high probability when $\delta = 0$. Aybat, Goldfarb, and Ma formulated an alternative to the minimization problem (1.61). Since, $\pi_\Omega(A-X-S) \subset X+S-\pi_\Omega(A)$, they formulated the following problem:

$$\min_{X,S\in\mathbb{R}^{m\times n}}\{\|X\|_* + \lambda\|\pi_\Omega(S)\|_{\ell_1} : X, S \in \mathbb{R}^{m\times n} \in \mathcal{X}\}, \tag{1.62}$$

where $\mathcal{X} := \{X, S \in \mathbb{R}^{m\times n} : \text{for given } \delta > 0; \|X+S-\pi_\Omega(A)\|_F \leq \delta\}$.

Theorem 9. *[74] If (X^*, S^*) is an optimal solution to (1.62) then $(X^*, \pi_\Omega(S^*))$ is an optimal solution to(1.61).*

Using the smoothing technique discussed in Section 1.6, Ayabat, Goldfarb and Ma proposed the following RPCA problem with smooth objective function:

$$\min_{X,S\in\mathbb{R}^{m\times n}}\{f_\mu(X) + \lambda g_\nu(S) : X, S \in \mathbb{R}^{m\times n} \in \mathcal{X}\}, \tag{1.63}$$

and partially smooth objective function

$$\min_{X,S\in\mathbb{R}^{m\times n}}\{f_\mu(X) + \lambda\|\pi_\Omega(S)\|_{\ell_1} : X, S \in \mathbb{R}^{m\times n} \in \mathcal{X}\}, \tag{1.64}$$

and showed the inexact solution to the problems (1.63) and (1.64) are closely related to the solution to (1.62).

Theorem 10. *[74] If $(X(\mu)^*, S(\nu)^*)$ is an $\epsilon/2$ optimal solution to (1.63) then $(X(\mu)^*, S(\nu)^*)$ is an ϵ optimal solution to(1.62) with $\mu = \nu = \frac{\epsilon}{4\tau}$, where $\tau = 0.5 \min\{m, n\}$.*

1.4 Weighted Low-Rank Approximation

In this section, we will briefly discuss the solutions of two classic weighted low-rank approximation problems: (i) Problem (1.4) proposed by Srebro and Jakkolla, and (ii) problem (1.5) proposed by Manton, Mahony, and Hua. Recall that working with a weighted norm is fundamentally difficult, as the weighted low-rank approximation problems do not admit a closed form solution, in general. Therefore a numerical procedure must be devised to solve the problems.

It is easy to see that problem (1.4) is a special case of problem (1.5) with $Q = \text{diag}(\text{vec}(W))$, where vec : $\mathbb{R}^{m\times n} \to \mathbb{R}^{mn\times 1}$ [23]. When W be a matrix of 1s, the solution to (1.4) can be given using the classical PCA, otherwise problem (1.4) has no closed form solution in general [39]. Note that the minimization problem (1.5), also becomes a regular low rank approximation problem (1.1) when Q is an identity matrix, and its solution can be approximated using the classical PCA [35]. It is a very common practice in non-negetive matrix factorization and shape and motion from image streams (SfM) to replace the rank constraint by the product of two matrices of compatible sizes [29, 40, 43, 45, 67, 68, 70, 71, 72]. That is, if $X \in \mathbb{R}^{m\times n}$ be such that $\text{r}(X) \leq r$, then X can be factorized as $X = UV^T$, where $U \in \mathbb{R}^{m\times r}$ and $V \in \mathbb{R}^{n\times r}$. Srebro and Jakkolla followed the above convention in studying the solution to (1.4). In order to solve (1.4), first, the authors used a numerical procedure inspired by the alternating direction method of updating U and V alternatively. The partial derivatives of

$$F(U, V) = \|(A - UV^T) \odot W\|_F^2$$

with respect to U and V respectively are given by:

$$\frac{\partial F}{\partial U} = (W \odot (UV^T - A))V \tag{1.65}$$

$$\frac{\partial F}{\partial V} = (W \odot (VU^T - A^T))V. \tag{1.66}$$

The system of equations obtained by setting $\frac{\partial F}{\partial U} = 0$, for a fixed V, is a linear system in U which after solving for U row-wise yields:

$$U(i,:)^T = (V^T \underline{W_i} V)^{-1} V^T \underline{W_i} A(:,i)^T, \tag{1.67}$$

where $\underline{W_i} \in \mathbb{R}^{n \times n}$ is a diagonal matrix with the weight from ith row of W along the diagonal and the vector $A(i,:)$ is the ith row of the matrix A. In 1997, Lu, Pei, and Wang used a similar technique to update U and V in closed form by using an alternating projection algorithm [29] (see also [23] for the algorithm and software package). They proposed to update U and V via the following iterative procedure: At $k+1$th step do:

$$\text{vec}(V_{k+1}) = \left((I_n \otimes U_k)^T \text{diag}(\text{vec}(W))(I_n \otimes U_k)\right)^{-1} (I_n \otimes U_k)^T \text{diag}(\text{vec}(W))\text{vec}(A),$$

and

$$\text{vec}(U_{k+1}) = \left((V_{k+1} \otimes I_m)\text{diag}(\text{vec}(W))(V_{k+1} \otimes I_m)^T\right)^{-1} (V_{k+1} \otimes I_m)\text{diag}(\text{vec}(W))\text{vec}(A).$$

But the above update rule is computationally expensive as one iteration of the alternating projection algorithm requires $O(mnr^2)$ flops for $\text{diag}(\text{vec}(W))$. However, with recovered U, Srebro and Jaakkola used a gradient descent method to update V. It is computationally efficient, because, with recovered $U = U^*$, it will take only $O(mnr)$ operations to compute $\frac{\partial F}{\partial V}$ using the formula $\frac{\partial F}{\partial V} = (W \odot (VU^{*T} - A^T))V$ and at the $k+1$ th iteration V_{k+1} is given via:

$$V_{k+1} = V_k - \eta \left((W \odot (V_k U^{*T} - A^T))V_k\right), \tag{1.68}$$

where η is the step length. Next, Srebro and Jaakkola proposed an Expectation-Maximization inspired approach to solve (1.4), which is much simpler to implement, though it could settle

down to a local minimum instead of a global minimum. The method is based on viewing (1.4) as a maximum-likelihood problem with missing entries. If W_{ij} only takes values either 0 or 1, corresponding to unobserved and observed entires of A, respectively, then the key observation for EM method is to refer A to a probabilistic model parameterized by the low-rank matrix X and write:

$$A = X + E,$$

where E is white Gaussian noise. Each EM update is inspired by finding a new low-rank matrix X which maximizes the expected log-likelihood of A as its missing entires are recovered by the current low-rank estimate X. In summary, in the expectation step one recovers the missing values of A from recent estimate X, and in the maximization step X is estimated as a low-rank approximation of newly formed A. The authors extended this approach to a general weighted case by considering a system with several target matrices: $A_1, A_2, \cdots, A_N$, but with a unique low-rank parameter matrix X such that

$$A_r = X + E_r,$$

where E_r are independent white Gaussian noise matrices. For $W_{ij} \in \mathbb{N} \cup \{0\}$, they rescaled the weight matrix to $W_{EM} = \frac{1}{\max_{ij}(W_1)_{ij}} W$ such that $(W_{EM})_{ij} \in [0, 1]$. By scaling the weight matrix, it is easy to see that problem (1.4) is transformed to a missing value problem with 0/1 weights and the EM update for X in each iterate is given by:

$$X_{k+1} = \mathbf{H}_r \left(W_{EM} \odot A + (\mathbb{1}_{m \times n} - W_{EM}) \odot X_k \right),$$

where $\mathbf{H}_r$ is the hard thresholding operator and $\mathbb{1}_{m \times n}$ is a $m \times n$ matrix of all 1s. The initialization for the EM method could be tricky. For a given threshold of weight bound ϵ_{EM}, the authors proposed to initialize X to a zero matrix if $\min_{ij}(W_{EM})_{ij} \leq \epsilon_{W_{EM}}$, otherwise initialize X to A.

Now we will give a brief outline of the method proposed by Manton, Mahony, and Hua to solve (1.5). Instead of using a matrix factorization to replace the rank constraint, Manton

et al. proposed a more generalized approach on a Grassman manifold to solve (1.5) by converting (1.5) to a double-minimization problem:

$$\min_{\substack{N\in\mathbb{R}^{n\times(n-r)}\\ N^TN=I_{n-r}}} \left(\min_{\substack{R\in\mathbb{R}^{m\times n}\\ RN=0}} \|A-R\|_Q^2 \right). \tag{1.69}$$

It is clear from the above formulation that $\mathrm{r}(N) = n - r$, which together with the condition $RN = 0$ implies $\mathrm{r}(R) \leq r$ as every column of $N \in \mathcal{N}(R)$, where $\mathcal{N}$ is the null-space of R. Since $\mathrm{r}(N) = n - r$, $\mathrm{r}(\mathcal{N}(R)) \leq n - r$ and using the rank-nullity theorem it is easy to see that $\mathrm{r}(R) \leq r$. They proposed: If $\hat{R}$ be the solution to the inner minimization problem

$$\hat{R} = \arg \min_{\substack{R\in\mathbb{R}^{m\times n}\\ RN=0}} \|A-R\|_Q^2,$$

then $\hat{R}$ is given by

$$\mathrm{vec}(\hat{R}) = \mathrm{vec}(A) - Q^{-1}(N\otimes I_m)\left((N\otimes I_m)^TQ^{-1}(N\otimes I_m)\right)^{-1}(N\otimes I_m)^T\mathrm{vec}(A),$$

where $\otimes$ is Kronecker's product. Using the expression for $\hat{R}$ in the inner minimization problem the objective function for the outer minimization problem is given by

$$\|X-\hat{R}\|_Q^2 = \mathrm{vec}(A)^T(N\otimes I_m)\left((N\otimes I_m)^TQ^{-1}(N\otimes I_m)\right)^{-1}(N\otimes I_m)^T\mathrm{vec}(A) := f(N),$$

a function of N. Finiding a minimum $\hat{N}$ for the minimization problem

$$\min_{\substack{N\in\mathbb{R}^{n\times(n-r)}\\ N^TN=I_{n-r}}} f(N),$$

is an $n(n-r)$ dimensional optimization problem. Consequently, by exploiting the symmetry, the optimization problem can be reduced to $r(n-r)$ parameters, as $f(N)$ depends on the range space of N, not on its individual elements. In [36], Edelman, Arias, and Smith introduced a Riemannian structure to solve the outer optimization problem. However, in [37], Manton, Mahony, and Hua argued that instead of a "flat space approximation" of the geodesic based algorithm, one can solve $f(N)$ subject to $N^TN = I$ only under the

assumption that f at any point N only depends on the range space of N. As a result they shown the geodesic-based optimization algorithms (see, for example [36]) are not only the "natural" algorithms.

Let $N^{\perp} \in \mathbb{R}^{n\times r}$ be the orthogonal complement of N satisfying $N^T N^{\perp} = 0$. For an arbitrary $N \in \mathbb{R}^{n\times(n-r)}$ with $N^T N = I$, and a certain perturbation matrix $Z \in \mathbb{R}^{n\times(n-r)}$, if $\mathcal{R}(N+Z) = \mathcal{R}(N)$, then $f(N+Z) = f(N)$, where $\mathcal{R}$ denotes the range space. Manton, Mahony, and Hua argued that, it is not necessary to consider all $n(n-r)$ search directions while minimizing $f(N)$. For fixed N and $N^{\perp}$, a perturbation $Z \in \mathbb{R}^{n\times(n-r)}$ uniquely decomposes as

$$Z = NL + N^{\perp}K,$$

where $L \in \mathbb{R}^{(n-r)\times(n-r)}$ and $K \in \mathbb{R}^{r\times(n-r)}$. Since $\mathcal{R}(N + NL) \subset \mathcal{R}(N)$, it is sufficient to consider only search direction $Z = N^{\perp}K$. Since the total number of elements in K is $r(n-r)$, minimizing $f(N)$ is an $r(n-r)$ dimensional problem. In order to solve

$$\min_{\substack{N\in\mathbb{R}^{n\times(n-r)} \\ N^T N = I_{n-r}}} f(N),$$

Manton, Mahony, and Hua outlined the following numerical procedure: Choose $N \in \mathbb{R}^{n\times(n-r)}$ and $N^{\perp} \in \mathbb{R}^{n\times r}$ such that $N^T N = I$ and $[N \;\; N^{\perp}]^T[N \;\; N^{\perp}] = I$ and define

$$\phi(K) = N + N^{\perp}K,$$

where $K \in \mathbb{R}^{r\times(n-r)}$ to form the local cost function

$$f(\phi(K)) = f(N + N^{\perp}K).$$

Apply Newton's method or simple steepest descent method to $f(\phi(K))$ to calculate $\nabla f(\phi(K))$ at $K = 0$, and compute a descent step $\triangle K$. A QR decomposition can be used to compute an N such that $N^T N = I$ and $\mathcal{R}(N) = \mathcal{R}(\phi(\triangle K))$. Repeat the above steps until convergence.

CHAPTER TWO: AN ELEMENTARY WAY TO SOLVE SVT AND SOME RELATED PROBLEMS

In this chapter, we want to give a new and elementary treatment of Theorem 8 that is accessible to a vast group of researchers, as it only requires basic knowledge of calculus and linear algebra, to the singular value thresholding (SVT) and some other related sparse recovery problems. We also show how naturally the shrinkage function can be used in solving more advanced problems.

2.1 A Calculus Problem

We start with a regular calculus problem. Let $\lambda > 0$ and $a \in \mathbb{R}$ be given. Consider the following problem:

$$S_\lambda(a) := \arg\min_{x\in\mathbb{R}} \left\{ \lambda|x| + \frac{1}{2}(x-a)^2 \right\}. \tag{2.1}$$

Theorem 11. *Let $\lambda > 0$ be fixed. For each $a \in \mathbb{R}$, there is one and only one solution $S_\lambda(a)$, to the minimization problem (2.1). Furthermore,*

$$S_\lambda(a) = \begin{cases} a-\lambda, & a > \lambda \\ 0, & |a| \le \lambda \\ a+\lambda, & a < -\lambda \end{cases}.$$

Proof. Let $f(x) = \lambda|x| + \frac{1}{2}(x-a)^2$. Note that $f(x) \to \infty$ when $|x| \to \infty$ and f is continuous on $\mathbb{R}$ and differentiable everywhere except a single point $x = 0$. So, f achieves its minimum value on $\mathbb{R}$. Let $x^* = \arg\min_{x\in\mathbb{R}} f(x)$.

We consider three cases.

Case 1: Let $x^* > 0$. Since f is differentiable at $x = x^*$ and achieves its minimum, we must have $f'(x^*) = 0$. Note that, for $x > 0$, we have

$$f'(x) = \frac{d}{dx}(\lambda x + \frac{1}{2}(x-a)^2) = \lambda + (x-a).$$

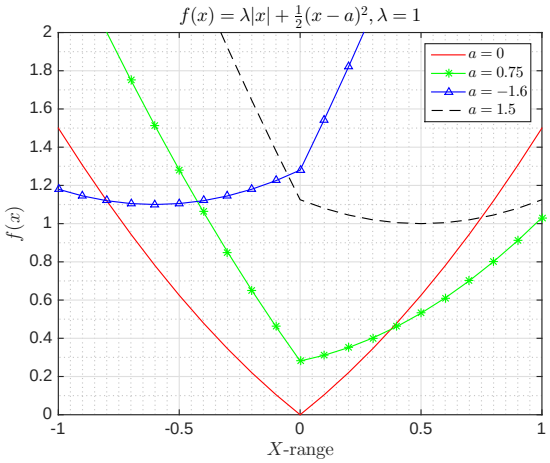

Figure 2.1: Plots of $f(x)$ for different values of a with $\lambda = 1$.

So,

$$\lambda + (x^* - a) = 0,$$

which implies

$$x^* = a - \lambda.$$

To be consistent with $x^* > 0$, we must require $a - \lambda > 0$ or, equivalently, $a > \lambda$.

Case 2: Let $x^* < 0$. By proceeding similarly as in Case 1 above, we can arrive at

$$x^* = a + \lambda \ \text{ with } \ a < -\lambda.$$

Case 3: Let $x^* = 0$. Note that $f(x)$ is no longer differentiable at $x = 0$ (So we could not use the condition $f'(x^*) = 0$ as before). But since f has a minimum at $x^* = 0$ and since f is differentiable on each side of $x^* = 0$, we must have

$$f'(x) > 0 \ \text{ for } \ x > 0 \ \text{ and } \ f'(x) < 0 \text{ for } \ x < 0.$$

So,

$$\lambda + x - a > 0 \ \text{ for } x > 0 \ \text{ and } \ -\lambda + x - a < 0 \ \text{ for } x < 0.$$

Thus,

$$\lambda - a > 0 \text{ and } -\lambda - a < 0,$$

or, equivalently,

$$|a| \leq \lambda.$$

To summarize, we have

$$x^* = \begin{cases} a - \lambda & \text{with } a > \lambda, \\ a + \lambda & \text{with } a < -\lambda, \\ 0 & \text{with } |a| \leq \lambda. \end{cases}$$

Since one and only one of the three cases (1) $a > \lambda$, (2) $a < -\lambda$, and (3) $|a| \leq \lambda$ holds, we obtain the uniqueness in general. With the uniqueness, it is straightforward to verify that each of the three cases would imply the corresponding formula for x^*. This completes the proof. □

2.2 A Sparse Recovery Problem

Recently, research in compressive sensing leads to the recognition that the ℓ_1-norm of a vector is a good substitute for the count of the number of non-zero entries of the vector in many minimization problems. In this section, we solve some simple minimization problems using the count of non-zero entries or ℓ_1-norm. Given a vector $v \in \mathbb{R}^n$, we want to solve

$$\min_{u \in \mathbb{R}^n} \{\text{card}(u) + \frac{\beta}{2}\|u - v\|_{\ell_2}^2\}, \tag{2.2}$$

where $\text{card}(u)$ denotes the number of non-zero entries of u, $\|\cdot\|_{\ell_2}$ denotes the Euclidean norm in $\mathbb{R}^n$, and $\beta > 0$ is a given balancing parameter. We can solve problem (2.2) componentwise (in each u_i) as follows. Notice that, given $u \in \mathbb{R}^n$, each entry u_i of u contributes 1 to $\text{card}(u)$ if u_i is non-zero, and contributes 0 if u_i is zero. If $v_i = 0$, then $u_i = 0$. We now will investigate the case when $v_i \neq 0$. Since we are minimizing $g(u) := \text{card}(u) + \frac{\beta}{2}\|u - v\|_{\ell_2}^2$, if u_i is zero then the contribution to $g(u)$ depending on this u_i is $\frac{\beta}{2}v_i^2$; otherwise, if u_i is

non-zero, then we should minimize $\frac{\beta}{2}(u_i - v_i)^2$ for $u_i \in \mathbb{R} \setminus \{0\}$, which forces that $u_i = v_i$ and contributes 1 to $g(u)$ as the minimum value. Combining all the cases, the solution u to problem (2.2) is given component-wise by

$$u_i = \begin{cases} 0, & \text{if } \frac{\beta}{2}(v_i)^2 \leq 1 \\ v_i, & \text{otherwise.} \end{cases}$$

Next, we replace card(u) by $\|u\|_{l_1}$ in (2.2) and solve:

$$\min_{u \in \mathbb{R}^n} [\|u\|_{\ell_1} + \frac{\beta}{2}\|u - v\|_{\ell_2}^2], \tag{2.3}$$

where $\|\cdot\|_{\ell_1}$ denotes the ℓ_1 norm in $\mathbb{R}^n$.

Using Theorem 11, we can solve (2.3) component-wise as follows.

Theorem 12. *[60] Let $\beta > 0$ and $v \in \mathbb{R}^n$ be given and let*

$$u^* = \arg\min_{u \in \mathbb{R}^n} [\|u\|_{\ell_1} + \frac{\beta}{2}\|u - v\|_{\ell_2}^2],$$

then

$$u^* = S_{1/\beta}(v),$$

where, $S_{1/\beta}(v)$ denotes the vector whose entries are obtained by applying the shrinkage function $S_{1/\beta}(\cdot)$ to the corresponding entries of v.

Proof. If u_i and v_i denote the ith entry of the vectors u and v, respectively, $i = 1, 2, \ldots, n$, then we have,

$$\begin{aligned} u^* &= \arg\min_{u \in \mathbb{R}^n} [\|u\|_{\ell_1} + \frac{\beta}{2}\|u - v\|_{\ell_2}^2] \\ &= \arg\min_{u \in \mathbb{R}^n} \sum_{i=1}^{n} |u_i| + \frac{\beta}{2}\sum_{i=1}^{n}(u_i - v_i)^2 \\ &= \arg\min_{u \in \mathbb{R}^n} \sum_{i=1}^{n} \left(|u_i| + \frac{\beta}{2}(u_i - v_i)^2 \right) \\ &= \arg\min_{u \in \mathbb{R}^n} \left\{ \sum_{i=1}^{n} \left(\frac{1}{\beta}|u_i| + \frac{1}{2}(u_i - v_i)^2 \right) \right\}. \end{aligned}$$

Since $|u_i|$ and $(u_i - v_i)^2$ are both nonnegative for all i, the vector $\mathbf{u}^*$ must have components u_i^* satisfying

$$u_i^* = \arg\min_{u_i^* \in \mathbb{R}} \{\frac{1}{\beta}|u_i| + \frac{1}{2}(u_i - v_i)^2\},$$

for $i = 1, 2, \ldots, n$. But by Proposition 1, the solution to each of these problems is given precisely by $S_{1/\beta}(v_i)$. This yields the result. $\square$

Remark 13. *The previous proof still works if we replace the vectors by matrices and use the extension of the norms ℓ_1 and ℓ_2 to matrices by treating them as vectors. By using the same argument we can obtain the following more general version of the previous theorem.*

Theorem 14. *[60] Let $\beta > 0$ and $V \in \mathbb{R}^{m \times n}$ be given. Then*

$$S_{1/\beta}(V) = \arg\min_{U \in \mathbb{R}^{m \times n}} \{\|U\|_{\ell_1} + \frac{\beta}{2}\|U - V\|_{\ell_2}^2\},$$

where $S_{1/\beta}(V)$ is again defined component-wise.

Theorem 14 solves the problem of approximating a given matrix by a sparse matrix by using the shrinkage function.

2.3 Solution to (1.55) via Problem (2.1)

We are ready to show how problem (1.55) is problem (2.1) in disguise. Given $\beta > 0$, using the unitary invariance of the Frobenius norm and the nuclear norm we have

$$\begin{aligned}
\min_X \{\|X\|_* + \frac{\beta}{2}\|X - A\|_F^2\} &= \min_X \{\|X\|_* + \frac{\beta}{2}\|X - U\tilde{A}V^T\|_F^2\} \\
&= \min_X \{\lambda\|X\|_* + \frac{1}{2}\|U(U^TXV - \tilde{A})V^T\|_F^2\} \\
&= \min_X \{\lambda\|X\|_* + \frac{1}{2}\sum_{i=1}^{\min\{m,n\}} \sigma_i(U(U^TXV - \tilde{A})V^T)^2\} \\
&= \min_X \{\lambda\|X\|_* + \frac{1}{2}\sum_{i=1}^{\min\{m,n\}} \sigma_i(U^TXV - \tilde{A})^2\} \\
&= \min_X \{\|U^TXV\|_* + \frac{\beta}{2}\|U^TXV - \tilde{A}\|_F^2\}.
\end{aligned}$$

It is now obvious from the last expression that the minimum occurs when U^TXV is diagonal since both terms in that expression get no larger when U^TXV is replaced by its diagonal matrix (with the help of (1.15)). So, the matrix $E = (e_{ij}) := U^TXV - \tilde{A}$ has no non-zero off-diagonal entries: $e_{ij} = 0$ if $i \neq j$. Thus,

$$X = U\tilde{X}V^T, \text{ with } \tilde{X} = \tilde{A} + E,$$

which yields a SVD of X (using the same matrices U and V as in a SVD of A !). Then,

$$\begin{aligned}\min_X\{\|X\|_* + \frac{\beta}{2}\|X - A\|_F^2\} &= \min_{\tilde{X}\in\text{diag}}\left\{\|\tilde{X}\|_* + \frac{\beta}{2}\|\tilde{X} - \tilde{A}\|_F^2\right\} \\ &= \min_{\tilde{X}\in\text{diag}}\left\{\sum_i \sigma(\tilde{X}) + \frac{\beta}{2}\sum_i(\sigma_i(\tilde{X}) - \sigma_i(\tilde{A}))^2\right\},\end{aligned}$$

where "diag" is the set of diagonal matrices in $\mathbb{R}^{m\times n}$. Above is an optimization problem like (2.1) (for vectors $(\sigma_1(\tilde{X}), \sigma_2(\tilde{X}), ...)^T$ as $\tilde{X}$ varies) whose solution is given by[1]

$$\sigma_i(\tilde{X}) = S_{1/\beta}(\sigma_i(\tilde{A})),\ i = 1, 2, ...$$

To summarize, we have proven Theorem 8.

Remark 15. *1. The most recent proof of this theorem is given by Cai, Candes, and Shen in [48] where they give an advanced verification of the result as discussed in the proof of Theorem 8. Our proof given above has the advantage that it is elementary and allows the reader to "discover" the result.*

2. There are many earlier discoveries of related results ([55]) where rank(X) is used instead of the nuclear norm $\|X\|_*$. We will examine one such variant in the next section.

3. One key ingredient in the above discussion is the unitary invariance of the norms $\|\cdot\|_*$ and $\|\cdot\|_F$. It was von Neumann (see, e.g., [66]) who was among the first to study the

[1]A careful reader will notice the additional requirement on $\sigma_i(\tilde{X})$: they are non-negative and sorted in descending order. Fortunately, this property can be automatically inherited from that of $\sigma_i(\tilde{A})$ and the monotone property of the shrinkage function.

family of all unitarily invariant matrix norms in matrix approximation, $\|\cdot\|_F$ being one of them.

4. *A closely related (but harder) problem is compressive sensing ([61, ?]). Readers are strongly recommended to the recently survey by Bryan and Leise ([59]).*

2.4 A Variation [5]

Some related problems can be solved by applying similar ideas. For example, let us consider a variant of a well-known result of Schmidt (see, e.g., [55, Section 5]), replacing the rank by the nuclear norm: For a fixed positive number τ, consider

$$\min_{X\in\mathbb{R}^{m\times n}} \|X-A\|_F \ \text{ subject to } \ \|X\|_* \le \tau. \tag{2.4}$$

Using similar methods as in the previous section, this problem can be transformed into the following:

$$\min_{u\in\mathbb{R}^{\min\{m,n\}}} \|u-v\|_{\ell_2} \ \text{ subject to } \ \|u\|_{\ell_1} \le \tau. \tag{2.5}$$

Note that, (2.5) related to a LASSO problem [58, 63, 62]. But unlike a LASSO problem, no special assumption is made on v in (2.5). In spite of this difference with LASSO, as in [58], one can form a *Lagrange relaxation* of (2.5), and solve the same problem as defined in Theorem 2:

$$u^* = \arg\min_{u\in\mathbb{R}^{\min\{m,n\}}} \{\frac{1}{2}\|u-v\|_{\ell_2}^2 + \lambda\|u\|_{\ell_1}\}, \ \text{ with } \|S_\lambda(v)\|_{\ell_1} = \tau, \tag{2.6}$$

which has a solution $u^* = S_\lambda(v)$. We will now verify this. It is easy to see that

$$\min_{u\in\mathbb{R}^{\min\{m,n\}}} \{\frac{1}{2}\|u-v\|_{\ell_2}^2 + \lambda\|u\|_{\ell_1}\} \le \frac{1}{2}\|u-v\|_{\ell_2}^2 + \lambda\|u\|_{\ell_1},$$

for all $u \in \mathbb{R}^{\min\{m,n\}}$. Since $S_\lambda(v)$ solves (2.6) we have,

$$\frac{1}{2}\|S_\lambda(v)-v\|_2^2 + \lambda\tau \le \frac{1}{2}\|u-v\|_{\ell_2}^2 + \lambda\|u\|_{\ell_1},$$

for all $u \in \mathbb{R}^{\min\{m,n\}}$; which implies,

$$\frac{1}{2}\|S_\lambda(v) - v\|_2^2 \leq \frac{1}{2}\|u - v\|_{\ell_2}^2 + \lambda(\|u\|_{\ell_1} - \tau),$$

for all $u \in \mathbb{R}^{\min\{m,n\}}$. Therefore,

$$\frac{1}{2}\|S_\lambda(v) - v\|_2^2 \leq \frac{1}{2}\|u - v\|_{\ell_2}^2,$$

for all $u \in \mathbb{R}^{\min\{m,n\}}$, such that $\|u\|_{\ell_1} \leq \tau$. Hence $u^* = S_\lambda(v)$ solves (2.5). We now give the following sketch of the derivation of converting (2.4) to (2.5): As in section 2.7.3, we use a SVD of A: Let $A = U\tilde{A}V^T$ be a SVD of A. Then,

$$\min_{X \in \mathbb{R}^{m\times n}} \|X - A\|_F = \min_{X \in \mathbb{R}^{m\times n}} \|U^T X V - \tilde{A}\|_F.$$

Note that, by the unitary invariance of matrix norm, $\|X\|_* = \|U^T X V\|_*$, so (2.4) can be written as

$$\min_{X \in \mathbb{R}^{m\times n}} \|X - \tilde{A}\|_F \ \text{ subject to } \ \|X\|_* \leq \tau,$$

which, by using (1.15), can be further transformed to

$$\min_{X \in \mathbb{R}^{m\times n}} \|X - \tilde{A}\|_F \ \text{ subject to } \ X \text{ being diagonal and } \|X\|_* \leq \tau. \tag{2.7}$$

Next, if we let u and v be two vectors in $\mathbb{R}^{\min\{m,n\}}$ consisting of the diagonal elements of X and $\tilde{A}$, respectively, then (2.7) is (2.5). Thus we have established the following result.

Theorem 16. *[5] With the notations above, the solution to problem (2.4) is given by*

$$\hat{X} = U S_\lambda(\tilde{A}) V^T,$$

for some λ such that $\|S_\lambda(\tilde{A})\|_{\ell_1} = \tau$.

CHAPTER THREE: WEIGHTED SINGULAR VALUE THRESHOLDING PROBLEM

In Chapter 1, we discussed the formulation of some classical low-rank approximation problems. Both classical PCA and SVT problems can be solved using closed form formulas based on SVD of the given matrix. However, if the Frobenius norm is replaced by the l_1 norm, no closed form is available for the solution (for example RPCA). This situation does not happen just to l_1 norm but to many other norms, including a weighted version of the Frobenius norm [37, 39].

In this chapter, we formulate a weighted low-rank approximation problem and discuss its numerical solution. We also present a detailed convergence analysis of our algorithm and, through numerical experiments on real data, we demonstrate the improvements in performance when weight is learned from the data over other state of the art methods.

3.1 Motivation Behind Our Problem: The Work of Golub, Hoffman, and Stewart

Recall that the solution to (1.1) suffers form the fact that none of the entries of A is preserved in the solution X. Let $A \in \mathbb{R}^{m\times n}$ be the given matrix with k fixed columns. Write A as $A = (A_1 \;\; A_2)$. In 1987, Golub, Hoffman, and Stewart were the first to consider the following *constrained* low rank approximation problem [1]:

Given $A = (A_1 \;\; A_2) \in \mathbb{R}^{m\times n}$ with $A_1 \in \mathbb{R}^{m\times k}$ and $A_2 \in \mathbb{R}^{m\times(n-k)}$, find $\tilde{A}_2$ such that (with $\tilde{A}_1 = A_1$)

$$(\tilde{A}_1 \;\; \tilde{A}_2) = \arg \min_{\substack{X_1, X_2 \\ \mathrm{r}(X_1 \;\; X_2)\le r \\ X_1 = A_1}} \|(A_1 \;\; A_2) - (X_1 \;\; X_2)\|_F^2. \tag{3.1}$$

That is, Golub, Hoffman, and Stewart required that the first few columns, A_1, of A must be preserved when one looks for a low rank approximation of $(A_1 \;\; A_2)$. As in the standard low

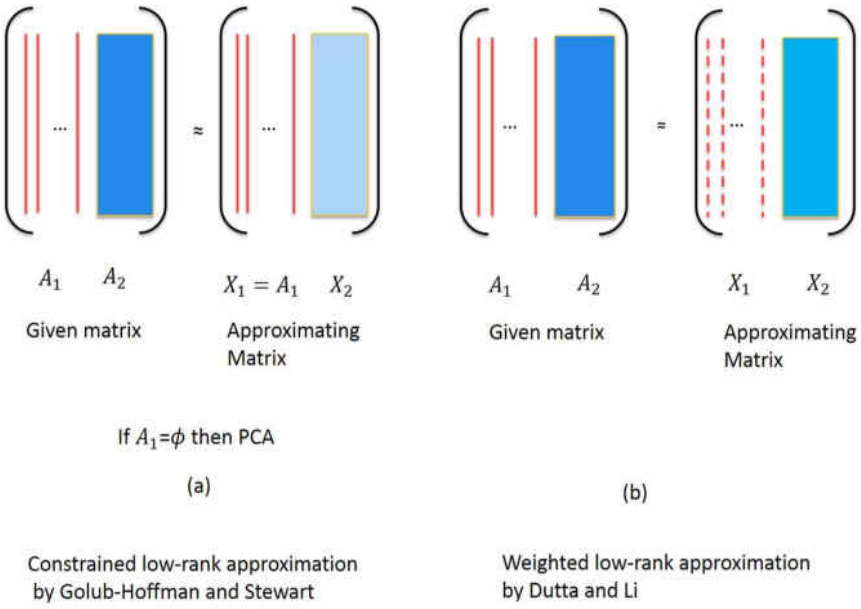

Figure 3.1: Visual interpretation of constrained low-rank approximation by Golub, Hoffman, and Stewart and weighted low-rank approximation by Dutta and Li.

rank approximation, the constrained low-rank approximation problem of Golub, Hoffman, and Stewart also has a closed form solution.

Theorem 17. *[1] With $k = \mathrm{r}(A_1)$ and $r \geq k$, the solutions $\tilde{A}_2$ in (3.1) are given by*

$$\tilde{A}_2 = P_{A_1}(A_2) + H_{r-k}\left(P^{\perp}_{A_1}(A_2)\right), \tag{3.2}$$

where P_{A_1} and $P^{\perp}_{A_1}$ are the projection operators to the column space of A_1 and its orthogonal complement, respectively.

Later in Chapter 4 we present a thorough proof of Theorem 17 as it is more appropriate to the context of that chapter. Recently, to solve background estimation problems, Xin et al. [42] proposed a supervised model learning algorithm. They assumed that some pure background frames are given and the data matrix A can be written into $A = (A_1\ A_2)$, where

A_1 contains the given pure background frames. Xin et al. required with $B = (B_1\ B_2)$ and $F = (F_1\ F_2)$ partitioned in the same way as in A, find B and F satisfying

$$\min_{\substack{B,F \\ B_1=A_1}} \left(\operatorname{rank}(B) + \|F\|_{gfl}\right),$$

where $\|\cdot\|_{gfl}$ denotes a norm that is a combination of l_1 norm and a local spatial total variation norm (to encourage connectivity of the foreground). Indeed, [42] further simplified the above model by assuming $\operatorname{rank}(B) = \operatorname{rank}(B_1)$. Since $B_1 = A_1$ and A_1 is given, so $r := \operatorname{rank}(B_1)$ is also given and thus, we can re-write the model of [42] as follows:

$$\min_{\substack{B=(B_1\ B_2) \\ \operatorname{rank}(B)\leq r \\ B_1=A_1}} \|A - B\|_{gfl}. \tag{3.3}$$

This formulation resembles the constrained low rank approximation problem of Golub et al. Inspired by Theorem 17 above and motivated by applications in which A_1 may contain noise, it makes more sense if we require $\|X_1 - A_1\|_F$ small (as in the case of the total least squares) instead of asking for $X_1 = A_1$. This leads us to consider the following problem: Let $\lambda > 0$ and $W_\lambda = \begin{pmatrix} \lambda I_k & 0 \\ 0 & I_{n-k} \end{pmatrix}$, find $(\hat{X}_1\ \ \hat{X}_2)$ such that

$$(\hat{X}_1\ \ \hat{X}_2) = \arg \min_{\substack{X_1,X_2 \\ \mathrm{r}(X_1\ \ X_2)\leq r}} \|\left((A_1\ \ A_2) - (X_1\ \ X_2)\right) W_\lambda\|_F^2. \tag{3.4}$$

This problem can be viewed as "approximately" preserving (controlled by a parameter λ), instead of requiring exactly matching, in the first few columns (see Figure 3.1).

Note that multiplying a matrix from right by W_λ is same as multiply λ to each element of the first k columns of that matrix and leaving the rest of the elements unchanged. As it turns out, this formulation can be viewed as generalized total least squares problem (GTLS) [23, 24]. Problem (3.4) is a special case of weighted low-rank approximation with a rank-one weight matrix and can be solved in closed form by using a single SVD of the given matrix $(\lambda A_1\ \ A_2)$ [23, 24]. A careful reader should also note that, both problems (3.1) and (3.4) can be cast as special cases of structured low-rank problems with element-wise weights [26, 31].

But what about an unconstrained version of the problem (3.4) where one can replace the rank constraint by its convex surrogate, the nuclear norm? Can it still be capable of making $\|X_1 - A_1\|_F$ small when one looks for a low-rank approximation of A? First, we will answer these questions. Indeed, as in a related work of background estimation from video sequence, shadow and specularity removal from face image, and domain adaptation problems in computer vision and machine learning in ([3]), this idea of unconstrained weighted low-rank approximation is shown to be more effective. An unconstrained version of (3.4) is:

$$\min_{X_1,X_2}\{\|\left((A_1 \;\; A_2) - (X_1 \;\; X_2)\right) W_\lambda\|_F^2 + \tau\|(X_1 \;\; X_2)\|_*\}, \tag{3.5}$$

where $\tau > 0$ is a balancing parameter. The above problem can be written as:

$$\min_{X=(X_1 \;\; X_2)}\{\lambda^2\|A_1 - X_1\|_F^2 + \|A_2 - X_2\|_F^2 + \tau\|X\|_*\}.$$

Let

$$\tilde{X} = \arg\min_X\{\lambda^2\|A_1 - X_1\|_F^2 + \|A_2 - X_2\|_F^2 + \tau\|X\|_*\},$$

and $\tilde{X} = (\tilde{X}_1 \;\; \tilde{X}_2)$ be a compatibale block partition. Therefore,

$$\begin{aligned}\lambda^2\|\tilde{X}_1 - A_1\|_F^2 &\leq \min_{X=(X_1 \;\; X_2)}\{\lambda^2\|A_1 - X_1\|_F^2 + \|A_2 - X_2\|_F^2 + \tau\|X\|_*\} \\ &\leq \|A_2\|_F^2 + \tau\|(A_1 \;\; 0)\|_*.\end{aligned}$$

The first inequality is due to the fact $\|\tilde{X}_2 - A_2\|_F^2 + \tau\|\tilde{X}\|_* \geq 0$. Since $X = (A_1 \;\; 0)$ is a special choice of X we obtain the second inequality. Denote $m := \|A_2\|_F^2 + \tau\|(A_1 \;\; 0)\|_*$ and we find

$$\lambda^2\|\tilde{X}_1 - A_1\|_F^2 \leq m.$$

As $\lambda \to \infty$ we have $\tilde{X}_1 \to A_1$. This shows problem (3.5) can also make $\|X_1 - A_1\|_F$ small as claimed in the formulation of its constrained version, problem (3.4). Note that (3.5) is a special unconstrained version of the problem (1.3), where the ordinary matrix multiplication is used and the weight $W_\lambda \in \mathbb{R}^{n\times n}$ is non-singular. A derivation of the above claim is provided in Chapter 4. Considering its resemblance to the classical singular value thresholding (SVT)

problem [48] one can denote problem (3.5) as the weighted SVT (WSVT) problem. Unlike SVT there is no closed form solution for problem (3.5), as $\|XW\|_* \neq \|X\|_*$, in general. In contrast to the many numerical methods ([39, 40, 41, 43, 45, 47]) for solving the weighted low-rank approximation problem (1.3), we are not aware of any numerical solutions to the weighted SVT problem. Based on the formulation of the problem (3.5) above, one of the main problem we will study in this chapter is the numerical solution to the WSVT problem. Our algorithm can solve problem (3.5) for any non-singular weight matrix W_λ. But in the numerical experiment section, we consider two computer vision applications where we use diagonal weight matrix. Depending on the nature of the problem, the multiplication of the diagonal weight matrix could be from left (when the rows of A need to be constrained) or from right (when the columns of A need to be constrained). In many real world applications, the data matrix is a "tall and skinny" matrix, which means it has more rows than columns. For example, in analyzing a video sequence for background estimation the columns of the test matrix is comprised of the video frames, where the total number of rows of A is the total number of pixels in each video frame (see Figure 3.3). So indeed this is the case when $m >> n$. In this chapter, we will study the case when the weight matrix is multiplied from the right.

The rest of the chapter is organized as follows. In Section 3.2, we propose a numerical algorithm to solve problem (3.5) for any general invertible weight matrix W using the fast and simple alternating direction method. In Section 3.3, we propose a numerical algorithm to solve problem (3.5) by using augmented Lagrange multiplier method. In Section 3.4, we present the convergence analysis of our proposed algorithm in Section 3.3. Qualitative and quantitative results demonstrating the efficiency of our algorithm on some real world computer vision applications, using a special diagonal weight matrix W are given in Section 3.5.

3.1.1 Formulation of the Problem

Given a target matrix $A = (a_{ij}) \in \mathbb{R}^{m\times n}$ and a weight matrix $W = (w_{ij}) \in \mathbb{R}_+^{n\times n}$ with non negative entries. Assume that W is invertible and $m >> n$. Our goal is to find a low rank matrix $X = (x_{ij}) \in \mathbb{R}^{m\times n}$ of rank less than or equal to a given integer r, (where necessarily r$(A) \geq r$) such that the matrix X is the best approximation to A under the weighted Frobenious norm. That is,

$$B = \arg\min_X \|(A - X)W\|_F^2; \text{ subject to } \mathrm{r}(X) \leq r. \tag{3.6}$$

Using the nuclear norm a related unconstrained convex relaxation of the above problem is

$$\begin{aligned} B &= \arg\min_X \{\frac{1}{2}\|(A - X)W\|_F^2 + \tau\|X\|_*\} \\ &= \arg\min_X \{\frac{1}{2}\|AW - XW\|_F^2 + \tau\|X\|_*\}. \end{aligned} \tag{3.7}$$

3.2 A Numerical Algorithm for Weighted SVT Problem

We propose to introduce auxiliary variables and use alternating direction method to solve (3.7). The novelty of our weighted SVT algorithm (WSVT) is that by using auxiliary variables, we can employ the simple and fast alternating direction method (ADM) to numerically solve the minimization problem (3.7). Denote $XW = C \in \mathbb{R}^{m\times n}$ and as W is non-singular we can rewrite (3.7) as

$$\begin{aligned} &\min_C \{\frac{1}{2}\|AW - C\|_F^2 + \tau\|XWW^{-1}\|_*\} \\ &= \min_C \{\frac{1}{2}\|AW - C\|_F^2 + \tau\|CW^{-1}\|_*\}, \end{aligned}$$

write $D = CW^{-1}$ in the above to get

$$\min_{C,D} \{\frac{1}{2}\|AW - C\|_F^2 + \tau\|D\|_*\},$$

subject to $CW^{-1} = D$. A regularized version of the above problem can be written as:

$$\min_{C,D} \{\frac{1}{2}\|AW - C\|_F^2 + \tau\|D\|_* + \frac{\mu}{2}\|D - CW^{-1}\|_F^2\}, \tag{3.8}$$

where $\mu \geq 0$ is a fixed balancing parameter. If $(\hat{C}, \hat{D})$ solves (3.8) then we have

$$(\hat{C}, \hat{D}) = \arg\min_{C,D} h(C, D) = \arg\min_{C,D} \{\frac{1}{2}\|AW - C\|_F^2 + \tau\|D\|_* + \frac{\mu}{2}\|D - CW^{-1}\|_F^2\},$$

where $h(C, D) = \frac{1}{2}\|AW - C\|_F^2 + \tau\|D\|_* + \frac{\mu}{2}\|D - CW^{-1}\|_F^2$ is a convex function and we can justify our claim by using the following argument: Let $h(C, D) = h_1(C, D) + h_2(C, D)$ where $h_1(C, D) = \frac{1}{2}\|AW - C\|_F^2 + \frac{\mu}{2}\|D - CW^{-1}\|_F^2$, and $h_2(C, D) = \tau\|D\|_*$. One way to show $h(C, D)$ is convex is to use the well known fact that if a function $f(x)$ is convex then $f(\beta x + (1 - \beta)y) \leq \beta f(x) + (1 - \beta)f(y)$ for $0 \leq \beta \leq 1$. We need to use the following result: for $A, B \in \mathbb{R}^{m\times n}$, we have

$$\begin{aligned}
\|A + B\|_F^2 &= \text{trace}((A + B)^T(A + B)) \\
&= \text{trace}((A^T + B^T)(A + B)) \\
&= \text{trace}(A^TA + B^TA + A^TB + B^TB)) \\
&\leq \text{trace}(A^TA) + \text{trace}(B^TB) \\
&= \|A\|_F^2 + \|B\|_F^2.
\end{aligned}$$

Consider the linear combinations of C_1,C_2 and D_1,D_2 with respect to the parameter $0 \leq \alpha \leq 1$. Using the above result on $h(\alpha C_1 + (1 - \alpha)C_2, \alpha D_1 + (1 - \alpha)D_2)$ we find:

$$\begin{aligned}
&h(\alpha C_1 + (1 - \alpha)C_2, \alpha D_1 + (1 - \alpha)D_2) \\
&= \frac{1}{2}\|WA - \alpha C_1 - (1 - \alpha)C_2\|_F^2 + \tau\|\alpha D_1 + (1 - \alpha)D_2\|_* \\
&+ \frac{\mu}{2}\|\alpha D_1 + (1 - \alpha)D_2 - \alpha W^{-1}C_1 - (1 - \alpha)W^{-1}C_2\|_F^2 \\
&= \frac{1}{2}\|(\alpha + 1 - \alpha)WA - \alpha C_1 - (1 - \alpha)C_2\|_F^2 + \tau\|\alpha D_1 + (1 - \alpha)D_2\|_* \\
&+ \frac{\mu}{2}\|\alpha D_1 + (1 - \alpha)D_2 - \alpha W^{-1}C_1 - (1 - \alpha)W^{-1}C_2\|_F^2 \\
&= \frac{1}{2}\|\alpha WA + (1 - \alpha)WA - \alpha C_1 - (1 - \alpha)C_2\|_F^2 + \tau\|\alpha D_1 + (1 - \alpha)D_2\|_* \\
&+ \frac{\mu}{2}\|\alpha D_1 + (1 - \alpha)D_2 - \alpha W^{-1}C_1 - (1 - \alpha)W^{-1}C_2\|_F^2
\end{aligned}$$

$$\leq \frac{\alpha}{2}\|WA - C_1\|_F^2 + \left(\frac{1-\alpha}{2}\right)\|WA - C_2\|_F^2 + \tau\alpha\|D_1\|_* + \tau(1-\alpha)\|D_2\|_*$$
$$+ \frac{\alpha\mu}{2}\|D_1 - W^{-1}C_1\|_F^2 + \left(\frac{(1-\alpha)\mu}{2}\right)\|D_2 - W^{-1}C_2\|_F^2$$
$$= \alpha h(C_1, D_1) + (1-\alpha)h(C_2, D_2).$$

Hence our claim is justified.

Since $h(C, D)$ is convex it has a unique minimizer and $(\hat{C}, \hat{D})$ minimizes $h(C, D)$ if and only if

$$0 \in \partial_{(C,D)}h(\hat{C}, \hat{D}) \quad \text{implies} \quad 0 = \frac{\partial}{\partial C}h(\hat{C}, \hat{D}) \text{ and } 0 \in \partial_D h(\hat{C}, \hat{D}).$$

The first optimality condition gives

$$\hat{C} - AW + \mu(\hat{C}W^{-1} - \hat{D})(W^{-1})^T = 0, \tag{3.9}$$

that is,

$$\hat{C}(I_n + \mu(W^TW)^{-1}) = AW + \mu\hat{D}(W^T)^{-1},$$

which after solving for $\hat{C}$ gives (since $(I_m + \mu(W^TW)^{-1})$ is invertible for a positive μ),

$$\hat{C} = (AW + \mu\hat{D}(W^T)^{-1})(I_n + \mu(W^TW)^{-1})^{-1}.$$

From the second optimality condition we find

$$0 \in \tau\partial\|\hat{D}\|_* + \mu(\hat{D} - W^{-1}\hat{C}),$$

which is a typical SVT problem. Using the well known result of Cai-Candes-Shen [48] we can write

$$US_{\frac{\tau}{\mu}}(\Sigma)V^T = \arg\min_D\{\frac{\mu}{2}\|D - \hat{C}W^{-1}\|_F^2 + \tau\|D\|_*\},$$

where $U\Sigma V^T$ is a SVD of $\hat{C}W^{-1}$.

In summary we have, $\hat{C} = (AW + \mu\hat{D}(W^T)^{-1})(I_n + \mu(W^TW)^{-1})^{-1}$ and $\hat{D} = US_{\frac{\tau}{\mu}}(\Sigma)V^T$ where $U\Sigma V^T$ be a SVD of $\hat{C}W^{-1}$. Therefore, our algorithm is:

Algorithm 1: WSVT algorithm

Input : $A \in \mathbb{R}^{m\times n}$, weight matrix $W \in \mathbb{R}_+^{m\times m}$ and $\tau > 0$, $\rho > 1$;

Initialize: $C = AW, D = A, Y = 0; \mu > 0$;

while *not converged* **do**

  $C_{k+1} = (AW + \mu D(W^T)^{-1})(I_n + \mu(W^TW)^{-1})^{-1}$;

  $[U \ \ \Sigma \ \ V] = SVD(CW^{-1})$;

  $D = US_{\frac{\tau}{\mu}}(\Sigma)V^T$;

  $\mu = \rho\mu$;

end

Output : $X = CW^{-1}$

3.3 Augmented Lagrange Multiplier Method

In this section we use the classic augmented Lagrange multiplier method to solve (3.7). As proposed in Section 3.2, first we introduce the auxiliary variables $XW = C$, and $CW^{-1} = D$ to make the alternating direction method applicable. After introducing the auxiliary variables the augmented Lagrange function for the minimization problem (3.7) is

$$L(C, D, Y, \mu) = \frac{1}{2}\|AW - C\|_F^2 + \tau\|D\|_* + \langle Y, D - CW^{-1}\rangle + \frac{\mu}{2}\|D - CW^{-1}\|_F^2, \quad (3.10)$$

where $Y \in \mathbb{R}^{m\times n}$ is the Lagrange multiplier and μ and τ are two positive balancing parameters. If $(\hat{C}, \hat{D})$ be a solution to (3.10) then

$$(\hat{C}, \hat{D}) = \arg\min_{C,D} L(C, D, Y, \mu).$$

The solution can be approximated using an alternating strategy of minimizing the augmented Lagrange function with respect each component iteratively via the following rule: At $(k+1)$th

iteration do:

$$\begin{cases} C_{k+1} = \arg\min_C L(C, D_k, Y_k, \mu_k), \\ D_{k+1} = \arg\min_D L(C_{k+1}, D, Y_k, \mu_k), \\ Y_{k+1} = Y_k + \mu_k(D_{k+1} - C_{k+1}W^{-1}), \end{cases}$$

where (C_k, D_k, Y_k) is the given triple of iterate. We begin by completing the square on (3.10):

$$\begin{aligned} L(C, D, Y, \mu) &= \frac{1}{2}\|AW - C\|_F^2 + \tau\|D\|_* + \langle Y, D - CW^{-1}\rangle + \frac{\mu}{2}\|D - CW^{-1}\|_F^2 \\ &= \frac{1}{2}\|AW - C\|_F^2 + \tau\|D\|_* + \frac{\mu}{2}(\|D - CW^{-1}\|_F^2 + \frac{2}{\mu}\langle Y, D - CW^{-1}\rangle \\ &+ \frac{1}{\mu^2}\|Y\|_F^2) - \frac{1}{2\mu}\|Y\|_F^2 \\ &= \frac{1}{2}\|AW - C\|_F^2 + \tau\|D\|_* + \frac{\mu}{2}\|D - CW^{-1} + \frac{1}{\mu}Y\|_F^2 - \frac{1}{2\mu}\|Y\|_F^2. \end{aligned}$$

Note that, by completing the squares, we have

$$\begin{aligned} \arg\min_C L(C, D_k, Y_k, \mu_k) &= \arg\min_C \{\frac{1}{2}\|AW - C\|_F^2 + \frac{\mu_k}{2}\|D_k - CW^{-1} + \frac{1}{\mu_k}Y_k\|_F^2\}, \\ \arg\min_D L(C_{k+1}, D, Y_k, \mu_k) &= \arg\min_D \{\tau\|D\|_* + \frac{\mu_k}{2}\|D - C_{k+1}W^{-1} + \frac{1}{\mu_k}Y_k\|_F^2\}. \end{aligned}$$

Since $L(C, D, Y, \mu)$ is a convex function in the argument C and D, it has a unique minimizer and $(\hat{C}, \hat{D})$ minimizes $L(C, D, Y, \mu)$ if and only if

$$0 \in \partial_{(C,D)} L(\hat{C}, \hat{D}, Y, \mu) \quad \text{which implies,} \quad 0 = \frac{\partial}{\partial C} L(\hat{C}, \hat{D}, Y, \mu) \text{ and } 0 \in \partial_D L(\hat{C}, \hat{D}, Y, \mu).$$

Note that,

$$\frac{\partial}{\partial C} L(\hat{C}, \hat{D}, Y, \mu) = \hat{C} - AW + \mu(\hat{C}W^{-1} - \hat{D} - \frac{1}{\mu}Y)(W^{-1})^T,$$

which after solving for $\hat{C}$ yields (since the matrix $(I_n + \mu(WW^T)^{-1})$ is invertible for $\mu \geq 0$)

$$\hat{C} = (AW + \mu\hat{D}(W^T)^{-1} + Y(W^T)^{-1})(I_n + \mu(W^TW)^{-1})^{-1}.$$

The second optimality condition gives,

$$0 \in \partial_D L(\hat{C}, \hat{D}, Y, \mu), \text{ which is, } 0 \in \tau\partial\|\hat{D}\|_* + \mu(\hat{D} - \hat{C}W^{-1} + \frac{1}{\mu}Y).$$

Using the well known result from Cai-Candes-Shen [48] we have

$$US_{\frac{\tau}{\mu}}(\Sigma)V^T = \arg\min_D \{\frac{\mu}{2}\|D - \hat{C}W^{-1} + \frac{1}{\mu}Y\|_F^2 + \tau\|D\|_*\}$$

where $U\Sigma V^T$ is a SVD of $\hat{C}W^{-1} - \frac{1}{\mu}Y$. Therefore, we propose Algorithm 2.

Algorithm 2: WSVT Algorithm: Augmented Lagrange Multiplier Method

Input : $A \in \mathbb{R}^{m\times n}$, weight matrix $W \in \mathbb{R}_+^{n\times n}$ and $\tau > 0, \rho > 1$;

Initialize: $C = AW, D = A, Y = 0; \mu > 0$;

while *not converged* **do**

  $C = (XW + \mu D(W^T)^{-1} + Y(W^{-1})^T)(I_n + \mu(W^TW)^{-1})^{-1}$;

  $[U \ \ \Sigma \ \ V] = SVD(CW^{-1} - \frac{1}{\mu}Y)$;

  $D = US_{\frac{\tau}{\mu}}(\Sigma)V^T$;

  $Y = Y + \mu(D - CW^{-1})$;

  $\mu = \rho\mu$;

end

Output : $X = CW^{-1}$

3.4 Convergence of the Algorithm

In this section, we will establish the convergence of Algorithm 2. To do so, we will take advantage of the special form of our augmented Lagrangian function $L(C, D, Y, \mu)$ in Section 3.3. We follow the main ideas from [7, 9, 44]. We will also use the same notation as defined in the previous section. Recall that $Y_{k+1} = Y_k + \mu_k(D_{k+1} - C_{k+1}W^{-1})$ and define $\hat{Y}_{k+1} := Y_k + \mu_k(D_k - C_{k+1}W^{-1})$. Also note that for $\rho > 1$, $\{\mu_k\}$ is an increasing geometric sequence. We will require the situation when

$$\sum_k \frac{1}{\mu_k} < \infty$$

to prove the convergence results.

Theorem 18. *We have*

1. *The sequences* $\{C_k\}$ *and* $\{D_k\}$ *are convergent. Moreover, if* $\lim_{k\to\infty}\{C_k\} = C_\infty$ *and* $\lim_{k\to\infty}\{D_k\} = D_\infty$*, then* $C_\infty = D_\infty W$ *with*

$$\|D_k - C_k W^{-1}\| \leq \frac{C}{\mu_k}, \quad k = 1, 2, \cdots,$$

for some constant C *independent of* k*.*

2. *If* $L_{k+1} := L(C_{k+1}, D_{k+1}, Y_k, \mu_k)$*, then the sequence* $\{L_k\}$ *is bounded above and*

$$L_{k+1} - L_k \leq \frac{\mu_k + \mu_{k-1}}{2}\|D_k - C_k W^{-1}\|_F^2 = O(\frac{1}{\mu_k}), \quad \textit{for } k = 1, 2, \cdots.$$

Theorem 19. *Let* (C_∞, D_∞) *be the limit point of* (C_k, D_k) *and define*

$$f_\infty = \frac{1}{2}\|AW - C_\infty\|_F^2 + \tau\|D_\infty\|_*.$$

Then $C_\infty = D_\infty W$ *and*

$$-O(\mu_{k-1}^{-2}) \leq \frac{1}{2}\|AW - C_k\|_F^2 + \tau\|D_k\|_* - f_\infty \leq O(\mu_{k-1}^{-1}).$$

To establish our main results, we need two lemmas.

Lemma 20. *The sequence* $\{Y_k\}$ *is bounded.*

The boundedness of the sequence $\{\hat{Y}_k\}$ is true but requires a different argument.

Lemma 21. *We have the following:*

1. *The sequence* $\{C_k\}$ *is bounded.*

2. *The sequence* $\{\hat{Y}_k\}$ *is bounded.*

3.4.1 Proofs

We need the following lemma (see also [9]).

Lemma 22. *[46] Let $P \in \mathbb{R}^{m\times n}$ and $\|\cdot\|$ be a unitary invariant matrix norm. Let $Q \in \mathbb{R}^{m\times n}$ be such that $Q \in \partial\|P\|$, where $\partial\|P\|$ denotes the set of subdifferentials of $\|\cdot\|$ at P. Then $\|Q\|^* \leq 1$; where $\|\cdot\|^*$ is the dual norm of $\|\cdot\|$.*

Proof of Lemma 20. By the optimality condition for D_{k+1} we have, $0 \in \partial_D L(C_{k+1}, D_{k+1}, Y_k, \mu_k)$. So,

$$0 \quad \in \quad \tau\partial\|D_{k+1}\|_* + Y_k + \mu_k(D_{k+1} - C_{k+1}W^{-1}).$$

Therefore, $-Y_{k+1} \in \tau\partial\|D_{k+1}\|_*$. By using Lemma 22, we conclude that the sequence $\{Y_k\}$ is bounded by τ in the dual norm of $\|\cdot\|_*$. But the dual of $\|\cdot\|_*$ is the spectral norm, $\|\cdot\|_2$. So $\|Y_{k+1}\|_2 \leq \tau$. Hence $\{Y_k\}$ is bounded. $\square$

Proof of Lemma 21. We start with the optimality of C_{k+1}:

$$0 = \frac{\partial}{\partial C}L(C_{k+1}, D_k, Y_k, \mu_k).$$

We get

$$(C_{k+1}W^{-1} - A)WW^T = Y_k + \mu_k(D_k - C_{k+1}W^{-1}), \tag{3.11}$$

which equals $\hat{Y}_{k+1}$ by our definition at the beginning of this section.

1. Solving for D_k in (3.11), we arrive at

$$D_k = C_{k+1}(W^{-1} + \frac{1}{\mu_k}W^T) - \frac{1}{\mu_k}(AWW^T - Y_k).$$

Next, using the definition of $\{Y_k\}$ to write

$$D_k = C_kW^{-1} - \frac{1}{\mu_{k-1}}Y_{k-1} + \frac{1}{\mu_{k-1}}Y_k$$

and now equating the two expressions for D_k to obtain

$$C_kW^{-1} - \frac{1}{\mu_{k-1}}Y_{k-1} + \frac{1}{\mu_{k-1}}Y_k = C_{k+1}(W^{-1} + \frac{1}{\mu_k}W^T) - \frac{1}{\mu_k}(AWW^T - Y_k),$$

which after post multiplying throughout by W leads to

$$C_k - \frac{1}{\mu_{k-1}} Y_{k-1} W + \frac{1}{\mu_{k-1}} Y_k W = C_{k+1}(I_n + \frac{1}{\mu_k} W^T W) - \frac{1}{\mu_k}(AWW^T - Y_k)W,$$

and can be simplified further to

$$C_k - \frac{1}{\mu_{k-1}} Y_{k-1} W = C_{k+1}(I_n + \frac{1}{\mu_k} W^T W) - \frac{1}{\mu_k} AWW^T W,$$

To simplify the notations, we will use $O(\frac{1}{\mu_k})$ to denote matrices whose norm is bounded by a constant (independent of k) times $\frac{1}{\mu_k}$. Note that, for a fixed W the matrix AWW^TW is a constant matrix. So, by using the boundedness of $\{Y_k\}$, the above equation can be written as

$$C_{k+1}(I + \frac{1}{\mu_k} W^T W) = C_k + O(\frac{1}{\mu_k}). \tag{3.12}$$

Since W^TW is a symmetric positive definite matrix it is orthogonal diagonalizable. Diagonalize W^TW as $W^TW = Q\Lambda Q^T$, where $Q \in \mathbb{R}^{n\times n}$ be column orthogonal ($Q^TQ = I_n$) and use it in (3.12) to get

$$C_{k+1}(I_n + \frac{1}{\mu_k} Q\Lambda Q^T) = C_k + O(\frac{1}{\mu_k}),$$

which is

$$C_{k+1}(QQ^T + \frac{1}{\mu_k} Q\Lambda Q^T) = C_k + O(\frac{1}{\mu_k}),$$

and reduces to

$$C_{k+1}Q(I_n + \frac{1}{\mu_k}\Lambda) = C_kQ + O(\frac{1}{\mu_k}).$$

Taking the Frobenius norm on both sides and using the triangle inequality yield

$$\|C_{k+1}Q(I + \frac{1}{\mu_k}\Lambda)\|_F \leq \|C_kQ\|_F + O(\frac{1}{\mu_k}). \tag{3.13}$$

Since the diagonal matrix $I + \frac{1}{\mu_k}\Lambda$ has all diagonal entries no smaller than $1 + \lambda/\mu_k$ where $\lambda > 0$ denotes the smallest eigenvalue of W^TW, we see that

$$\|C_{k+1}Q\|_F \leq (1 + \frac{\lambda}{\mu_k})^{-1}\|C_{k+1}Q(I + \frac{1}{\mu_k}\Lambda)\|_F.$$

Thus, (3.13) implies

$$\|C_{k+1}Q\|_F \leq (1+\frac{\lambda}{\mu_k})^{-1}\|C_kQ\|_F + O(\frac{1}{\mu_k}),$$

which, by the unitary invariance of the norm, is equivalent to

$$\|C_{k+1}\|_F \leq (1+\frac{\lambda}{\mu_k})^{-1}\|C_k\|_F + \frac{C}{\mu_k} \text{ for all } k,$$

for some constant $C > 0$ independent of k. Finally, using the fact that $\mu_{k+1} = \rho\mu_k$ with $\rho > 1$, we see that the above inequality implies (by mathematical induction) that $\|C_k\|_F \leq C^*$ for some constant $C^* > 0$ (say, $C^* = C(\mu_0 + \lambda)/(\mu_0\lambda)$ works). This completes the proof of the boundedness of $\{C_k\}$.

2. Equation (3.11) gives us $\hat{Y}_{k+1} = (C_{k+1}W^{-1} - A)WW^T$, and so, the boundedness of $\{\hat{Y}_k\}$ follows immediately from the boundedness of $\{C_k\}$ established in 1 above. □

Proof of Theorem 18. 1. Since $Y_{k+1} - \hat{Y}_{k+1} = \mu_k(D_{k+1} - D_k)$ we have

$$D_{k+1} - D_k = \frac{1}{\mu_k}(Y_{k+1} - \hat{Y}_{k+1}).$$

So, by the boundedness of $\{Y_k\}$ and $\{\hat{Y}_k\}$, for all k,

$$\|D_{k+1} - D_k\| = \frac{1}{\mu_k}\|Y_{k+1} - \hat{Y}_{k+1}\| \leq \frac{2M}{\mu_k}.$$

There exists a $N > 0$ such that

$$\|\sum_{k=1}^{N}(D_{k+1} - D_k)\| \leq \sum_{k=1}^{N}\|D_{k+1} - D_k\| = \sum_{k=1}^{N}\|\frac{1}{\mu_k}(Y_{k+1} - \hat{Y}_{k+1})\| \leq \sum_{k=1}^{N}\frac{2M}{\mu_k}, \qquad (3.14)$$

where the first inequality is due to the triangle inequality. Hence, (3.14) implies, $\sum_{k=1}^{N}(D_{k+1} - D_k)$ is convergent if $\sum_{k=1}^{N}\frac{1}{\mu_k} < \infty$. Therefore, $\lim_{N\to\infty} D_N$ exists. Now, recall that

$$C_{k+1} = (AW + \mu_k D_k(W^{-1})^T + Y_k(W^{-1})^T)(I + \mu_k(W^TW)^{-1})^{-1}.$$

So, we see that $\{C_k\}$ is convergent as well and their limits satisfy

$$C_\infty W^{-1} = D_\infty.$$

Next, from the definition of $\{Y_k\}$, we have

$$\frac{1}{\mu_k}(Y_{k+1} - Y_k) = D_{k+1} - C_{k+1}W^{-1}.$$

Thus,

$$\|D_{k+1} - C_{k+1}W^{-1}\| = O(\frac{1}{\mu_k}). \tag{3.15}$$

Hence the result.□

2. We have,

$$\begin{aligned}
L_{k+1} &= L(C_{k+1}, D_{k+1}, Y_k, \mu_k) \\
&\leq L(C_{k+1}, D_k, Y_k, \mu_k) \\
&\leq L(C_k, D_k, Y_k, \mu_k) \\
&= \frac{1}{2}\|AW - C_k\|_F^2 + \tau\|D_k\|_* + \langle Y_k, D_k - C_kW^{-1}\rangle + \frac{\mu_k}{2}\|D_k - C_kW^{-1}\|_F^2 \\
&= \frac{1}{2}\|AW - C_k\|_F^2 + \tau\|D_k\|_* + \langle Y_{k-1}, D_k - C_kW^{-1}\rangle + \frac{\mu_{k-1}}{2}\|D_k - C_kW^{-1}\|_F^2 \\
&\quad + \langle Y_k - Y_{k-1}, D_k - C_kW^{-1}\rangle + \frac{\mu_k - \mu_{k-1}}{2}\|D_k - C_kW^{-1}\|_F^2 \\
&= L_k + \langle \mu_{k-1}(D_k - C_kW^{-1}), D_k - C_kW^{-1}\rangle + \frac{\mu_k - \mu_{k-1}}{2}\|D_k - C_kW^{-1}\|_F^2 \\
&= L_k + \mu_{k-1}\|D_k - C_kW^{-1}\|_F^2 + \frac{\mu_k - \mu_{k-1}}{2}\|D_k - C_kW^{-1}\|_F^2 \\
&= L_k + \frac{\mu_k + \mu_{k-1}}{2}\|D_k - C_kW^{-1}\|_F^2.
\end{aligned}$$

Therefore,

$$L_{k+1} - L_k \leq \frac{\mu_k + \mu_{k-1}}{2}\|D_k - C_kW^{-1}\|_F^2.$$

In addition to that we find

$$L_{k+1} - L_k \leq \frac{\mu_k + \mu_{k-1}}{2}\|D_k - C_kW^{-1}\|_F^2 \;\; = \;\; \frac{(1+\rho)}{2\mu_{k-1}}\|Y_k - Y_{k-1}\|_F^2.$$

Boundedness of the sequence $\{Y_K\}$ implies

$$L_{k+1} - L_k \leq O(\mu_{k-1}^{-1}), \text{ as } \frac{1}{\mu_k} \to 0, \;\; k \to \infty.$$

Hence the result. □

Proof of Theorem 19. By Theorem 18 (i) and by taking the limit as $k \to \infty$, we get

$$C_\infty W^{-1} = D_\infty. \tag{3.16}$$

Note that

$$\begin{aligned} L(C_k, D_k, Y_{k-1}, \mu_{k-1}) &= \min_{C,D} L(C, D, Y_{k-1}, \mu_{k-1}) \\ &\leq \min_{CW^{-1}=D} L(C, D, Y_{k-1}, \mu_{k-1}) \\ &\leq \|AW - C_\infty\|_F^2 + \tau\|D_\infty\|_* \\ &= f_\infty, \end{aligned} \tag{3.17}$$

where we applied (3.16) to get the last inequality. Note also that

$$\begin{aligned} &\|AW - C_k\|_F^2 + \tau\|D_k\|_* \\ &= L(C_k, D_k, Y_{k-1}, \mu_{k-1}) - \langle Y_{k-1}, D_k - C_k W^{-1}\rangle - \frac{\mu_{k-1}}{2}\|D_k - C_k W^{-1}\|_F^2, \end{aligned}$$

which, by using the definition of Y_k and (3.17), can be further rewritten into

$$\begin{aligned} &\|AW - C_k\|_F^2 + \tau\|D_k\|_* \\ &= L(C_k, D_k, Y_{k-1}, \mu_{k-1}) - \langle Y_{k-1}, \frac{1}{\mu_{k-1}}(Y_k - Y_{k-1})\rangle - \frac{\mu_{k-1}}{2}\|\frac{1}{\mu_{k-1}}(Y_k - Y_{k-1})\|_F^2 \\ &\leq f_\infty + \frac{1}{2\mu_{k-1}}(\|Y_{k-1}\|_F^2 - \|Y_k\|_F^2). \end{aligned} \tag{3.18}$$

Next, by using triangle inequality we get

$$\begin{aligned} &\|AW - C_k\|_F^2 + \tau\|D_k\|_* \\ &= \|AW - C_k + D_k W - D_k W\|_F^2 + \tau\|D_k\|_* \\ &\geq \|AW - D_k W\|_F^2 + \tau\|D_k\|_* - \|C_k - D_k W\|_F^2 \\ &\geq f_\infty - \|\frac{1}{\mu_{k-1}}(Y_{k-1} - Y_k)W\|_F^2 \\ &= f_\infty - \frac{1}{\mu_{k-1}^2}\|(Y_{k-1} - Y_k)W\|_F^2. \end{aligned} \tag{3.19}$$

Combining (3.18) and (3.19), we obtain the desired result. □

3.5 Numerical Experiments

In this section, we will demonstrate the performance of Algorithm 2 on two computer vision applications: background estimation from video sequences and shadow removal from face images under varying illumination. We will show that even with diagonal weight matrix W we can improve the performance as compared with other state-of-the-art unweighted low-rank algorithms. All experiments were performed on a computer with 3.1 GHz Intel Core i7 processor and 8GB memory.

3.5.1 Background Estimation from video sequences

Background estimation from video sequences is a classic computer vision problem. A robust background estimation model used for surveillance may efficiently deal with the dynamic foreground objects present in the video sequence. Additionally, it is expected to handle several other challenges, which include, but are not limited to: gradual or sudden change of illumination, a dynamic background containing non-stationary objects and a static foreground, camouflage, and sensor noise or compression artifacts. In these problems, one can consider if the camera motion is small, the scene in the background is presumably static; thus, the background component is expected to be the part of the matrix which is of low rank [50]. Minimizing the rank of the matrix A emphasizes the structure of the linear subspace containing the column space of the background. However, the exact desired rank is questionable, as a background of rank 1 is often unrealistic. For background estimation, we use three different sequences: the Stuttgart synthetic video data set [51], the airport sequence, and the fountain sequence [75]. We give qualitative analysis results on all three sequences. For performing quantitative analysis between different methods, we use the Stuttgart video sequence. It is a computer generated sequence from the vantage point of a static camera

located on the side of a building viewing a city intersection. The reason for choosing this sequence is two fold. First, this is a challenging video sequence which comprises both static and dynamic foreground objects and varying illumination in the background. Second, because of the availability of ample amount of ground truth, we can provide a rigorous quantitative comparison of the various methods. We choose the first 600 frames of the *BASIC* sequence to capture the changing illumination and foreground object. Correspondingly, we have 600 high quality ground truth frames. Frame numbers 551 to 600 have static foreground, and frame numbers 6 to 12 and 483 to 528 have no foreground. Given the sequence of 600

Figure 3.2: Sample frame from Stuttgart artificial video sequence.

test frames $\{I_1, I_2, \cdots I_{600}\}$ and corresponding 600 ground truth frames, each frame in the test sequence and in ground truth is resized to 64×80; originally they were 600×800. Each resized frame is stacked as a column vector of size 5120×1. We form the test matrix as $A = \{vec(I'_1), vec(I'_2), \cdots, vec(I'_{600})\}$, where $vec(I'_i) \in \mathbb{R}^{5120\times 1}$, $I'_i \in \mathbb{R}^{64\times 80}$, and $vec(\cdot) : \mathbb{R}^{64\times 80} \rightarrow \mathbb{R}^{5120\times 1}$ is an operator which maps the entries of $\mathbb{R}^{64\times 80}$ to a column vector $\mathbb{R}^{5120\times 1}$. Figure 3.2 shows a sample video frame from the Stuttgart video sequence and Figure 3.3 demonstrates an outline of processing the video frames defined above.

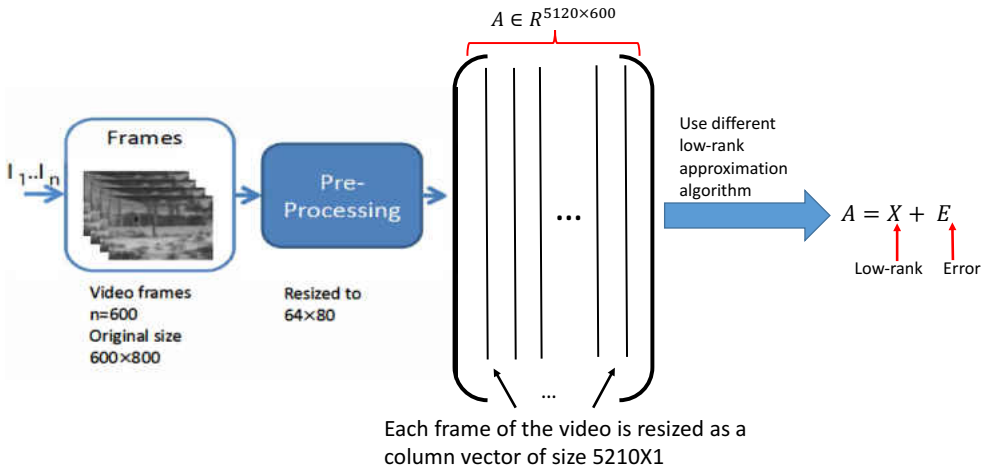

Figure 3.3: Processing the video frames.

We compare the performance of our algorithm to RPCA and SVT methods. We set a uniform threshold 10^{-7} for each method. For iEALM and APG we set $\lambda = 1/\sqrt{\max\{m,n\}}$, and for iEALM we choose $\mu = 1.5, \rho = 1.25$ as suggested in [9, 32, 49]. To choose the right set of parameters for WSVT we perform a grid search using a small holdout subset of frames. For WSVT we set $\tau = 4500$, $\mu = 5, \rho = 1.1$ for a fixed weight matrix W. For SVT we set $\tilde{\tau} = \tau/\mu$ since our method is equivalent to SVT for $W = I_n$. Next, we show the effectiveness of the weighted SVT and propose a mechanism for automatically estimating the weights from the data.

3.5.2 First Experiment: Can We Learn the Weight From the Data?

We present a mechanism for estimating the weights from the data for the weighted SVT. We use the heuristic that the data matrix A can be comprised of two blocks A_1 and A_2 such that A_1 mainly contains the information about the background frames which have the least foreground movements. However, the changing illumination, reflection, and noise are typically also a part of those frames and pose a lot of challenges. Our goal is to recover a low-rank matrix $X = (X_1 \quad X_2)$ with compatible block partition such that $X_1 \to A_1 + \epsilon$. Therefore, we want to choose a weight $\tilde{\lambda}$ corresponding to the frames of A_1. For this purpose,

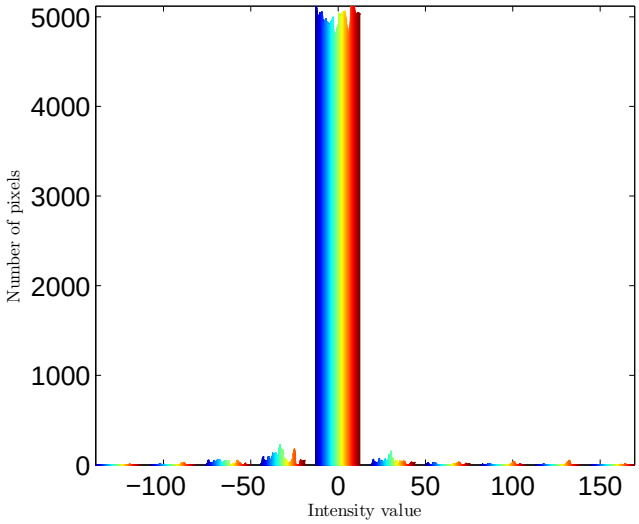

Figure 3.4: Histogram to chose the threshold ϵ_1.

the main idea is to have a coarse estimation of the background using an identity weight matrix, infer the weights from the coarse estimation, and then use the inferred weights to refine the background.

We denote the test matrix as T, and ground truth matrix as G. We borrow some notations from MATLAB to explain the experimental setup. The last 200 frames of the video sequence are chosen for this experiment because they contain static foreground (last 50 frames) along with moving foreground object and varying illumination. Jointly the different types of foreground objects and illumination pose a big challenge to the conventional SVT or RPCA algorithms.

We use our method with $W = I_n$ for 2 iterations on the frames and then detect the initial foreground F_{In}. We plot the histogram of our initially detected foreground to determine the threshold ϵ_1 of the intensity value. In our experiments we pick $\epsilon_1 = 31.2202$, the second smallest value of $|(F_{In})_{ij}|$, where $|\cdot|$ denotes the absolute value (see Figure 3.4). We replace everything below ϵ_1 by 0 in F_{In} and convert it in to a logical matrix LF_{In}. Arguably, for each such logical video frame, the number of pixels whose values are on (+1) is a good

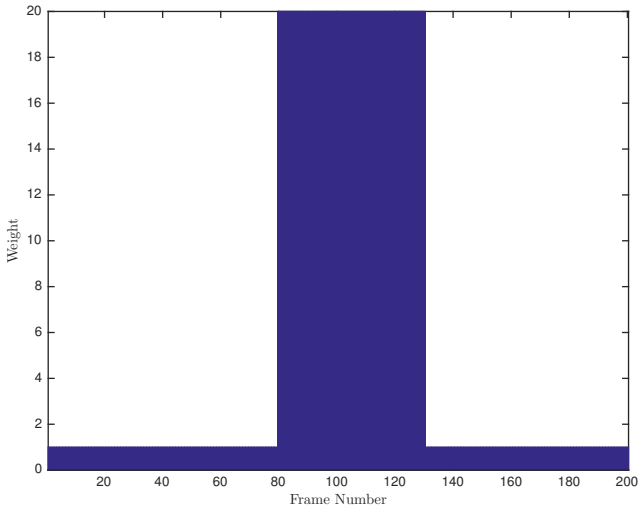

Figure 3.5: Diagonal of the weight matrix $W_{\tilde{\lambda}}$ with $\tilde{\lambda} = 20$ on the frames which has less than 5 foreground pixels and 1 elsewhere. The frame indexes are chosen from the set $\{\sum_i (LF_{IN})_{i1}, \sum_i (LF_{IN})_{i2}, \cdots \sum_i (LF_{IN})_{in}\}$.

indicator about whether the frame is mainly about the background. We thus set a weight $\tilde{\lambda}$ to the frames which has less than or equal to 5 foreground pixels and set a weight equal to 1 to other frames and formed the diagonal weight matrix $W_{\tilde{\lambda}}$. In Figure 3.5 we plot the diagonal of the weight matrix $W_{\tilde{\lambda}}$. Using our method defined above there is a weight $\tilde{\lambda} = 20$ is set to the frames which are contender of the best background frames. Figure 3.6 validates that we are able pick up the indexes correctly corresponding to the frames which has least foreground movement. Originally there are 48 frames in last 200 ground truth frames which has less than 5 pixels, our method picks up 51 frames. Next, we run our algorithm with weight as $W_{\tilde{\lambda}}$ and compare the performance with RPCA and SVT.

3.5.3 Second Experiment: Learning the Weight on the Entire Sequence

We perform the same procedure as defined in Section 3.5.2 on the entire video sequence. Figure 3.7 shows the histogram of our initially detected foreground to determine the threshold

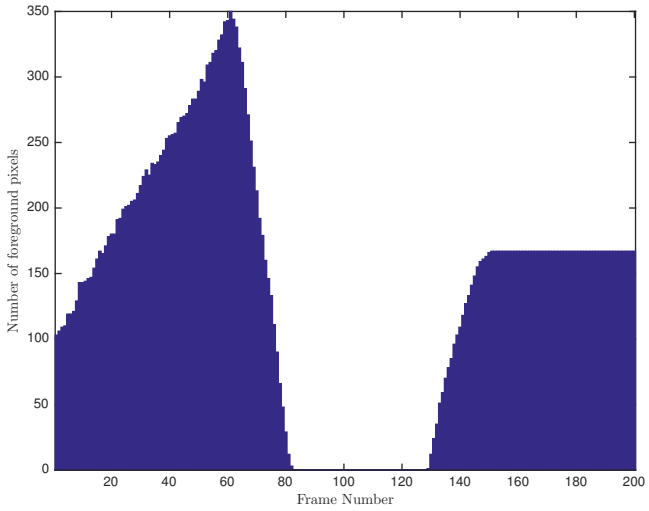

Figure 3.6: Original logical $G(:, 401:600)$ column sum. From the ground truth we estimated that there are 46 frames with no foreground movement and the frames 551 to 600 have static foreground.

ϵ_1 of the intensity value. In Figure 3.8 and 3.9 we show that using the method described in Section 3.5.2 we are able to distinguish the correct frame indexes with least foreground movement. Originally there are 57 frames in G which has less than 5 pixels, our method picks up 61 frames.

3.5.4 Third Experiment: Can We Learn the Weight More Robustly?

Since our approach of learning the weights in Sections 3.5.1 and 3.5.2 relies on extracting the initial background B_{In} and foreground F_{In} by performing the WSVT algorithm with $W = I_n$, it might not always make sense to specify the number of pixels manually for each test video sequence.

The initial success on learning the weights from Sections 3.5.1 and 3.5.2 motivates us to propose a robust alternative. As mentioned before, we use WSVT with $W = I_n$ for 2 iterations on the frames and detect the initial foreground F_{In} and background B_{In}. We

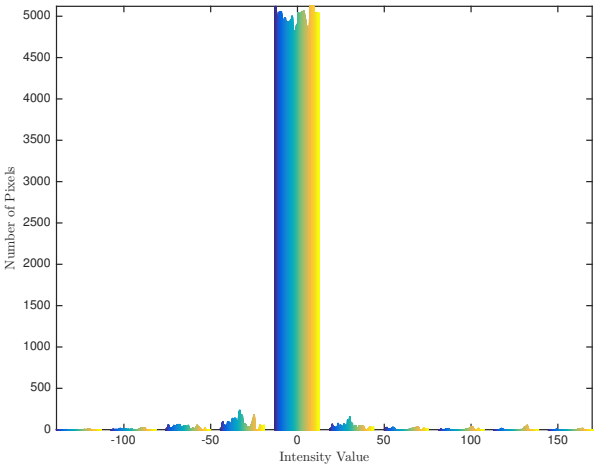

Figure 3.7: Histogram to chose the threshold $\epsilon_1' = 31.2202$.

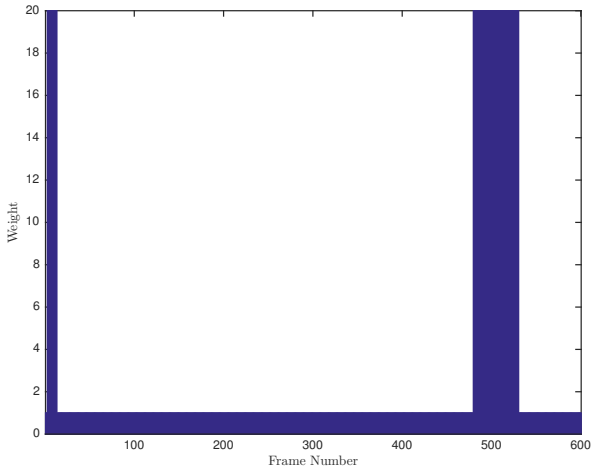

Figure 3.8: Diagonal of the weight matrix $W_{\tilde{\lambda}}$ with $\tilde{\lambda} = 20$ on the frames which has less than 5 foreground pixels and 1 elsewhere.

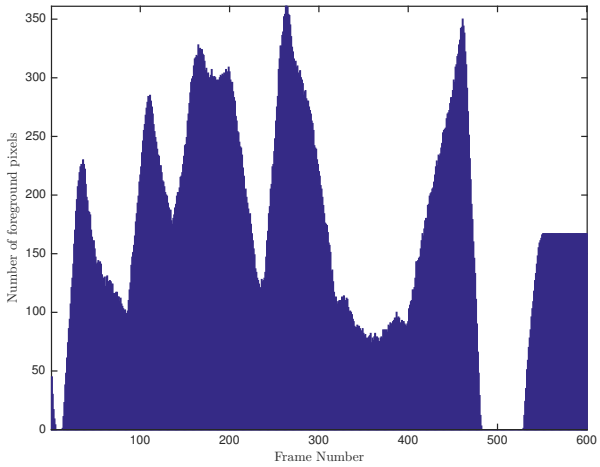

Figure 3.9: Original logical G column sum. From the ground truth we estimated that there are 53 frames with no foreground movement and the frames 551 to 600 have static foreground.

plot the histogram of our initially detected foreground to determine the threshold ϵ_1 of the intensity value. We replace everything below ϵ_1 by 0 in F_{In} and convert it into a logical matrix LF_{In}. We convert B_{In} directly to a logical matrix LB_{In}. We calculate the percentage score for each background and foreground frame and choose the threshold ϵ_2 as

$$\epsilon_2 := \text{mode}(\{\frac{\sum_i (LF_{IN})_{i1}}{\sum_i (LB_{IN})_{i1}}, \frac{\sum_i (LF_{IN})_{i2}}{\sum_i (LB_{IN})_{i2}}, \cdots, \frac{\sum_i (LF_{IN})_{in}}{\sum_i (LB_{IN})_{in}}\}),$$

and finally the frame indexes with least foreground movement are chosen from the following set:

$$I = \{i : (\frac{\sum_i (LF_{IN})_{i1}}{\sum_i (LB_{IN})_{i1}}, \frac{\sum_i (LF_{IN})_{i2}}{\sum_i (LB_{IN})_{i2}}, \cdots, \frac{\sum_i (LF_{IN})_{in}}{\sum_i (LB_{IN})_{in}}) \leq \epsilon_2\}.$$

Figures 3.10-3.13 demonstrate the percentage score plot for the Stuttgart video sequence, the fountain sequence, and the airport sequence. Comparing with the ground truth frames in Figures 3.6 and 3.9, we can see the effectiveness of the process on the Stuttgart video sequence. Using the percentage score, our method picks up 49 and 58 frame indexes respectively. For the airport sequence and fountain sequence our method selects 104 and 44 frames respectively. In the fountain sequence there is an almost static foreground object for the

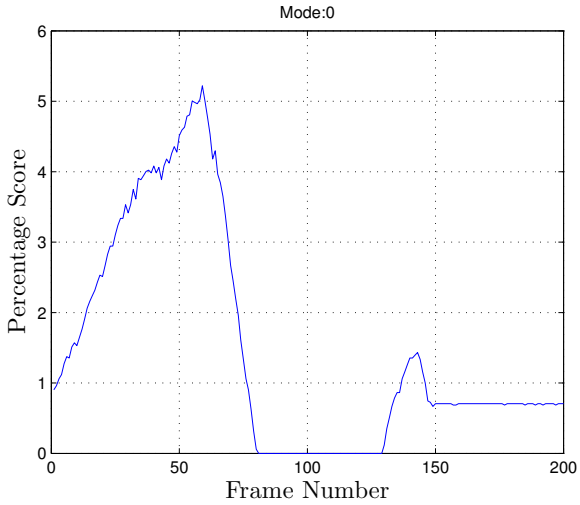

Figure 3.10: Percentage score versus frame number for Stuttgart video sequence. The method was performed on last 200 frames.

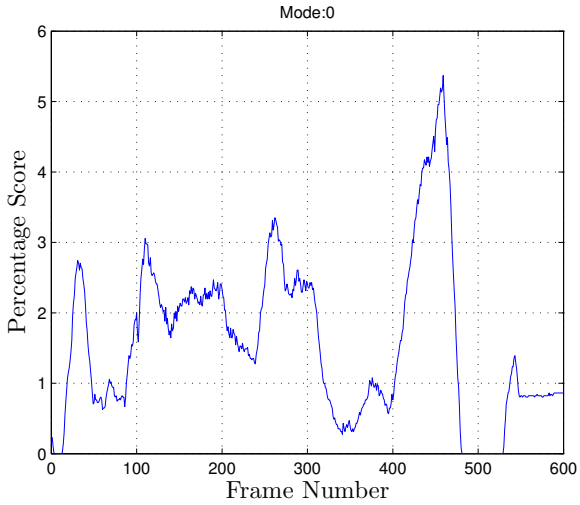

Figure 3.11: Percentage score versus frame number for Stuttgart video sequence. The method was performed on the entire sequence.

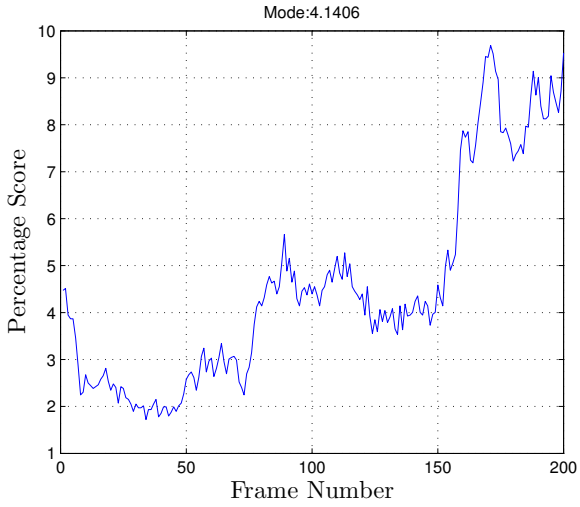

Figure 3.12: Percentage score versus frame number on first 200 frames for the fountain sequence.

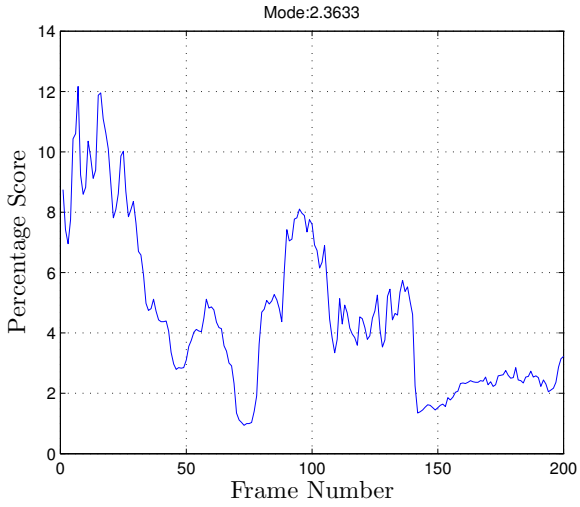

Figure 3.13: Percentage score versus frame number on first 200 frames for the airport sequence.

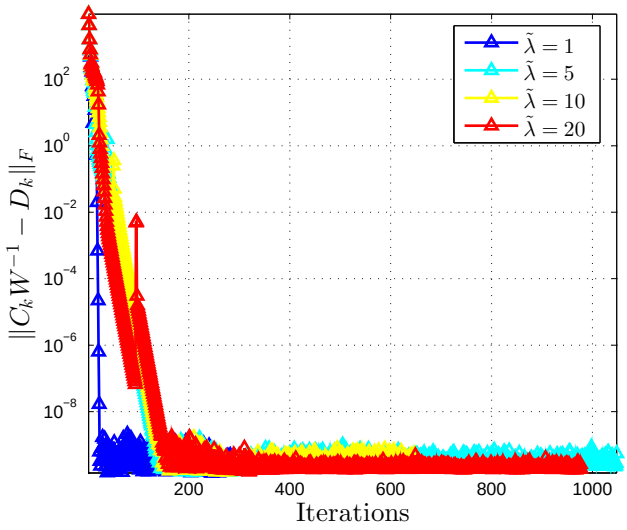

Figure 3.14: Iterations vs. $\mu_k \|D_k - C_k W^{-1}\|_F$ for $\tilde{\lambda} \in \{1,\ 5,\ 10,\ 20\}$

first 100 frames.

3.5.5 Convergence of the Algorithm

In Figure 3.14 and 3.15 we demonstrate the convergence of our algorithm as claimed in Theorem 18. For a given $\epsilon > 0$, the main stopping criteria of our WSVT algorithm is $|L_{k+1} - L_k| < \epsilon$ or if it reaches the maximum iteration. To demonstrate the convergence of our algorithm as claimed in Theorem 18, we run it on the entire Stuttgart artificial video sequence. The weights were chosen using the idea explained in Subsection 3.5.3. We choose $\tilde{\lambda} \in \{1, 5, 10, 20\}$ and ϵ is set to 10^{-7}. To conclude, in Figure 3.14 and 3.15, we show that for any $\tilde{\lambda} > 0$, there exists $\alpha, \beta \in \mathbb{R}$ such that $\|D_k - C_k W^{-1}\|_F \leq \alpha/\mu_k$ and $|L_{k+1} - L_k| \leq \beta/\mu_k$ as $\mu_k \to \infty$, for $k = 1, 2, \cdots$.

3.5.6 Qualitative and Quantitative Analysis

In this section we perform rigorous qualitative and quantitative comparison between WSVT, SVT, and RPCA algorithms on three different video sequences: Stuttgart artificial video se-

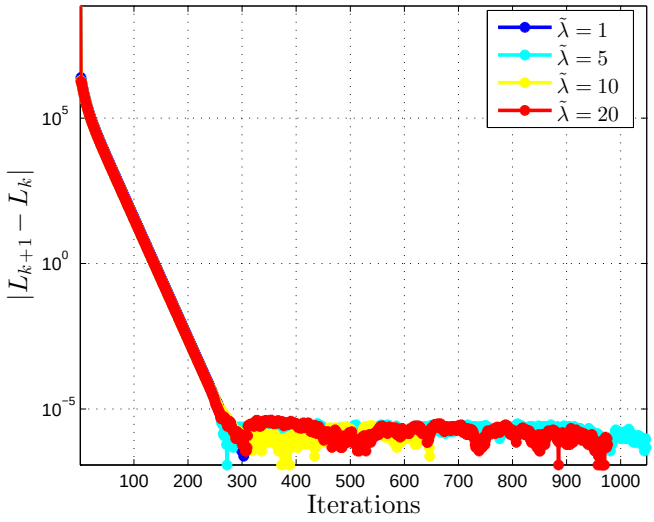

Figure 3.15: Iterations vs. $\mu_k|L_{k+1} - L_k|$ for $\tilde{\lambda} \in \{1, 5, 10, 20\}$.

quence, the airport sequence, and the fountain sequence. For the quantitative comparison between different methods, we only use Stuttgart artificial video sequence. We use two different metric for quantitative comparison: The receiver and operating characteristic (ROC) curve, and peak signal-to-noise ratio (PSNR). In Figure 3.16, we tested each method on 200 resized video frames. We employ the method defined in Section 3.5.3 to adaptively choose the weighted frame indexes for WSVT. Next, we test our method on the entire Stuttgart video sequence and compare its performance with the other unweighted low-rank methods. Unless specified, a weight $\tilde{\lambda} = 5$ is used to show the qualitative results for the WSVT algorithm in Figure 3.16 and 3.17. It is evident from Figure 3.16 that WSVT outperforms SVT and recovers the background as efficiently as RPCA methods. However, in Figure 3.17, WSVT shows superior performance over each method.

Next, in Figure 3.18 and 3.19, we perform the first set quantitative analysis of different methods. For quantitative analysis we use the following measure: Denote true positive rate (TPR) and false positive rate (FPR) as:

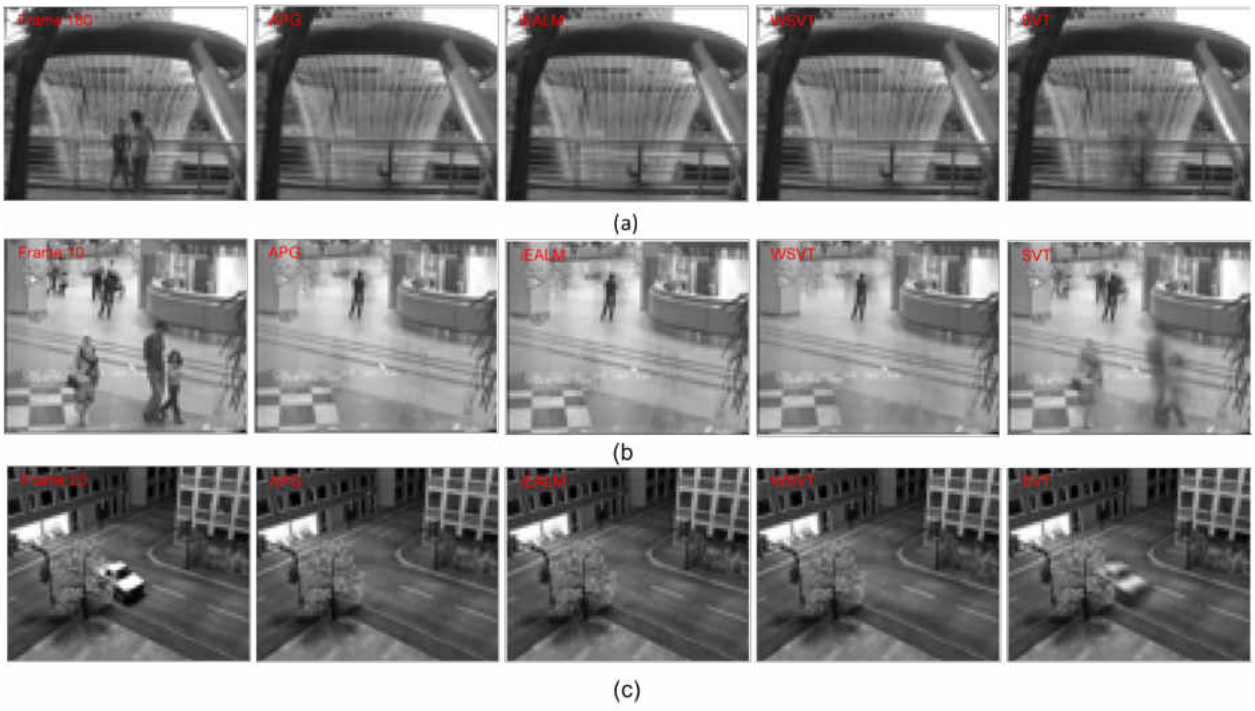

Figure 3.16: Qualitative analysis: From left to right: Original, APG low-rank, iEALM low-rank, WSVT low-rank, and SVT low-rank. Results on (from top to bottom): (a) Stuttgart video sequence, frame number 420 with dynamic foreground, methods were tested on last 200 frames; (b) airport sequence, frame number 10 with static and dynamic foreground, methods were tested on 200 frames; (c) fountain sequence, frame number 180 with static and dynamic foreground, methods were tested on 200 frames.

$$\text{TPR} = \frac{\text{correctly classified foreground pixels}}{\text{correctly classified foreground pixels} \ + \ \text{incorrectly rejected foreground pixels}}$$

and

$$\text{FPR} = \frac{\text{incorrectly classified foreground pixels}}{\text{incorrectly identified foreground pixels} \ + \ \text{correctly rejected foreground pixels}}.$$

Using the above relations we generate the receiver operating characteristic (ROC) curves for

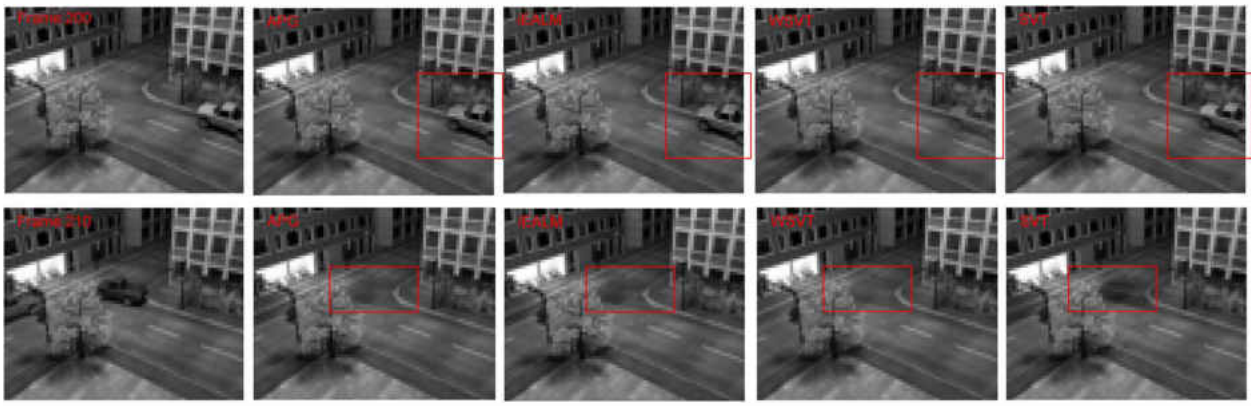

Figure 3.17: Qualitative analysis: From left to right: Original, APG low-rank, iEALM low-rank, WSVT low-rank, and SVT low-rank. (a) Stuttgart video sequence, frame number 600 with static foreground, methods were tested on last 200 frames; (b) Stuttgart video sequence, frame number 210 with dynamic foreground, methods were tested on 600 frames and WSVT provides the best low-rank background estimation.

different methods. A uniform threshold vector `linspace(0,255,100)` is used for plotting the receiver and operating characteristic (ROC) curves in Figure 3.18 and 3.19. From both ROC curves in Figures 3.18 and 3.19, the increments in performance of WSVT after using the weights seem to be trivial compared to the original SVT method, considering the computational complexity of proposed method is much higher according to Table 1. On the basis of the quantitative results performed using a uniform threshold vector in Figures 3.18 and 3.19, it supports the fact that WSVT performs better, albeit marginally. But the qualitative analysis results in Figures 3.16 and 3.17 show the performance of WSVT is superior to all state-of-the-art methods. We now provide a more detailed demonstration of the foreground objects recovered by different methods corresponding to the same video frames provided in Figures 3.20 and 3.21. We use color map for better comparison.

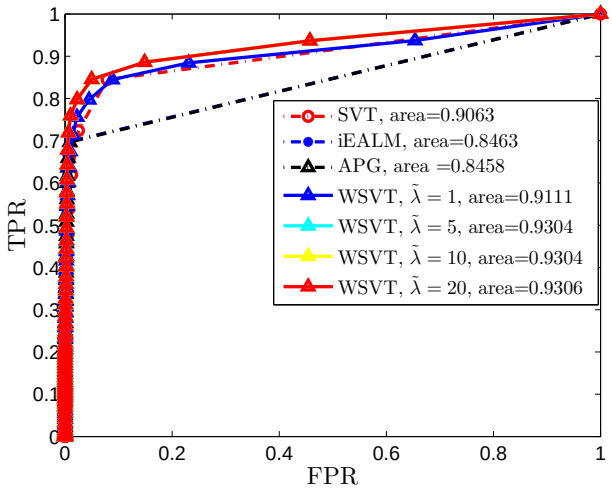

Figure 3.18: Quantitative analysis. ROC curve to compare between different methods on Stuttgart artificial sequence: 200 frames. For WSVT we choose $\tilde{\lambda} \in \{1, 5, 10, 20\}$. We see that for $W = I_n$, WSVT and SVT have the same quantitative performance, but indeed weight makes a difference in the performance of WSVT.

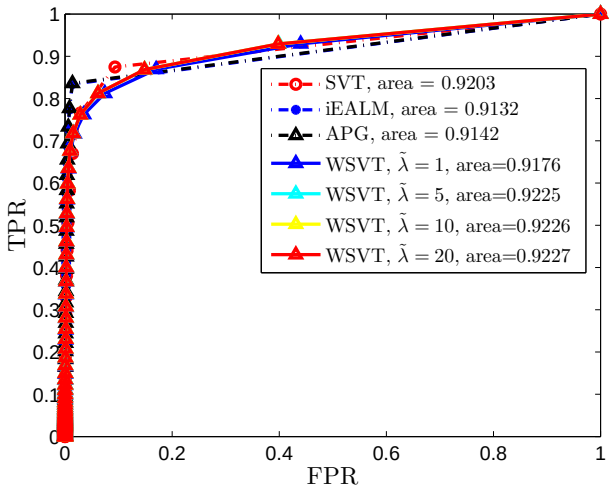

Figure 3.19: ROC curve to compare between the methods WSVT, SVT, iEALM, and APG on Stuttgart artificial sequence: 600 frames. For WSVT we choose $\tilde{\lambda} \in \{1, 5, 10, 20\}$.

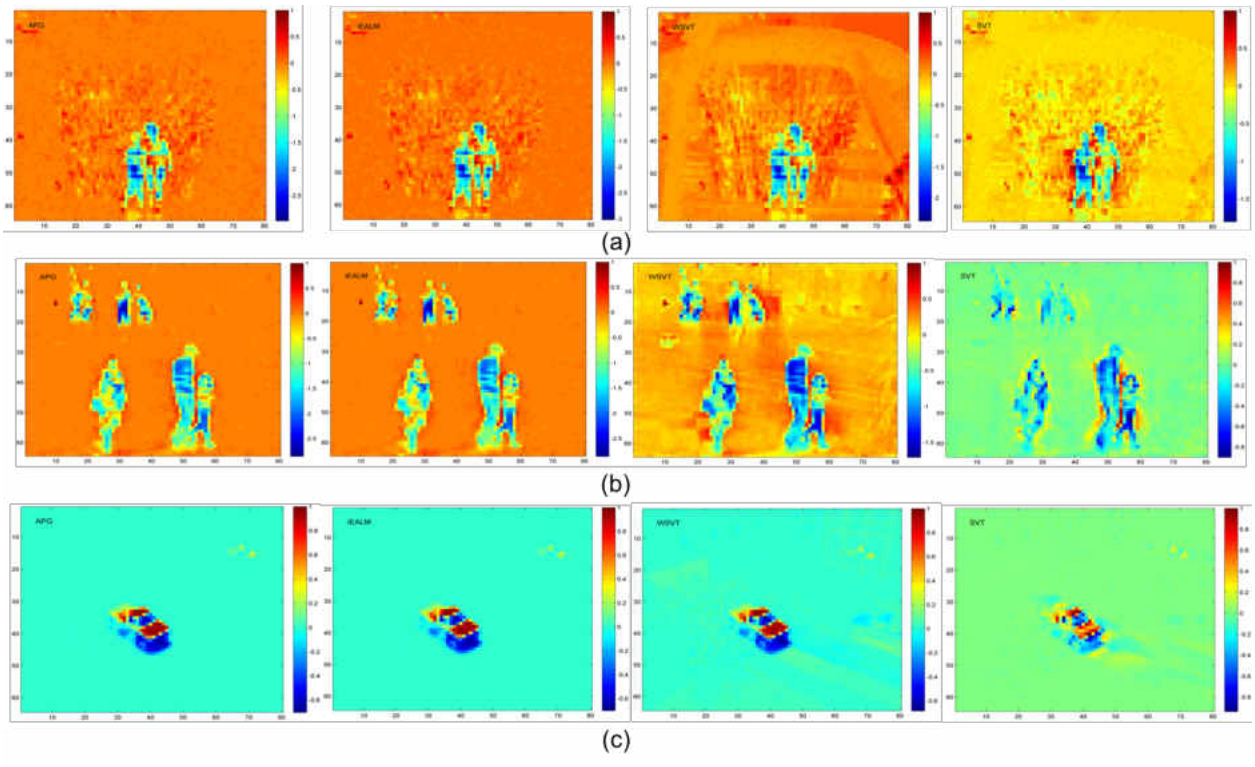

Figure 3.20: Foreground recovered by different methods: (a) fountain sequence, frame number 180 with static and dynamic foreground, (b) airport sequence, frame number 10 with static and dynamic foreground, (c) Stuttgart video sequence, frame number 420 with dynamic foreground.

From Figures 3.20 and 3.21 it is evident that in recovering the foreground objects, static or dynamic, WSVT outperforms other methods. A careful reader must also note that WSVT uniformly removes the noise (the changing light and illumination, and movement of the leaves of the tree for the Stuttgart sequence) from each video sequence.

Inspired by the empirical results in Figures 3.20 and 3.21, we propose a nonuniform threshold vector to plot the ROC curves and compare between the methods using the same metric. In Figures 3.22 and 3.23, we provide quantitative comparisons between the methods using a new non-uniform threshold vector `[0,15,20,25,30,31:2.5:255]`. This way we can reduce the number of false negatives and increase the number of true positives detected by

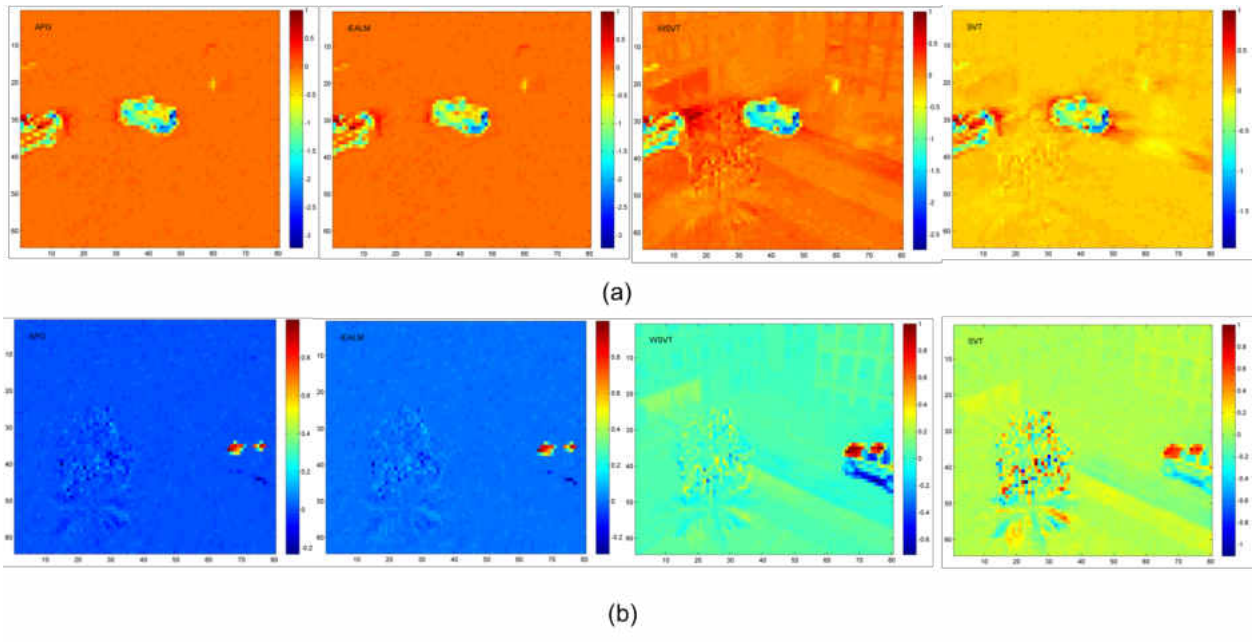

Figure 3.21: Foreground recovered by different methods for Stuttgart sequence: (a) frame number 210 with dynamic foreground, (b) frame number 600 with static foreground.

WSVT as it appears in Figure 3.16, 3.17, 3.20 and 3.21. To conclude, WSVT has better quantitative and qualitative results when there is a static foreground in the video sequence.

Next we will provide another quantitative comparison of different methods. For this purpose we will use peak signal to noise ratio (PSNR). PSNR is defined as $10log_{10}$ of the ratio of the peak signal energy to the mean square error (MSE) observed between the processed video signal and the original video signal.

If $E(:,i)$ denotes each reconstructed vectorized foreground frame in the video sequence and $G(:,i)$ be the corresponding ground truth frame, then PSNR is defined as $10log_{10}\frac{\mathrm{M}_I^2}{\mathrm{MSE}}$, where $\mathrm{MSE} = \frac{1}{mn}\|E(:,i)-G(:,i)\|_2^2$ and M_I is the maximum possible pixel value of the image. In our case the pixels are represented using 8 bits per sample, and therefore, M_I is 255. The proposal is that the higher the PSNR, the better degraded image has been reconstructed to match the original image and the better the reconstructive algorithm. This would occur

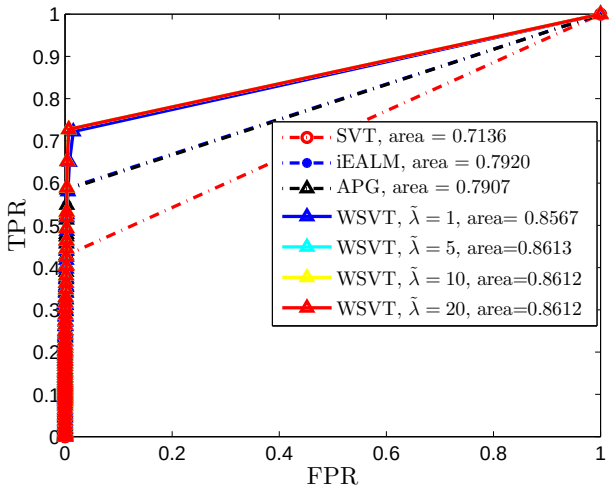

Figure 3.22: Quantitative analysis. ROC curve to compare between the methods WSVT, SVT, iEALM, and APG : 200 frames. For WSVT we choose $\tilde{\lambda} \in \{1, 5, 10, 20\}$. The performance gain by WSVT compare to iEALM, APG, and SVT are: 8.92%, 8.74%, and 20.68% respectively on 200 frames (with static foreground)

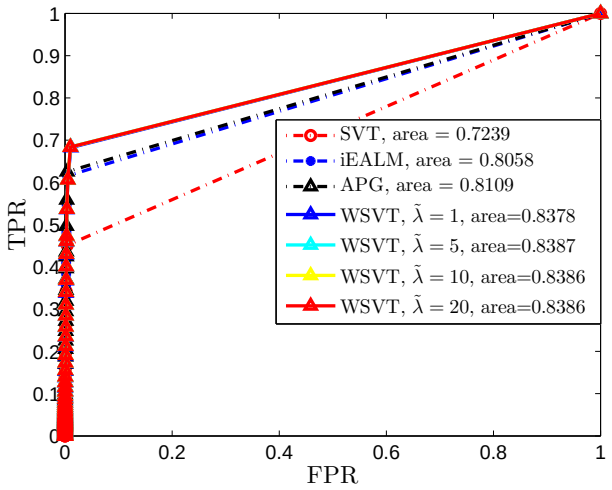

Figure 3.23: Quantitative analysis. ROC curve to compare between the methods WSVT, SVT, iEALM, and APG : 600 frames. For WSVT we choose $\tilde{\lambda} \in \{1, 5, 10, 20\}$. The performance gain by WSVT compare to iEALM, APG, and SVT are 4.07%, 3.42%, and 15.85% respectively on 600 frames.

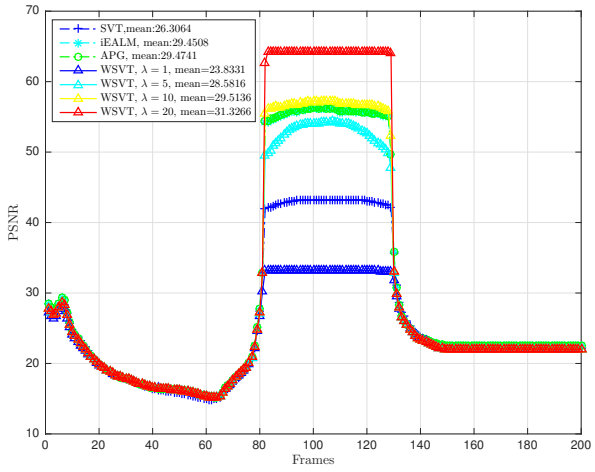

Figure 3.24: PSNR of each video frame for WSVT, SVT, iEALM, and APG. The methods were tested on last 200 frames of the Stuttgart data set. For WSVT we choose $\tilde{\lambda} \in \{1, 5, 10, 20\}$.

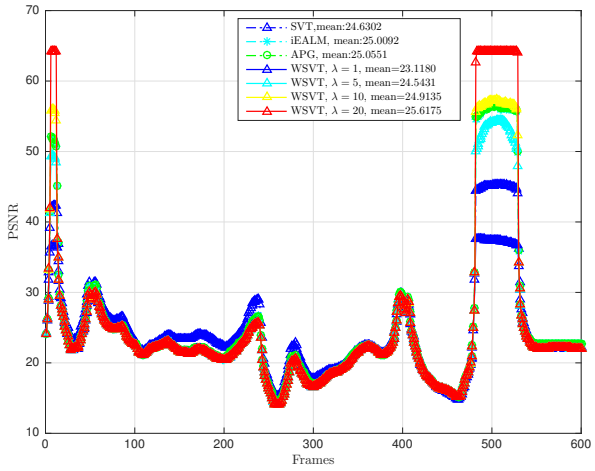

Figure 3.25: PSNR of each video frame for WSVT, SVT, iEALM, and APG when methods were tested on the entire sequence. For WSVT we choose $\tilde{\lambda} \in \{1, 5, 10, 20\}$. WSVT has increased PSNR when a weight is introduced corresponding to the frames with least foreground movement.

Table 3.1: Average computation time (in seconds) for each algorithm in background estimation

No. of frames	iEALM	APG	SVT	WSVT
200	4.994787	14.455450	0.085675	1.4468
600	131.758145	76.391438	0.307442	8.7885334

because we wish to minimize the MSE between images with respect to the maximum signal value of the image. For a reconstructed image with 8 bits bit depth, the PSNR are between 30 and 50 dB, where the higher is the better.

In Figures 3.24 and 3.25, we demonstrate the PSNR and mean PSNR of different methods on the Stuttgart sequence. For their implementation, first we calculate the PSNR of the last 200 frames of the sequence containing the static foreground, and finally we use 600 frames of the video sequence. It is evident from Figures 3.24 and 3.25 that weight improves the PSNR of WSVT significantly over the other existing methods. More specifically, we see that the weighted background frames or the frames with least foreground movement has higher PSNR than all other models traditionally used for background estimation. In Figures 3.24 and 3.25, for $\lambda = 1$, PSNR of the frames with least foreground movement is a little higher than 30 dB, but for $\lambda = 10$ and 20 they are about 55 dB and 65 dB respectively.

3.5.7 Facial Shadow Removal: Using identity weight matrix

Removal of shadow and specularity from face images under varying illumination and camera position is a challenging problem in computer vision. In 2003, Basri and Jacobs showed the images of the same face exposed to a wide variety of lighting conditions can be approximated accurately by a low-dimensional linear subspace [53]. More specifically, the images under distant, isotropic lighting lie close to a 9-dimensional linear subspace which is known as

Table 3.2: Average computation time (in seconds) for each algorithm in shadow removal

No. of images	iEALM	APG	SVT	WSVT
65	1.601427	10.221226	0.039598	1.047922

harmonic plane.

For our experiment we use test images from the Extended Yale Face Database B [54].[1] The mechanism used to perform this experiments is fairly similar to the processing of the video frames. A set of training images of same person taken under varying illumination and camera position are first resized and vectorized to form the columns of the test matrix. We use different low-rank approximation algorithms on the test matrix to decompose it in the low-rank and error part. The low-rank component of the test-matrix is assumed to contain the face images without shadow and specularities. We choose 65 sample images and perform our experiments. The images are resized to [96,128], originally they were [480,640]. We set a uniform threshold 10^{-7} for each algorithm. For APG and iEALM, $\lambda = 1/\sqrt{\max\{m,n\}}$, and the parameters for iEALM are set to $\mu = 1.5, \rho = 1.25$ [9, 49]. For WSVT we choose $\tau = 500,\ \mu = 15,$ and $\rho = 3$ and the weight matrix is set to I_n. Since we have no access to the ground truth for this experiment we will only provide the qualitative result. Note that the rank of the low-dimensional linear model recovered by RPCA methods is 35, while WSVT and SVT are able to find a rank 4 subspace. Figure 3.26 and 3.27 show that WSVT outperforms SVT and RPCA algorithms. Since iEALM and APG has same reconstruction we only provide qualitative analysis for APG.

[1] see also, http://vision.ucsd.edu/content/extended-yale-face-database-b-b

Figure 3.26: Left to right: Original image (person B11, image 56, partially shadowed), low-rank approximation using APG, SVT, and WSVT. WSVT removes the shadows and specularities uniformly form the face image especially from the left half of the image.

Figure 3.27: Left to right: Original image (person B11, image 21, completely shadowed), low-rank approximation using APG, SVT, and WSVT. WSVT removes the shadows and specularities uniformly form the face image especially from the eyes, chin, and nasal region.

In both cases, WSVT removes the shadow and specularity uniformly from the face image and provides a superior qualitative result compare to SVT and RPCA algorithms.

CHAPTER FOUR: ON A PROBLEM OF WEIGHTED LOW RANK APPROXIMATION OF MATRICES

In image processing, rank-reduced signal processing, computer vision, and in many other engineering applications SVD is a successful designing tool. But SVD has limitations and in many applications, it may fail. Recall from Chapter 1, the solutions to (1.1) are given by

$$X^* = H_r(A) := U(A)\Sigma_r(A)V(A)^T, \tag{4.1}$$

where $A = U(A)\Sigma(A)V(A)^T$ is a SVD of A and $\Sigma_r(A)$ is the diagonal matrix obtained from $\Sigma(A)$ by thresholding: keeping only r largest singular values and replacing other singular values by 0 along the diagonal. This is also referred to as Eckart-Young-Mirsky's theorem ([38]) and is closely related to the PCA method in statistics [35]. Note that the solutions to (1.1) as given in (4.1) suffer from the fact that none of the entries of A is guaranteed to be preserved in X^*. In many applications this could be a typical weak point of SVD. For example if SVD is used in quadrantally-symmetric two-dimensional (2-D) filter design, as pointed out in ([37, 29, 30]), it might lead to a degraded construction in some cases as it is not able to discriminate between the important and unimportant components of A. So it is required to put more emphasis on some elements of the matrix A. In Chapter 3, we formulated and solved a weighted low-rank approximation problem that approximately preserve k columns of the data matrix when we put a large weight on them. But what about putting more emphasis on individual entries of a column, rather preserving an entire column? The method we defined in Chapter 3 is not able to answer this question.

In this chapter, we study a more general weighted low-rank approximation that is also inspired by the work of Golub, Hoffman, and Stewart (see Chapter 3). The problem we study in this chapter is more generalized in the sense is that we use a pointwise matrix multiplication with the weight matrix. This serves two purposes for us: one by using the pointwise weight we have the freedom to control the elements of the given matrix to be preserved in

the approximating low-rank matrix; and second, it helps us to show the convergence of our solution to that of Golub, Hoffman, and Stewart for the limiting case of weights.

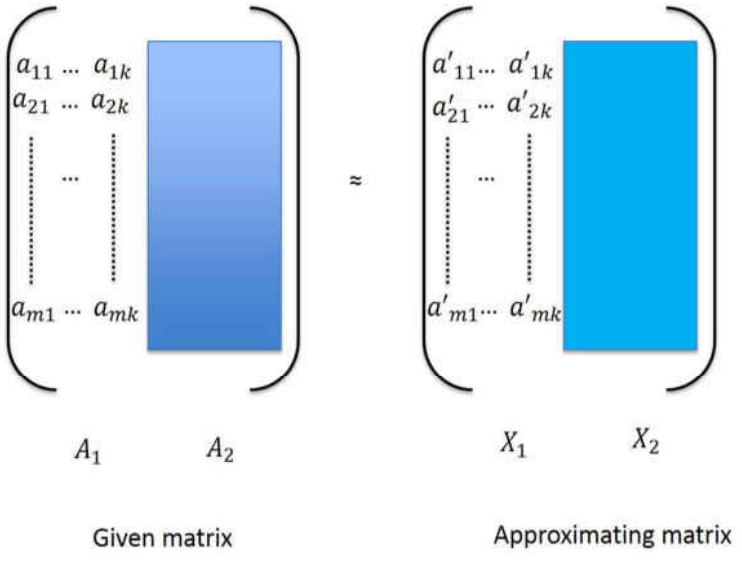

Figure 4.1: Pointwise multiplication with a weight matrix. Note that the elements in block A_1 can be controlled.

We also propose an algorithm based on the alternating direction method and demonstrate convergence asserted in our theorems.

4.1 Proof of Theorem 17

Recall that, in Chapter 3 we quote a theorem proposed by Golub, Hoffman, and Stewart [1]. We start by giving a detailed proof of the Theorem.

Proof. Without loss of generality let us assume $\mathrm{r}(A_1) = k$. If $\mathrm{r}(A_1) = l < k$ then A_1 can be replaced by a matrix with l linearly independent columns chosen from A_1 [1].The proof

is based on the QR decomposition of the matrix A. Let the QR decomposition of A be

$$A = (A_1 \;\; A_2) = QR \;\; = \;\; \begin{pmatrix} Q_1 & Q_2 & Q_3 \end{pmatrix} \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \\ 0 & 0 \end{pmatrix}, \tag{4.2}$$

where Q is an orthogonal matrix with blocks Q_1, Q_2 and Q_3 of size, $m \times k$, $m \times (n-k)$ and $m \times (m-n)$, respectively. Note that, if $m \geq n$ then the block Q_3 is considered to complete the entire space. The column vectors of Q_1 form an orthogonal basis for the column space of A_1, where those of Q_2 and Q_3 lie in the orthogonal complement of the column space of A_1. In other words, the column vectors of Q_2 and Q_3 form an orthonormal basis for the column space of A_2. The coefficient matrices, R_{11} and R_{22}, are square matrices of size $k \times k$ and $(n-k) \times (n-k)$ respectively, and they are upper triangular with R_{11} invertible (because A_1 has k linearly independent columns with $k \leq m$ and $\mathrm{r}(A_1) = k$ and hence R_{11} is full of rank and nonsingular). The other coefficient matrix R_{12} is of size $k \times (n-k)$. We can rewrite (4.2) as,

$$\begin{pmatrix} Q_1^T A_1 & Q_1^T A_2 \\ Q_2^T A_1 & Q_2^T A_2 \\ Q_3^T A_1 & Q_3^T A_2 \end{pmatrix} \;\; = \;\; \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \\ 0 & 0 \end{pmatrix}. \tag{4.3}$$

Write X as:

$$X \;\; = Q\hat{R} = (X_1 \;\; X_2) = \begin{pmatrix} Q_1 & Q_2 & Q_3 \end{pmatrix} \begin{pmatrix} R_{11} & \hat{R}_{12} \\ 0 & \hat{R}_{22} \\ 0 & \hat{R}_{32} \end{pmatrix}.$$

Using the unitary invariance of the Frobenius norm one can rewrite (3.1) as:

$$\begin{cases} \min\limits_{\hat{R}_{12}, \hat{R}_{22}, \hat{R}_{32}} \|R_{12} - \hat{R}_{12}\|_F^2 + \|R_{22} - \hat{R}_{22}\|_F^2 + \|R_{32} - \hat{R}_{32}\|_F^2, \\ \text{subject to } \; \mathrm{r}\left(\begin{pmatrix} R_{11} & \hat{R}_{12} \\ 0 & \hat{R}_{22} \\ 0 & \hat{R}_{32} \end{pmatrix} \right) \leq r. \end{cases} \tag{4.4}$$

Since R_{11} is nonsingular and of full rank, the choice of $\hat{R}_{12}$ does not change the rank of the matrix $\hat{R}$. The elementary column transformation on $\hat{R}$ can make $\hat{R}_{12}$ identically 0 without affecting the rank and changing $\hat{R}_{22}$ and $\hat{R}_{32}$ as well. [1] Therefore, one can choose $\hat{R}_{12} = R_{12}$, and (4.4) becomes

$$\begin{cases} \min\limits_{\hat{R}_{22},\hat{R}_{32}} \| \begin{pmatrix} R_{22} \\ 0 \end{pmatrix} - \begin{pmatrix} \hat{R}_{22} \\ \hat{R}_{32} \end{pmatrix} \|_F^2 \\ \text{such that } \mathrm{r}(\begin{pmatrix} \hat{R}_{22} \\ \hat{R}_{32} \end{pmatrix}) \leq r-k. \end{cases} \tag{4.5}$$

The above problem is equivalent to the classical PCA [1, 35, 38]. If R_{22} has a SVD $U\Sigma V^T$ then the matrix $\begin{pmatrix} R_{22} \\ 0 \end{pmatrix}$ has a SVD $\begin{pmatrix} U\Sigma V^T \\ 0 \end{pmatrix}$ as well. Hence, the solution to (4.5) is $H_{r-k}\begin{pmatrix} R_{22} \\ 0 \end{pmatrix} = \begin{pmatrix} H_{r-k}(R_{22}) \\ 0 \end{pmatrix}$. Therefore, $\begin{pmatrix} R_{11} & \hat{R}_{12} \\ 0 & \hat{R}_{22} \\ 0 & \hat{R}_{32} \end{pmatrix} = \begin{pmatrix} R_{11} & R_{12} \\ 0 & H_{r-k}(R_{22}) \\ 0 & 0 \end{pmatrix}$, and,

$$\begin{aligned} X &= Q\hat{R} = (A_1 \;\; X_2) = \begin{pmatrix} Q_1 & Q_2 & Q_3 \end{pmatrix} \begin{pmatrix} R_{11} & R_{12} \\ 0 & H_{r-k}(R_{22}) \\ 0 & 0 \end{pmatrix} \\ &= \begin{pmatrix} Q_1R_{11} & Q_1R_{12} + Q_2H_{r-k}(R_{22}) \end{pmatrix} \\ &= \begin{pmatrix} Q_1R_{11} & Q_1R_{12} + H_{r-k}(Q_2R_{22}) \end{pmatrix}. \end{aligned}$$

The last equality is due to the fact that R_{22} and Q_2R_{22} has the same SVD and can be shown using the following argument: Let $R_{22} = U\Sigma V^T$ be a SVD of R_{22}. So, $Q_2R_{22} = Q_2U\Sigma V^T = U_1\Sigma V^T$, where $U_1 = Q_2U$ is a column orthogonal matrix and it implies, $Q_2H_{r-k}(R_{22}) =$

[1] One such elementary column transformation is post multiplying $\hat{R}$ by $\begin{pmatrix} I_{k\times k} & -R_{11}^{-1}\hat{R_{12}}_{k\times n-k} \\ 0_{n-k\times n-k} & I_{n-k\times n-k} \end{pmatrix}$ which shows $\hat{R_{22}}$ can be eliminated by only using R_{11}.

$H_{r-k}(Q_2R_{22})$. Using (4.3) we can write,

$$\begin{aligned}(A_1 \mid X_2) &= (Q_1R_{11} \;\; Q_1R_{12} + H_{r-k}(Q_2R_{22}) \\ &= (Q_1Q_1^TA_1 \;\; Q_1Q_1^TA_2 + H_{r-k}(Q_2Q_2^TA_2)) \\ &= (Q_1Q_1^TA_1 \;\; P_{A_1}(A_2) + H_{r-k}\left(P_{A_1}^{\perp}(A_2)\right)) \\ &= (A_1 \;\; P_{A_1}(A_2) + H_{r-k}\left(P_{A_1}^{\perp}(A_2)\right)).\end{aligned}$$

Therefore, $\tilde{A}_2 = P_{A_1}(A_2) + H_{r-k}\left(P_{A_1}^{\perp}(A_2)\right)$. This completes the proof. □

Remark 23. *According to Section 3 of [1], the matrix $\tilde{A}_2$ is unique if and only if $H_{r-k}\left(P_{A_1}^{\perp}(A_2)\right)$ is unique, which means the $(r-k)$th singular value of $P_{A_1}^{\perp}(A_2)$ is strictly greater than $(r-k+1)$th singular value. When $\tilde{A}_2$ is not unique, the formula for $\tilde{A}_2$ given in Theorem 20 should be understood as the membership of the set specified by the right-hand side of (3.2). We will use this convention in this paper.*

In this chapter, we consider the following problem by using a more general point-wise multiplication with a weight matrix W of non-negative terms: given $A = (A_1 \;\; A_2) \in \mathbb{R}^{m\times n}$ with $A_1 \in \mathbb{R}^{m\times k}$ and $A_2 \in \mathbb{R}^{m\times(n-k)}$, and a weight matrix $W = (W_1 \;\; W_2) \in \mathbb{R}^{m\times n}$ of compatible block partition solve:

$$\min_{\substack{X_1,X_2 \\ \mathrm{r}(X_1 \;\; X_2)\le r}} \| ((A_1 \;\; A_2) - (X_1 \;\; X_2)) \odot (W_1 \; W_2)\|_F^2. \tag{4.6}$$

This is the weighted low-rank approximation problem studied first when W is an indicator weight for dealing with the missing data case ([40, 41]) and then for more general weight in machine learning, collaborative filtering, 2-D filter design, and computer vision [39, 43, 45, 37, 29, 30]. One can consider (4.6) as a special case of the weighted low-rank approximation problem (1.5) defined in [37]:

$$\min_{X\in\mathbb{R}^{m\times n}} \|A - X\|_Q^2, \;\; \text{subject to } \mathrm{r}(X) \le r,$$

where $Q \in \mathbb{R}^{mn\times mn}$ is a symmetric positive definite weight matrix. Denote $\|A-X\|_Q^2 := \mathrm{vec}(A-X)^TQ\mathrm{vec}(A-X)$, where $\mathrm{vec}(\cdot)$ is an operator which maps the entries of $\mathbb{R}^{m\times n}$ to

$\mathbb{R}^{mn\times 1}$. Unlike problem (3.4) the weighted low-rank approximation problem (4.6) has no closed form solution in general [39, 37]. Also, note that the entry-wise multiplication is not associative with the regular matrix multiplication: $(A \cdot B) \odot C \neq A \cdot (B \odot C)$, and as a consequence, we lose the unitary invariance property in case of using the Frobenius norm. We are interested in finding out the limit behavior of the solutions to problem (4.6) when $(W_1)_{ij} \to \infty$ and $W_2 = \mathbb{1}$, a matrix whose entries are equal to 1. One can expect that with appropriate conditions, the solutions to (4.6) will converge and the limit is A_G. We will verify this with an estimate on the rate of convergence. We will also extend the convergence result to the unconstrained version of the problem (4.6) and propose a numerical algorithm to solve (4.6) for the special case of the weight matrix $(W_1)_{ij} \to \infty$ and $W_2 = \mathbb{1}$.

The rest of the chapter is organized as follows. In Section 4.2, we state our main results. Their proofs will be given in Section 4.3. In Section 4.4, we will propose a numerical algorithm to solve problem (4.6) for a special choice of weights and present the convergence of our proposed algorithm. Numerical results verifying our main results are given in section 4.5.

4.2 Main Results

We will start with a simple example. The example will support the fact why SVD can not be used to find a solution to the problem (4.6). Next, we will present our main analytical results.

Example 24. *Let* $A = \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix}$ *with* $\sigma_1 > \sigma_2 > 0$ *and let* $W = \begin{pmatrix} 1 & 0 \\ 0 & w_2 \end{pmatrix}, w_2 > 0$. *Solve:*

$$\min_{\mathrm{r}(\hat{X})\leq 1} \|(A - \hat{X}) \odot W\|_F^2. \tag{4.7}$$

Writing $\hat{X} = \begin{pmatrix} a \\ b \end{pmatrix} \begin{pmatrix} c & d \end{pmatrix}$, we solve

$$\min_{a,b,c,d} \| \left(\begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix} - \begin{pmatrix} a \\ b \end{pmatrix} \begin{pmatrix} c & d \end{pmatrix} \right) \odot \begin{pmatrix} 1 & 0 \\ 0 & w_2 \end{pmatrix} \|_F^2$$
$$= \min_{a,b,c,d} \left((\sigma_1 - ac)^2 + (\sigma_2 - bd)^2 w_2^2 \right).$$

There are two critical points with critical values

$$\sigma_1^2 \text{ and } \sigma_2^2 w_2^2$$

which, when $w_2^2 > \frac{\sigma_1^2}{\sigma_2^2}$, yields a solution given by

$$\begin{pmatrix} 0 & 0 \\ 0 & \sigma_2 \end{pmatrix}$$

other than

$$\begin{pmatrix} \sigma_1 & 0 \\ 0 & 0 \end{pmatrix}$$

as expected from the SVD method.

Let $(\tilde{X}_1(W), \tilde{X}_2(W))$ be a solution to (4.6). Denote $\mathcal{A} = P_{A_1}^{\perp}(A_2)$ and $\tilde{\mathcal{A}} = P_{\tilde{X}_1(W)}^{\perp}(A_2)$. Also denote $s = \mathrm{r}(\mathcal{A})$ and let the ordered non-zero singular values of $\mathcal{A}$ be $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_s > 0$. Let $\lambda_j = \min_{1 \leq i \leq m} (W_1)_{ij}$ and $\lambda = \min_{1 \leq j \leq k} \lambda_j$.

Theorem 25. *Let $W_2 = \mathbb{1}_{m \times (n-k)}$. If $\sigma_{r-k} > \sigma_{r-k+1}$, then*

$$(\tilde{X}_1(W) \;\; \tilde{X}_2(W)) = A_G + O(\frac{1}{\lambda}), \; \lambda \to \infty,$$

where $A_G = (A_1 \; \tilde{A}_2)$ is defined to be the unique solution to (3.1).

Remark 26. *1. The assertion of the uniqueness of A_G is due to the assumption $\sigma_{r-k} > \sigma_{r-k+1}$ (see the Remark 23).*

2. *As in ([26]), with proper condition one can find* $(\tilde{X}_1(W) \;\; \tilde{X}_2(W)) \to A_G$ *as* $(W_1)_{ij} \to \infty$ *and* $W_2 = \mathbb{1}$. *We should mention, however, it does not give the convergence rate as proposed in Theorem 25.*

Theorem 27. *Assume* $r > k$. *For* $(W_1)_{ij} > 0$, *if* $(\tilde{X}_1(W), \tilde{X}_2(W))$ *is a solution to (4.6), then*

$$\tilde{X}_2(W) = P_{\tilde{X}_1(W)}(A_2) + H_{r-k}\left(P^{\perp}_{\tilde{X}_1(W)}(A_2)\right).$$

Next, if we do not know r but still want to reduce the rank in our approximation, consider the unconstrained version of (4.6): for $\tau > 0$,

$$\min_{X_1,X_2} \left\{ \| \left((A_1 \;\; A_2) - (X_1 \;\; X_2)\right) \odot (W_1 \;\; W_2) \|_F^2 + \tau \mathrm{r}(X_1 \;\; X_2) \right\}. \tag{4.8}$$

Note that problem (3.7) in Chapter 3 is a special case of problem (4.8), where the ordinary matrix multiplication is used with the nonsingular weight matrix $W \in \mathbb{R}^{n\times n}$ and $\mathrm{r}(X_1 \;\; X_2)$ is replaced by its convex function the nuclear norm X. We can establish our claim of (3.7) to be a special case of (4.8) by using the following argument: Note that, replacing $\mathrm{r}(X_1 \;\; X_2)$ by $\|X\|_*$ in problem (4.8) we have:

$$\min_{X_1,X_2} \left\{ \| \left((A_1 \;\; A_2) - (X_1 \;\; X_2)\right) \odot (W_1 \;\; W_2) \|_F^2 + \tau \|X\|_* \right\}. \tag{4.9}$$

Write W in its SVD form $W = U\Sigma V^T$, where $U, V \in \mathbb{R}^{n\times n}$ are unitary matrices and $\Sigma = \mathrm{diag}(\sigma_1, \;\sigma_2, \cdots, \sigma_n)$ is a full rank diagonal matrix. Therefore using the unitary invariance of the matrix norms (3.7) can be written as

$$\begin{aligned}
\min_X \{ \|(A - X)U\Sigma V^T\|_F^2 + \tau \|X\|_* \} &= \min_X \{ \|(AU - XU)\Sigma V^T\|_F^2 + \tau \|XU\|_* \} \\
&= \min_X \{ \|(AU - XU)\Sigma\|_F^2 + \tau \|XU\|_* \} \\
&= \min_{\substack{\tilde{X} \\ \tilde{X} = XU}} \{ \|(AU - \tilde{X}) \odot W_\Sigma\|_F^2 + \tau \|\tilde{X}\|_* \},
\end{aligned}$$

where $W_\Sigma = \left(\sigma_1 \mathbb{1} \;\; \sigma_2 \mathbb{1} \cdots \sigma_n \mathbb{1}\right) \in \mathbb{R}^{m\times n}$, and $\mathbb{1} \in \mathbb{R}^m$, is a vector whose entries are all 1. Thus (3.7) is in the form of (4.9) with data matrix AU and hence it is a special form of (4.8).

Again one can expect that the solutions to (4.8) will converge to A_G as $(W_1)_{ij} \to \infty$ and $(W_2)_{ij} \to 1$. Define $\mathcal{A}_G^r,\ 0 \le r \le \min\{m,n\}$, to be the set of all solutions to (3.1). Let $(\hat{X}_1(W)\ \ \hat{X}_2(W))$ be a solution to (4.8). With the notations above we will present the next two theorems.

Theorem 28. *Every accumulation point of* $(\hat{X}_1(W)\ \ \hat{X}_2(W))$ *as* $(W_1)_{ij} \to \infty, (W_2)_{ij} \to 1$ *belongs to* $\bigcup\limits_{0\le r\le \min\{m,n\}} \mathcal{A}_G^r$.

Theorem 29. *Assume that* $\sigma_1 > \sigma_2 > \cdots > \sigma_s > 0$. *Denote* $\sigma_0 := \infty$ *and* $\sigma_{s+1} := 0$. *Then the accumulation point of the sequence* $(\hat{X}_1(W)\ \hat{X}_2(W))$, *as* $(W_1)_{ij} \to \infty$ *and* $(W_2)_{ij} \to 1$ *is unique; and this unique accumulation point is given by*

$$\begin{pmatrix} A_1 & P_{A_1}(A_2) + H_{r^*}\left(P_{A_1}^{\perp}(A_2)\right) \end{pmatrix}$$

with r^* *satisfying*

$$\sigma_{r^*+1}^2 \le \tau < \sigma_{r^*}^2.$$

Remark 30. *For the case when* $P_{A_1}^{\perp}(A_2)$ *has repeated singular values, we leave it to the reader to verify the following more general statement by using a similar argument: Let* $\hat{\sigma}_1 > \hat{\sigma}_2 > ... > \hat{\sigma}_t > 0$ *be the singular values of* $P_{A_1}^{\perp}(A_2)$ *with multiplicity* $k_1, k_2, \cdots k_t$ *respectively. Note that* $\sum_{i=1}^t k_i = s$. *Let* $\sigma_{p^*+1}^2 \le \tau < \sigma_{p^*}^2$, *where* σ_{p^*} *has multiplicity* k_{p^*}. *Then the accumulation points of the set* $(\hat{X}_1(W), \hat{X}_2(W))$, *as* $(W_1)_{ij} \to \infty, (W_2)_{ij} \to 1$, *belongs to the set* $\bigcup\limits_{r^*} \mathcal{A}_G^{r^*}$, *where* $1 + \sum_{i=1}^{p^*-1} k_i \le r^* < \sum_{i=1}^{p^*} k_i$.

4.3 Proofs

To prove Theorem 25, we first establish the following lemmas.

Lemma 31. *As* $(W_1)_{ij} \to \infty$ *and* $W_2 = \mathbb{1}$, *we have the following estimates.*

(i) $\tilde{X}_1(W) = A_1 + O(\frac{1}{\lambda})$.

(ii) $P_{\tilde{X}_1(W)}(A_2) = P_{A_1}(A_2) + O(\frac{1}{\lambda})$.

(iii) $P^{\perp}_{\tilde{X}_1(W)}(A_2) = P^{\perp}_{A_1}(A_2) + O(\frac{1}{\lambda})$.

PROOF: *(i).* Note that,

$$\begin{aligned}
&\|(A_1 - \tilde{X}_1(W)) \odot W_1\|_F^2 + \|A_2 - \tilde{X}_2(W)\|_F^2 \\
&= \min_{\substack{X_1, X_2 \\ \mathrm{r}(X_1 \;\; X_2) \le r}} \left(\|(A_1 - X_1) \odot W_1\|_F^2 + \|A_2 - X_2\|_F^2 \right) \\
&\le \|A_2\|_F^2 \text{ (by taking } (X_1 \;\; X_2) = (A_1 \;\; 0)) \\
&= m_1 \text{ (say).}
\end{aligned}$$

Then $\sum_{\substack{1 \le i \le m \\ 1 \le j \le k}} ((A_1)_{ij} - (\tilde{X}_1(W))_{ij})^2 (W_1)_{ij}^2 \le m_1$ and so

$$|(A_1)_{ij} - (\tilde{X}_1(W))_{ij}| \le \frac{\sqrt{m_1}}{(W_1)_{ij}}; \quad 1 \le i \le m, 1 \le j \le k.$$

Thus

$$\tilde{X}_1(W) = A_1 + O(\frac{1}{\lambda}) \text{ as } \lambda \to \infty.$$

(ii). For simplicity, let us assume $\mathrm{r}(A_1) = k$, full rank. If $\mathrm{r}(A_1) = l < k$, then A_1 can be replaced by a matrix with l linearly independent columns chosen from A_1 [1]. We use the QR decomposition of $A = (A_1 \quad A_2)$. Let

$$(A_1 \quad A_2) = QR = (Q_1 \;\; Q_2 \;\; Q_3) \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \\ 0 & 0 \end{pmatrix},$$

where $Q \in \mathbb{R}^{m \times m}$ is an orthogonal matrix with block matrices Q_1, Q_2, and Q_3 of sizes $m \times k$, $m \times (n-k)$, and $m \times (m-n)$, respectively, and the matrices R_{11} and R_{22} are both upper triangular. Therefore,

$$\begin{cases} A_1 = Q_1 R_{11}, \\ A_2 = Q_1 R_{12} + Q_2 R_{22}. \end{cases} \tag{4.10}$$

Note that $Q_1R_{12} = P_{A_1}(A_2)$ and $Q_2R_{22} = P_{A_1}^{\perp}(A_2)$. By (i), we see that $\mathrm{r}(\tilde{X}_1(W)) = k$, for all large $(W_1)_{ij}$. We now look at the QR decomposition of $\tilde{X}_1(W)$:

$$\tilde{X}_1(W) = Q_1(W)R_{11}(W), \tag{4.11}$$

where $Q_1(W)$ is column orthogonal ($Q_1^T(W)Q_1(W) = I_k$), and $R_{11}(W)$ is upper triangular. The QR decomposition can be obtained via the Gram-Schmidt process. If we write the matrices as collection of column vectors:

$$\tilde{X}_1(W) = (x_1(W)\ x_2(W)\cdots x_k(W)),\ \ Q_1(W) = (q_1(W)\ q_2(W)\cdots q_k(W)),$$

and

$$A_1 = (a_1\ a_2\cdots a_k),\ \ Q_1 = (q_1\ q_2\cdots q_k),$$

where $x_i(W), q_i(W), a_i, q_i \in \mathbb{R}^m,\ i = 1, 2, \cdots k$, then by (i),

$$x_i(W) = a_i + O(\frac{1}{\lambda_i}),\ \lambda_i \to \infty. \tag{4.12}$$

Next, for each $i = 1, 2, \cdots, k$ we can show (where $\|\cdot\|_2$ denotes the ℓ_2 norm of vectors)

$$\begin{aligned}
\|x_i(W)\|_2 =& \sqrt{\sum_{j=1}^{m}(a_{ji} + O(\frac{1}{\lambda_{ji}}))^2)} \\
=& \sqrt{\sum_{j=1}^{m} a_{ji}^2 + 2\sum_{j=1}^{m} a_{ji}O(\frac{1}{\lambda_{ji}}) + \sum_{j=1}^{m}(O(\frac{1}{\lambda_{ji}}))^2} \\
=& \sqrt{(\sum_{j=1}^{m} a_{ji}^2)}\sqrt{1 + \frac{2}{\sum_{j=1}^{m} a_{ji}^2}\sum_{j=1}^{m} a_{ji}O(\frac{1}{\lambda_{ji}}) + \frac{1}{\sum_{j=1}^{m} a_{ji}^2}\sum_{j=1}^{m}(O(\frac{1}{\lambda_{ji}}))^2} \\
=& \|a_i\|_2\sqrt{1 + \frac{2}{\|a_i\|_2}\sum_{j=1}^{m} a_{ji}O(\frac{1}{\lambda_{ji}}) + \frac{1}{\|a_i\|_2}\sum_{j=1}^{m}(O(\frac{1}{\lambda_{ji}}))^2},
\end{aligned}$$

which together with the conditions: (i) $\min_{1\le j\le m}\lambda_{ji} > 1$, and (ii) $|\frac{2}{\|a_i\|_2}\sum_{j=1}^{m} a_{ji}O(\frac{1}{\lambda_{ji}}) + \frac{1}{\|a_i\|_2}\sum_{j=1}^{m}(O(\frac{1}{\lambda_{ji}}))^2| < 1$ gives

$$\|x_i(W)\|_2 \approx \|a_i\|_2(1 + \frac{1}{2}(\frac{2}{\|a_i\|_2}\sum_{j=1}^{m} a_{ji}O(\frac{1}{\lambda_{ji}}) + \frac{1}{\|a_i\|_2}\sum_{j=1}^{m}(O(\frac{1}{\lambda_{ji}}))^2)).$$

Therefore,

$$\|x_i(W)\|_2 \approx \|a_i\|_2 + O(\frac{1}{\min_{1\le j\le m}\lambda_{ji}}). \tag{4.13}$$

For each $i = 1, 2, \cdots, k$, using the same arguments as above, from (4.13) we can show

$$\begin{aligned}\frac{1}{\|x_i(W)\|_2} =&(\|a_i\|_2 + O(\frac{1}{\min_{1\le j\le m}\lambda_{ji}}))^{-1}\\ =&\frac{1}{\|a_i\|_2}(1+\frac{1}{\|a_i\|_2}O(\frac{1}{\min_{1\le j\le m}\lambda_{ji}}))^{-1}\\ =&\frac{1}{\|a_i\|_2}(1-\frac{1}{\|a_i\|_2}O(\frac{1}{\min_{1\le j\le m}\lambda_{ji}})).\end{aligned}$$

Finally for each $i = 1, 2, \cdots, k$, we find

$$\frac{x_i(W)}{\|x_i(W)\|_2} = \quad (a_i + O(\tfrac{1}{\lambda_i}))\tfrac{1}{\|a_i\|_2}(1-\tfrac{1}{\|a_i\|_2}O(\tfrac{1}{\lambda_i})) = \tfrac{a_i}{\|a_i\|_2} + O(\tfrac{1}{\lambda_i}). \tag{4.14}$$

In particular, as $\lambda_1 \to \infty$,

$$q_1(W) = \frac{x_1(W)}{\|x_1(W)\|_2} = \frac{a_1 + O(\frac{1}{\lambda_1})}{\|a_1 + O(\frac{1}{\lambda_1})\|_2} = \frac{a_1}{\|a_1\|_2} + O(\frac{1}{\lambda_1}) = q_1 + O(\frac{1}{\lambda_1}).$$

Similarly, we see that

$$\langle x_2(W), q_1(W)\rangle = \langle a_2, q_1\rangle + O\left(\frac{1}{\min\{\lambda_1,\lambda_2\}}\right),\ \ \min\{\lambda_1,\lambda_2\}\to\infty,$$

and

$$\begin{aligned}&x_2(W) - \langle x_2(W), q_1(W)\rangle q_1(W)\\ &= a_2 + O(\frac{1}{\lambda_2}) - \langle a_2 + O(\frac{1}{\lambda_2}), q_1 + O(\frac{1}{\lambda_1})\rangle(q_1 + O(\frac{1}{\lambda_1}))\\ &= a_2 - \langle a_2, q_1\rangle q_1 + O\left(\frac{1}{\min\{\lambda_1,\lambda_2\}}\right),\ \ \min\{\lambda_1,\lambda_2\}\to\infty.\end{aligned}$$

Therefore,

$$\begin{aligned}q_2(W) &= \frac{x_2(W) - \langle x_2(W), q_1(W)\rangle q_1(W)}{\|x_2(W) - \langle x_2(W), q_1(W)\rangle q_1(W)\|_2}\\ &= \frac{a_2 - \langle a_2, q_1\rangle q_1 + O\left(\frac{1}{\min\{\lambda_1,\lambda_2\}}\right)}{\|a_2 - \langle a_2, q_1\rangle q_1 + O\left(\frac{1}{\min\{\lambda_1,\lambda_2\}}\right)\|_2},\end{aligned}$$

which using the same idea as in (4.14) and considering $e_1 = a_2 - \langle a_2, q_1\rangle q_1$ reduces to

$$\begin{aligned} q_2(W) &= \frac{e_1}{\|e_1\|_2(1 + \frac{1}{\|e_1\|_2} O\left(\frac{1}{\min\{\lambda_1,\lambda_2\}}\right))} \\ &= \frac{e_1}{\|e_1\|_2}(1 - \frac{1}{\|e_1\|_2} O\left(\frac{1}{\min\{\lambda_1, \lambda_2\}}\right)). \end{aligned} \tag{4.15}$$

As $q_2 = \frac{e_1}{\|e_1\|_2}$, (4.15) leads us to

$$q_2(W) = q_2 + O\left(\frac{1}{\min\{\lambda_1, \lambda_2\}}\right), \ \min\{\lambda_1, \lambda_2\} \to \infty.$$

Continuing this process we obtain, as $\lambda \to \infty$,

$$Q_1(W) = (q_1 \ \ q_2 \cdots q_k) + O\left(\frac{1}{\min\{\lambda_1, \cdots, \lambda_k\}}\right) = Q_1 + O(\frac{1}{\lambda}).$$

Finally, we have

$$\begin{aligned} P_{\tilde{X}_1(W)}(A_2) &= Q_1(W)Q_1(W)^T A_2 \\ &= \left(Q_1 + O(\frac{1}{\lambda})\right)\left(Q_1 + O(\frac{1}{\lambda})\right)^T A_2 \\ &= P_{A_1}(A_2) + O(\frac{1}{\lambda}), \end{aligned}$$

as $\lambda \to \infty$.

(iii) We know that

$$P_{\tilde{X}_1(W)}(A_2) + P^{\perp}_{\tilde{X}_1(W)}(A_2) = A_2 = P_{A_1}(A_2) + P^{\perp}_{A_1}(A_2).$$

Using (ii)

$$P_{A_1}(A_2) + O(\frac{1}{\lambda}) + P^{\perp}_{\tilde{X}_1(W)}(A_2) = P_{A_1}(A_2) + P^{\perp}_{A_1}(A_2), \ \lambda \to \infty.$$

Therefore,

$$P^{\perp}_{\tilde{X}_1(W)}(A_2) = P^{\perp}_{A_1}(A_2) + O(\tfrac{1}{\lambda}), \ \lambda \to \infty. \tag{4.16}$$

This completes the proof of Lemma 31. ■

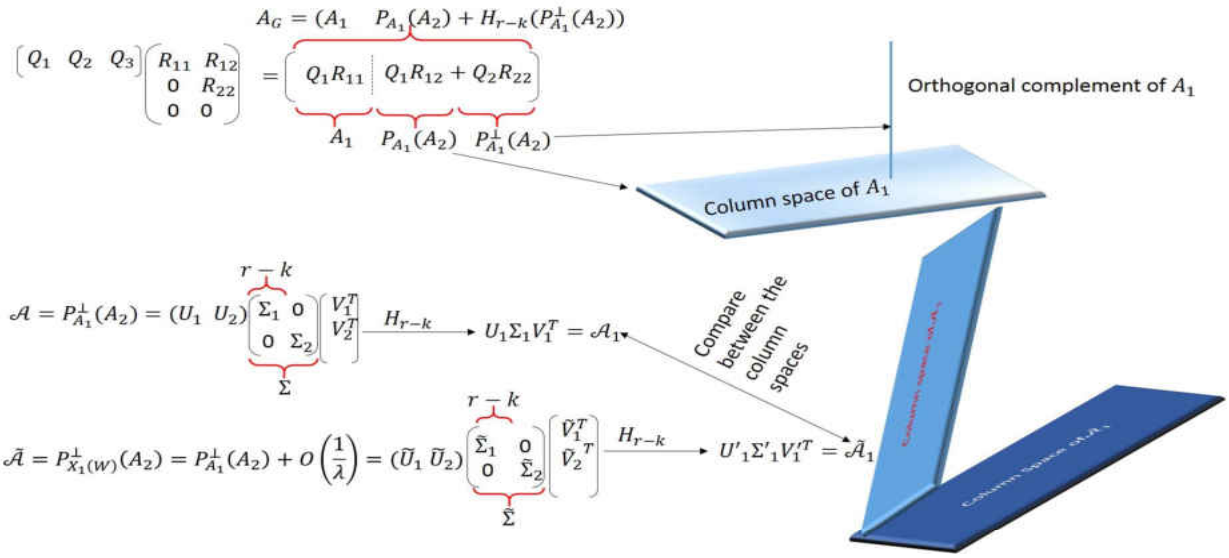

Figure 4.2: An overview of the matrix setup for Lemma 33, Lemma 34, and Lemma 35.

Remark 32. *For the case when there is an uniform weight in* $(W_1)_{ij} = \lambda > 0$*, one might refer to [27] for an alternative proof of Lemma 31. But the proof in [27] can not be applied in the more general case as in Lemma 31.*

Next, we will quote one of the most involved results of this chapter in Lemma 35. In this lemma, we will investigate how the weights $(W_1)_{ij} \to \infty$ and $W_2 = \mathbb{1}$ affect the hard-thresholding operator. We will first quote two classic results.

Lemma 33. *[13] Let* $\tilde{\mathcal{A}} = \mathcal{A} + E$ *and* $\sigma \neq 0$ *be a non-repeating singular value of the matrix* $\mathcal{A}$ *with* u *and* v *being left and right singular vectors respectively. Then as* $\lambda \to \infty$*, there is a unique singular value* $\tilde{\sigma}$ *of* $\tilde{\mathcal{A}}$ *such that*

$$\tilde{\sigma} = \sigma + u^T E v + O(\|E\|^2). \tag{4.17}$$

The lemma above will allow us to estimate the difference between the singular values of $\tilde{\mathcal{A}}$ and $\mathcal{A}$. However, the perturbation matrix E not only changes the singular values of $\mathcal{A}$, but also affects the column space of $\mathcal{A}$. Therefore, the perturbation measure of the singular values of $\tilde{\mathcal{A}}$ and $\mathcal{A}$ does not necessarily suffice our goal to compare between $H_{r-k}(\tilde{\mathcal{A}})$ and $H_{r-k}(\mathcal{A})$. This leads us to consider the column spaces of $\mathcal{A}_1$ and $\tilde{\mathcal{A}}_1$. One way to measure the distance between two subspaces is to measure the angle between them [14]. Davis and Kahan measured the difference of the angles between the invariant subspaces of a Hermitian matrix and its perturbed form as a function of their perturbation and the separation of their spectra. Wedin proposed a more generalized form. Using the generalized $\sin\theta$ Theorem of Wedin ([10]), the following results can be achieved (see Section 4.4 in [10]).

Lemma 34. *[10] Let $\mathcal{A}$ and $\tilde{\mathcal{A}}$ be given as*

$$\tilde{\mathcal{A}} = \tilde{\mathcal{A}}_1 + \tilde{\mathcal{A}}_2 = \mathcal{A}_1 + \mathcal{A}_2 + E = \mathcal{A} + E.$$

Assume there exists an $\alpha \geq 0$ and a $\delta > 0$ such that

$$\sigma_{min}(\tilde{\mathcal{A}}_1) \geq \alpha + \delta \ \text{ and } \ \sigma_{max}(\mathcal{A}_2) \leq \alpha,$$

then

$$\|\mathcal{A}_1 - \tilde{\mathcal{A}}_1\| \leq \|E\|(3 + \frac{\|\mathcal{A}_2\|}{\delta} + \frac{\|\tilde{\mathcal{A}}_2\|}{\delta}). \tag{4.18}$$

Now we will state our result.

Lemma 35. *If $\sigma_{r-k} > \sigma_{r-k+1}$, then*

$$H_{r-k}(\tilde{\mathcal{A}}) = H_{r-k}(\mathcal{A}) + O(\frac{1}{\lambda}),\ \lambda \to \infty. \tag{4.19}$$

Proof. Let the SVDs of $\mathcal{A},\ \tilde{\mathcal{A}}$ be given by

$$\mathcal{A} = U\Sigma V^T = (U_1\ U_2)\begin{pmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix}\begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix} =: \mathcal{A}_1 + \mathcal{A}_2, \tag{4.20}$$

$$\tilde{\mathcal{A}} = \tilde{U}\tilde{\Sigma}\tilde{V}^T = (\tilde{U}_1\ \tilde{U}_2)\begin{pmatrix}\tilde{\Sigma}_1 & 0\\ 0 & \tilde{\Sigma}_2\end{pmatrix}\begin{pmatrix}\tilde{V}_1^T\\ \tilde{V}_2^T\end{pmatrix} =: \tilde{\mathcal{A}}_1 + \tilde{\mathcal{A}}_2, \tag{4.21}$$

such that $U,\ \tilde{U} \in \mathbb{R}^{m\times m},\ V,\ \tilde{V} \in \mathbb{R}^{(n-k)\times(n-k)}$, and $\Sigma,\ \tilde{\Sigma} \in \mathbb{R}^{m\times(n-k)}$ with Σ and $\tilde{\Sigma}$ being diagonal matrices containing singular values of $\mathcal{A}$ and $\tilde{\mathcal{A}}$, respectively, arranged in a non-increasing order; $U_1,\ \tilde{U}_1 \in \mathbb{R}^{m\times(r-k)}, U_2,\ \tilde{U}_2 \in \mathbb{R}^{m\times(m-r+k)}, V_1,\ \tilde{V}_1 \in \mathbb{R}^{(n-k)\times(r-k)}$, and $V_2,\ \tilde{V}_2 \in \mathbb{R}^{(n-k)\times(n-r)}$. Using (4.20) and (4.21) we have (also following the structure proposed in Lemma 35):

$$\tilde{\mathcal{A}} = \tilde{\mathcal{A}}_1 + \tilde{\mathcal{A}}_2 = \mathcal{A}_1 + \mathcal{A}_2 + E = \mathcal{A} + E. \tag{4.22}$$

Then by (iii) of Lemma 31, we know that $E = O(\frac{1}{\lambda}),\ \lambda \to \infty$. Indeed, with the non-increasing arrangement of the singular values in Σ and $\tilde{\Sigma}$, and the fact that $E = O(\frac{1}{\lambda})$ as $\lambda \to \infty$, Lemma 33 immediately implies that

$$\Sigma_1 - \tilde{\Sigma}_1 = O(\frac{1}{\lambda}) \quad \text{and} \quad \Sigma_2 - \tilde{\Sigma}_2 = O(\frac{1}{\lambda}) \ \text{ as } \lambda \to \infty. \tag{4.23}$$

Note that, $\mathrm{r}(\mathcal{A}_1) = \mathrm{r}(\tilde{\mathcal{A}}_1) = r - k$, and, since $\sigma_{r-k} > \sigma_{r-k+1}$, we can choose δ such that

$$\delta \geq \frac{1}{2}(\sigma_{r-k} - \sigma_{r-k+1}) > 0.$$

In this way, for all large λ the assumption of Lemma 34 will be satisfied. Since $\mathcal{A}_1 = H_{r-k}(\mathcal{A})$ and $\tilde{\mathcal{A}}_1 = H_{r-k}(\tilde{\mathcal{A}})$, (4.18) can be written as

$$\|H_{r-k}(\mathcal{A}) - H_{r-k}(\tilde{\mathcal{A}})\| \leq \|E\|(3 + \frac{\|\mathcal{A}_2\|}{\delta} + \frac{\|\tilde{\mathcal{A}}_2\|}{\delta}). \tag{4.24}$$

Since $\mathcal{A}_2$ is fixed, $\|\mathcal{A}_2\| = O(1)$ as $\lambda \to \infty$. On the other hand, by (4.23), as $\lambda \to \infty$,

$$\tilde{\mathcal{A}}_2 = \tilde{U}_2\tilde{\Sigma}_2\tilde{V}_2^T = \tilde{U}_2(\Sigma_2 + O(\frac{1}{\lambda}))\tilde{V}_2^T = \tilde{U}_2\Sigma_2\tilde{V}_2^T + O(\frac{1}{\lambda}\tilde{U}_2\tilde{V}_2^T).$$

Now the unitary invariance of the matrix norm implies,

$$\|\tilde{\mathcal{A}}_2\| \leq \|\tilde{U}_2\Sigma_2\tilde{V}_2^T\| + O(\frac{1}{\lambda}\|\tilde{U}_2\tilde{V}_2^T\|) = \|\Sigma_2\| + O(\frac{1}{\lambda}),$$

which is bounded as $\lambda \to \infty$. Therefore (4.24) becomes

$$\|H_{r-k}(\mathcal{A}) - H_{r-k}(\tilde{\mathcal{A}})\| \leq C\|E\|, \tag{4.25}$$

for some constant $C > 0$ and for all large $\lambda \to \infty$. Thus

$$H_{r-k}(\tilde{\mathcal{A}}) = H_{r-k}(\mathcal{A}) + O(\frac{1}{\lambda}), \lambda \to \infty,$$

since $E = O(\frac{1}{\lambda})$ as $\lambda \to \infty$. This completes the proof of Lemma 35. □

Proof of Theorem 25. The proof is a consequence of Lemmas 31 and 35. □

Proof of Theorem 27. Note that,

$$\begin{aligned}
&\|(A_1 - \tilde{X}_1(W)) \odot W_1\|_F^2 + \|A_2 - \tilde{X}_2(W)\|_F^2 \\
=& \min_{\substack{X_1, X_2 \\ \mathrm{r}(X_1 \;\; X_2) \leq r}} \left(\|(A_1 - X_1) \odot W_1\|_F^2 + \|A_2 - X_2\|_F^2\right) \\
\leq& \|(A_1 - \tilde{X}_1(W)) \odot W_1\|_F^2 + \|A_2 - X_2\|_F^2,
\end{aligned}$$

for all $\mathrm{r}(\tilde{X}_1(W) \;\; X_2) \leq r$. So,

$$(\tilde{X}_1(W) \;\; \tilde{X}_2(W)) = \arg \min_{\substack{X_1 = \tilde{X}_1(W_1) \\ \mathrm{r}(X_1 \;\; X_2) \leq r}} \|(\tilde{X}_1(W) \;\; A_2) - (X_1 \;\; X_2)\|_F^2. \tag{4.26}$$

Therefore, by Theorem 17, $\tilde{X}_2(W) = P_{\tilde{X}_1(W)}(A_2) + H_{r-k}\left(P^{\perp}_{\tilde{X}_1(W)}(A_2)\right)$. □

Proof of Theorem 28. Let $\hat{X}(W) = (\hat{X}_1(W) \;\; \hat{X}_2(W))$. We need to verify that $\{\hat{X}(W)\}_W$ is a bounded set and every accumulation point is a solution to (3.1) for some r. Since $(\hat{X}_1(W) \;\; \hat{X}_2(W))$ is a solution to (4.8), we have

$$\begin{aligned}
&\|(A_1 - \hat{X}_1(W)) \odot W_1\|_F^2 + \|(A_2 - \hat{X}_2(W)) \odot W_2\|_F^2 + \tau \mathrm{r}(\hat{X}_1(W) \;\; \hat{X}_2(W)) \\
&\leq \|(A_1 - X_1) \odot W_1\|_F^2 + \|(A_2 - X_2) \odot W_2\|_F^2 + \tau \mathrm{r}(X_1 \;\; X_2).
\end{aligned} \tag{4.27}$$

for all $(X_1 \;\; X_2)$. By choosing $X_1 = A_1, X_2 = 0$, we can obtain a constant $m_3 := \|A_2 \odot W_2\|_F^2 + \tau \mathrm{r}(A_1 \;\; 0)$ such that $\|(A_1 - \hat{X}_1(W)) \odot W_1\|_F^2 + \|(A_2 - \hat{X}_2(W)) \odot W_2\|_F^2 \leq m_3$. Therefore, $\{\hat{X}_1(W) \;\; \hat{X}_2(W)\}$ is bounded. Let $(X_1^{**} \;\; X_2^{**})$ be an accumulation point of the sequence.

We only need to show that $(X_1^{**} \;\; X_2^{**}) \in \bigcup_r \mathcal{A}_G^r$. As in the proof of Lemma 31 (i), we can show that

$$\lim_{\substack{(W_1)_{ij}\to\infty \\ (W_2)_{ij}\to 1}} \hat{X}_1(W) = A_1. \tag{4.28}$$

Now, taking limit and setting $X_1 = A_1$ in (4.27), we can obtain,

$$\|A_2 - X_2^{**}\|_F^2 + \tau \mathrm{r}(A_1 \;\; X_2^{**}) \le \|A_2 - X_2\|_F^2 + \tau \mathrm{r}(A_1 \;\; X_2), \tag{4.29}$$

for all X_2. If we denote $r^{**} = \mathrm{r}(A_1 \;\; X_2^{**})$, then for X_2 with $\mathrm{r}(A_1 \;\; X_2) \le r^{**}$, (4.29) yields

$$\|A_2 - X_2^{**}\|_F^2 \le \|A_2 - X_2\|_F^2. \tag{4.30}$$

So, X_2^{**} is a solution to the problem of Golub, Hoffman, and Stewart. Thus, by Theorem 17,

$$X_2^{**} = P_{A_1}(A_2) + H_{r^{**}-k}\left(P_{A_1}^{\perp}(A_2)\right).$$

This, together with (4.28) completes the proof. □

Proof of Theorem 29. Let $\hat{X}(W) = (\hat{X}_1(W) \;\; \hat{X}_2(W))$ solve the minimization problem (4.8). For convenience, we will drop the dependence on W in our notations. Then $\hat{X}$ satisfies

$$\begin{aligned} &\|(A_1 - \hat{X}_1) \odot W_1\|_F^2 + \|(A_2 - \hat{X}_2) \odot W_2\|_F^2 + \tau \mathrm{r}(\hat{X}_1 \;\; \hat{X}_2) \\ &\le \|(A_1 - X_1^{\dagger}) \odot W_1\|_F^2 + \|(A_2 - X_2^{\dagger}) \odot W_2\|_F^2 + \tau \mathrm{r}(X_1^{\dagger} \;\; X_2^{\dagger}), \end{aligned} \tag{4.31}$$

for all $X^{\dagger} = (X_1^{\dagger} \;\; X_2^{\dagger}) \in \mathbb{R}^{m\times n}$. By choosing $X_1^{\dagger} = A_1$ and $X_1^{\dagger} = \hat{X}_2$ in (4.31) we obtain

$$\sum_{\substack{1\le i\le m \\ 1\le j\le k}} ((A_1)_{ij} - (\hat{X}_1)_{ij})^2 (W_1)_{ij}^2 \le \tau \mathrm{r}(A_1 \;\; \hat{X}_2) - \tau \mathrm{r}(\hat{X}_1 \;\; \hat{X}_2) =: C.$$

Therefore,

$$\hat{X}_1 \to A_1, \;\; (W_1)_{ij} \to \infty. \tag{4.32}$$

Next we choose $X_1^{\dagger} = \hat{X}_1$ in (4.31) and find, for all $X_2^{\dagger}$,

$$\|(A_2 - \hat{X}_2) \odot W_2\|_F^2 + \tau \mathrm{r}(\hat{X}_1 \;\; \hat{X}_2) \le \|(A_2 - X_2^{\dagger}) \odot W_2\|_F^2 + \tau \mathrm{r}(\hat{X}_1 \;\; X_2^{\dagger}). \tag{4.33}$$

As in the proof of (ii) of Lemma 31, assume $\mathrm{r}(A_1) = k$ and consider a QR decomposition of A :

$$A = QR = Q(R_1 \;\; R_2) = Q \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \\ 0 & 0 \end{pmatrix}.$$

Write $\hat{R} := Q^T \hat{X} = (\hat{R}_1 \;\; \hat{R}_2) = \begin{pmatrix} \hat{R}_{11} & \hat{R}_{12} \\ \hat{R}_{21} & \hat{R}_{22} \\ \hat{R}_{31} & \hat{R}_{32} \end{pmatrix}$ and let $R^\dagger := (R_1^\dagger \;\; R_2^\dagger) = \begin{pmatrix} R_{11}^\dagger & R_{12}^\dagger \\ R_{21}^\dagger & R_{22}^\dagger \\ R_{31}^\dagger & R_{32}^\dagger \end{pmatrix}$ be in compatible block partitions. Since the rank of a matrix is invariant under an unitary transformation, (4.33) can be rewritten as

$$\begin{aligned} &\|(A_2 - \hat{X}_2) \odot W_2\|_F^2 + \tau \mathrm{r}(Q^T \hat{X}_1 \;\; Q^T \hat{X}_2) \\ &\leq \|(A_2 - X_2^\dagger) \odot W_2\|_F^2 + \tau \mathrm{r}(Q^T \hat{X}_1 \;\; Q^T X_2^\dagger). \end{aligned} \tag{4.34}$$

When λ is large enough, $\hat{R}_{11}$ is nonsingular by (4.32) and the fact that $\mathrm{r}(A_1) = k$ and we can perform the row and column operations on the second term on left hand side of (4.34) to get:

$$\|(A_2 - \hat{X}_2) \odot W_2\|_F^2 + \tau \mathrm{r} \begin{pmatrix} \hat{R}_{11} & 0 \\ 0 & \hat{R}_{22} - \hat{R}_{21}\hat{R}_{11}^{-1}\hat{R}_{12} \\ 0 & \hat{R}_{32} - \hat{R}_{31}\hat{R}_{11}^{-1}\hat{R}_{12} \end{pmatrix},$$

which is equal to

$$\|(A_2 - \hat{X}_2) \odot W_2\|_F^2 + \tau k + \tau \mathrm{r} \begin{pmatrix} \hat{R}_{22} - \hat{R}_{21}\hat{R}_{11}^{-1}\hat{R}_{12} \\ \hat{R}_{32} - \hat{R}_{31}\hat{R}_{11}^{-1}\hat{R}_{12} \end{pmatrix}.$$

Performing the similar operations on the right hand side we obtain

$$\|(A_2 - X_2^\dagger) \odot W_2\|_F^2 + \tau \mathrm{r}(\hat{R}_{11}) + \tau \mathrm{r} \begin{pmatrix} R_{22}^\dagger - \hat{R}_{21}\hat{R}_{11}^{-1}R_{12}^\dagger \\ R_{32}^\dagger - \hat{R}_{31}\hat{R}_{11}^{-1}R_{12}^\dagger \end{pmatrix}.$$

Substituting these back in (4.34) we obtain

$$\|(A_2 - \hat{X}_2) \odot W_2\|_F^2 + \tau \mathrm{r} \begin{pmatrix} \hat{R}_{22} - \hat{R}_{21}\hat{R}_{11}^{-1}\hat{R}_{12} \\ \hat{R}_{32} - \hat{R}_{31}\hat{R}_{11}^{-1}\hat{R}_{12} \end{pmatrix}$$

$$\leq \|(A_2 - X_2^\dagger) \odot W_2\|_F^2 + \tau \mathrm{r} \begin{pmatrix} R_{22}^\dagger - \hat{R}_{21}\hat{R}_{11}^{-1}R_{12}^\dagger \\ R_{32}^\dagger - \hat{R}_{31}\hat{R}_{11}^{-1}R_{12}^\dagger \end{pmatrix}, \tag{4.35}$$

for all $R_{12}^\dagger, R_{22}^\dagger,$ and $R_{32}^\dagger$. From Theorem 28, we know that $(\hat{R}_1 \quad \hat{R}_2)$ has accumulation points which belong to $\underset{0\leq r\leq \min\{m,n\}}{\cup} \mathcal{A}_G^r$. We are going to show that $\lim\limits_{\substack{(W_1)_{ij}\to\infty \\ (W_2)_{ij}\to 1}} \hat{R}_2$ indeed exists. Assume $\lim\limits_{\substack{(W_1)_{ij}\to\infty \\ (W_2)_{ij}\to 1}} \begin{pmatrix} \hat{R}_{12} \\ \hat{R}_{22} \\ \hat{R}_{32} \end{pmatrix} = \begin{pmatrix} \hat{R}_{12}^* \\ \hat{R}_{22}^* \\ \hat{R}_{32}^* \end{pmatrix}$ be an accumulation point. From (4.32), using the fact that $\hat{R}_{11} \to R_{11}, \hat{R}_{21} \to 0,$ and $\hat{R}_{31} \to 0,$ as $(W_1)_{ij} \to \infty, (W_2)_{ij} \to 1$ in (4.35) we get

$$\|A_2 - \hat{X}_2^*\|_F^2 + \tau \mathrm{r} \begin{pmatrix} \hat{R}_{22}^* \\ \hat{R}_{32}^* \end{pmatrix} \leq \|A_2 - X_2^\dagger\|_F^2 + \tau \mathrm{r} \begin{pmatrix} R_{22}^\dagger \\ R_{32}^\dagger \end{pmatrix}, \tag{4.36}$$

for all $R_{12}^\dagger, R_{22}^\dagger,$ and $R_{32}^\dagger$. Since Frobenius norm is unitarily invariant, (4.36) reduces to

$$\| \begin{pmatrix} R_{12} \\ R_{22} \\ 0 \end{pmatrix} - \begin{pmatrix} \hat{R}_{12}^* \\ \hat{R}_{22}^* \\ \hat{R}_{32}^* \end{pmatrix} \|_F^2 + \tau \mathrm{r} \begin{pmatrix} \hat{R}_{22}^* \\ \hat{R}_{32}^* \end{pmatrix} \leq \| \begin{pmatrix} R_{12} \\ R_{22} \\ 0 \end{pmatrix} - \begin{pmatrix} R_{12}^\dagger \\ R_{22}^\dagger \\ R_{32}^\dagger \end{pmatrix} \|_F^2 + \tau \mathrm{r} \begin{pmatrix} R_{22}^\dagger \\ R_{32}^\dagger \end{pmatrix}, \tag{4.37}$$

for all $R_{12}^\dagger, R_{22}^\dagger,$ and $R_{32}^\dagger$. Substituting $R_{22}^\dagger = \hat{R}_{22}^*, R_{32}^\dagger = \hat{R}_{32}^*,$ and $R_{12}^\dagger = R_{12},$ in (4.37) yields

$$\|R_{12} - \hat{R}_{12}^*\|_F^2 \leq 0,$$

which implies $\lim\limits_{\substack{(W_1)_{ij}\to\infty \\ (W_2)_{ij}\to 1}} \hat{R}_{12} = R_{12}$. Next, substituting $R_{12}^\dagger = \hat{R}_{12}^*$ in (4.37) we find

$$\| \begin{pmatrix} R_{22} \\ 0 \end{pmatrix} - \begin{pmatrix} \hat{R}_{22}^* \\ \hat{R}_{32}^* \end{pmatrix} \|_F^2 + \tau \mathrm{r} \begin{pmatrix} \hat{R}_{22}^* \\ \hat{R}_{32}^* \end{pmatrix} \leq \| \begin{pmatrix} R_{22} \\ 0 \end{pmatrix} - \begin{pmatrix} R_{22}^\dagger \\ R_{32}^\dagger \end{pmatrix} \|_F^2 + \tau \mathrm{r} \begin{pmatrix} R_{22}^\dagger \\ R_{32}^\dagger \end{pmatrix}, \tag{4.38}$$

for all $R_{22}^{\dagger}, R_{32}^{\dagger}$. Let $\bar{R}^* = \begin{pmatrix} \hat{R}_{22}^* \\ \hat{R}_{32}^* \end{pmatrix}$ and $r^* = \mathrm{r}(\bar{R}^*)$, then (4.38) implies

$$\| \begin{pmatrix} R_{22} \\ 0 \end{pmatrix} - \bar{R}^* \|_F^2 \leq \| \begin{pmatrix} R_{22} \\ 0 \end{pmatrix} - R^* \|_F^2, \tag{4.39}$$

for all $R^* \in \mathbb{R}^{(m-k)\times(n-k)}$ with $\mathrm{r}(R^*) \leq r^*$. So $\bar{R}^*$ solves a problem of classical low-rank approximation of $\begin{pmatrix} R_{22} \\ 0 \end{pmatrix}$. Note that, $Q_2 \begin{pmatrix} R_{22} \\ 0 \end{pmatrix} = P_{A_1}^{\perp}(A_2)$ (see (4.10)) and it is assumed that $P_{A_1}^{\perp}(A_2)$ has distinct singular values. So there exists a unique $\bar{R}^*$ which is given by $\bar{R}^* = H_{r^*} \begin{pmatrix} R_{22} \\ 0 \end{pmatrix}$ as in (4.1)). Therefore there is only one accumulation point of $\{\hat{R}_2\}$ and so $\lim_{\substack{(W_1)_{ij}\to\infty \\ (W_2)_{ij}\to 1}} \hat{R}_2$ exists. It remains for us to identify this unique accumulation point. Assume that

$$\begin{pmatrix} R_{22} \\ 0 \end{pmatrix} = Q^T \Sigma P$$

is a SVD of $\begin{pmatrix} R_{22} \\ 0 \end{pmatrix}$. Then, for any $R^* \in \mathbb{R}^{(m-k)\times(n-k)}$, (4.38) gives

$$\begin{aligned} &\|\Sigma - Q\bar{R}^* P^T\|_F^2 + \tau \mathrm{r}(Q\bar{R}^* P^T) \\ &\leq \|\Sigma - QR^* P^T\|_F^2 + \tau \mathrm{r}(QR^* P^T), \end{aligned} \tag{4.40}$$

Since $r^* = \mathrm{r}(\bar{R}^*)$ and $Q\bar{R}^* P^T = \mathrm{diag}(\sigma_1\ \sigma_2\ \cdots \sigma_{r^*}\ 0 \cdots 0)$, choosing R^* such that

$$QR^* P^T = \mathrm{diag}(\sigma_1\ \sigma_2\ \cdots \sigma_{r^*+1}\ 0 \cdots 0),$$

and using (4.40) we find

$$\sigma_{r^*+2}^2 + \cdots + \sigma_n^2 + \tau \geq \sigma_{r^*+1}^2 + \sigma_{r^*+2}^2 + \cdots + \sigma_n^2.$$

Next we choose R^* such that

$$QR^* P^T = \mathrm{diag}(\sigma_1\ \sigma_2\ \cdots \sigma_{r^*-1}\ 0 \cdots 0),$$

and so $\mathrm{r}(R^*) = r^* - 1 < r^*$. Now (4.39) and Ektart-Young-Mirsky's theorem then imply the equality in (4.40) can not hold. So,

$$\sigma_{r^*}^2 + \cdots + \sigma_n^2 - \tau > \sigma_{r^*+1}^2 + \sigma_{r^*+2}^2 + \cdots + \sigma_n^2.$$

Therefore, we obtain

$$\sigma_{r^*}^2 > \tau \geq \sigma_{r^*+1}^2. \tag{4.41}$$

It is easy to see that if (4.41) holds then $\mathrm{r}(\bar{R}^*) = r^*$. So,

$$\mathrm{r}(\bar{R}^*) = r^* \text{ if and only if } \sigma_{r^*}^2 > \tau \geq \sigma_{r^*+1}^2,$$

and in this case when $\mathrm{r}(\bar{R}^*) = r^*$, we have shown that $\lim\limits_{\substack{(W_1)_{ij}\to\infty \\ (W_2)_{ij}\to 1}} \hat{R}_2 = \begin{pmatrix} R_{12} \\ H_{r^*} \begin{pmatrix} R_{22} \\ 0 \end{pmatrix} \end{pmatrix}$. Thus, together with (4.32), this implies

$$\begin{aligned} \lim_{\substack{(W_1)_{ij}\to\infty \\ (W_2)_{ij}\to 1}} (\hat{X}_1 \ \hat{X}_2) = Q(\lim_{\substack{(W_1)_{ij}\to\infty \\ (W_2)_{ij}\to 1}} (\hat{R}_1 \ \hat{R}_2)) &= Q \begin{pmatrix} & R_{12} \\ R_1 & H_{r^*} \begin{pmatrix} R_{22} \\ 0 \end{pmatrix} \end{pmatrix} \\ &= (A_1 \quad Q_1 R_{12} + H_{r^*} \left(Q_2 \begin{pmatrix} R_{22} \\ 0 \end{pmatrix} \right)), \end{aligned}$$

which is the same as

$$\begin{pmatrix} A_1 & P_{A_1}(A_2) + H_{r^*} \left(P_{A_1}^{\perp}(A_2) \right) \end{pmatrix}.$$

This completes the proof. □

4.4 Numerical Algorithm [2, 6]

In this section we propose a numerical algorithm to solve a special case of (4.6), which, in general, does not have a closed form solution [37, 39]. Note (4.6) can be written as

$$\min_{\substack{X_1, X_2 \\ \mathrm{r}(X_1 \ X_2) \leq r}} \left(\|(A_1 - X_1) \odot W_1\|_F^2 + \|(A_2 - X_2) \odot W_2\|_F^2 \right).$$

We assume that $\mathrm{r}(X_1) = k$. It can be verified that any X_2 such that $\mathrm{r}(X_1 \;\; X_2) \leq r$ can be given in the form

$$X_2 = X_1 C + BD,$$

for some arbitrary matrices $B \in \mathbb{R}^{m\times(r-k)}$, $D \in \mathbb{R}^{(r-k)\times(n-k)}$, and $C \in \mathbb{R}^{k\times(n-k)}$. Here we will focus on a special case when $W_2 = \mathbb{1}$ in solving:

$$\min_{X_1,C,B,D} \left(\|(A_1 - X_1) \odot W_1\|_F^2 + \|A_2 - X_1 C - BD\|_F^2 \right). \tag{4.42}$$

Writing (4.6) in the form (5.2) is not a new approach. A careful reader should note that, for the special choice of the weight matrix the problem (5.2) can be written using a block structure:

$$\min_{X_1,C,B,D} \left\{ \| \left((A_1 \;\; A_2) - (X_1 \;\; B) \begin{pmatrix} I_k & C \\ 0 & D \end{pmatrix} \right) \odot (W_1 \;\; \mathbb{1}) \|_F^2 \right\},$$

which is equivalent to the alternating weighted least squares algorithm in the literature [39, 23]. But in our case we will not follow the algorithm proposed in [23]. Because the structure we employed in (5.2) will serve two purposes for us: One is to verify the rate given by Theorem 25 numerically and to gain some insight on the sharpness of the rate ($O(\frac{1}{\lambda})$, as $\lambda \to \infty$); the other one is to demonstrate a fast and simple numerical procedure based on alternating direction method in solving the weighted low-rank approximation problem that also allows detailed convergence analysis which is usually hard to obtain in other algorithms proposed in the literature [39, 37, 23]. For the special structure of the weight our algorithm is more efficient than [23] (see Algorithm 3.1, page 42) and can handle bigger size matrices which we will demonstrate in the numerical result section. If $k = 0$, then (5.2) is an unweighted rank r factorization of A_2 and is known as alternating least squares problem [17, 18, 20]. Denote $F(X_1, C, B, D) = \|(A_1 - X_1) \odot W_1\|_F^2 + \|A_2 - X_1 C - BD\|_F^2$ as the objective function. The above problem can be numerically solved by using an alternating strategy [9, 22] of

minimizing the function with respect to each component iteratively:

$$\begin{cases} (X_1)_{p+1} = \arg\min_{X_1} F(X_1, C_p, B_p, D_p), \\ C_{p+1} = \arg\min_{C} F((X_1)_{p+1}, C, B_p, D_p), \\ B_{p+1} = \arg\min_{B} F((X_1)_{p+1}, C_{p+1}, B, D_p), \\ \text{and, } D_{p+1} = \arg\min_{D} F((X_1)_{p+1}, C_{p+1}, B_{p+1}, D). \end{cases} \tag{4.43}$$

Note that each of the minimizing problem for X_1, C, B, and D can be solved explicitly by looking at the partial derivatives of $F(X_1, C, B, D)$. But finding an update rule for X_1 turns out to be more involved than the other three variables. We update X_1 element wise along each row. Therefore we will use the notation $X_1(i,:)$ to denote the i-th row of the matrix X_1. We set $\frac{\partial}{\partial X_1} F(X_1, C_p, B_p, D_p)|_{X_1=(X_1)_{p+1}} = 0$ and obtain

$$-(A_1 - (X_1)_{p+1}) \odot W_1 \odot W_1 - (A_2 - (X_1)_{p+1} C_p - B_p D_p) C_p^T = 0. \tag{4.44}$$

Solving the above expression for X_1 sequentially along each row gives

$$(X_1(i,:))_{p+1} = (E(i,:))_p (\text{diag}(W_1^2(i,1)\ W_1^2(i,2) \cdots W_1^2(i,k)) + C_p C_p^T)^{-1},$$

where $E_p = A_1 \odot W_1 \odot W_1 + (A_2 - B_p D_p) C_p^T$. The reader should note that, for each row $X_1(i,:)$, we can find a matrix $\mathcal{L}_i = \text{diag}(W_1^2(i,1)\ W_1^2(i,2) \cdots W_1^2(i,k)) + C_p C_p^T$ such that the above system of equations are equivalent to solving a least squares solution of $\mathcal{L}_i (X_1(i,:))_{p+1}^T = (E(i,:))_p^T$ for each i. Next we find, C_{p+1} satisfies

$$\frac{\partial}{\partial C} F(X_1, C, B_p, D_p)|_{C=C_{p+1}} = 0,$$

which implies

$$-(X_1)_{p+1}^T (A_2 - (X_1)_{p+1} C_{p+1} - B_p D_p) = 0, \tag{4.45}$$

and consequently can be solved as long as $(X_1)_{p+1}$ is of full rank. Therefore solving for C_{p+1} gives

$$C_{p+1} = ((X_1)_{p+1}^T (X_1)_{p+1})^{-1} ((X_1)_{p+1}^T A_2 - (X_1)_{p+1}^T B_p D_p).$$

Similarly, B_{p+1} satisfies

$$-A_2 D_p^T + (X_1)_{p+1} C_{p+1} D_p^T + B_{p+1} D_p D_p^T = 0. \tag{4.46}$$

Solving (4.46) for B_{p+1} obtains (assuming D_p is of full rank)

$$B_{p+1} = (A_2 D_p^T - (X_1)_{p+1} C_{p+1} D_p^T)(D_p D_p^T)^{-1}.$$

Finally, D_{p+1} satisfies

$$-B_{p+1}^T A_2 + B_{p+1}^T (X_1)_{p+1} C_{p+1} + B_{p+1}^T B_{p+1} D_{p+1} = 0, \tag{4.47}$$

and we can write (assuming B_{p+1} is of full rank)

$$D_{p+1} = (B_{p+1}^T B_{p+1})^{-1}(B_{p+1}^T A_2 - B_{p+1}^T (X_1)_{p+1} C_{p+1}).$$

Algorithm 3: WLR Algorithm

Input : $A = (A_1 \;\; A_2) \in \mathbb{R}^{m\times n}$ (the given matrix);

$W = (W_1 \;\; W_2) \in \mathbb{R}^{m\times n}, W_2 = \mathbb{1} \in \mathbb{R}^{m\times(n-k)}$ (the weight); threshold $\epsilon > 0$.;

Initialize: $(X_1)_0, C_0, B_0, D_0$;

while *not converged* **do**

$E_p = A_1 \odot W_1 \odot W_1 + (A_2 - B_p D_p) C_p^T$;

$(X_1(i,:))_{p+1} = (E(i,:))_p (\mathrm{diag}(W_1^2(i,1)\; W_1^2(i,2) \cdots W_1^2(i,k)) + C_p C_p^T)^{-1}$;

$C_{p+1} = ((X_1)_{p+1}^T (X_1)_{p+1})^{-1}((X_1)_{p+1}^T A_2 - (X_1)_{p+1}^T B_p D_p)$;

$B_{p+1} = (A_2 D_p^T - (X_1)_{p+1} C_{p+1} D_p^T)(D_p D_p^T)^{-1}$;

$D_{p+1} = (B_{p+1}^T B_{p+1})^{-1}(B_{p+1}^T A_2 - B_{p+1}^T (X_1)_{p+1} C_{p+1})$;

$p = p + 1$;

end

Output : $(X_1)_{p+1}, (X_1)_{p+1} C_{p+1} + B_{p+1} D_{p+1}$.

4.4.1 Convergence Analysis

Next we will discuss the convergence of our numerical algorithm. Since the objective function F is convex only in each of the component X_1, B, C, and D; it is hard to argue about the global convergence of the algorithm. In Theorem 38 and 39, under some special assumptions when the limit of the individual sequence exists, we show that the limit points are going to be a stationary point of F. To establish our main convergence results in Theorem 38 and 39, the following equality will be very helpful.

Theorem 36. *For a fixed* $(W_1)_{ij} > 0$, *and* $p = 1, 2, \cdots$, *let* $m_p = F((X_1)_p, C_p, B_p, D_p)$. *Then,*

$$\begin{aligned} m_p - m_{p+1} =& \|((X_1)_p - (X_1)_{p+1}) \odot W_1\|_F^2 + \|((X_1)_p - (X_1)_{p+1})C_p\|_F^2 \\ &+ \|(X_1)_{p+1}(C_p - C_{p+1})\|_F^2 + \|(B_p - B_{p+1})D_p\|_F^2 + \|B_{p+1}(D_p - D_{p+1})\|_F^2. \end{aligned} \tag{4.48}$$

Proof: Denote

$$\begin{cases} m_p - F((X_1)_{p+1}, C_p, B_p, D_p) = d_1, \\ F((X_1)_{p+1}, C_p, B_p, D_p) - F((X_1)_{p+1}, C_{p+1}, B_p, D_p) = d_2, \\ F((X_1)_{p+1}, C_{p+1}, B_p, D_p) - F((X_1)_{p+1}, C_{p+1}, B_{p+1}, D_p) = d_3, \\ \text{and, } F((X_1)_{p+1}, C_{p+1}, B_{p+1}, D_p) - m_{p+1} = d_4. \end{cases} \tag{4.49}$$

Therefore,

$$\begin{aligned} d_1 &= \|(A_1 - (X_1)_p) \odot W_1\|_F^2 + \|A_2 - (X_1)_p C_p - B_p D_p\|_F^2 - \|(A_1 - (X_1)_{p+1}) \odot W_1\|_F^2 \\ &\quad - \|A_2 - (X_1)_{p+1} C_p - B_p D_p\|_F^2 \\ &= \sum_{i,j} ((A_1 - (X_1)_p)_{ij}^2 (W_1)_{ij}^2 - \sum_{i,j} ((A_1 - (X_1)_{p+1})_{ij}^2 (W_1)_{ij}^2 + \|A_2 - (X_1)_p C_p\|_F^2 \\ &\quad - \|A_2 - (X_1)_{p+1} C_p\|_F^2 - 2\langle A_2 - (X_1)_p C_p, B_p D_p\rangle + 2\langle A_2 - (X_1)_{p+1} C_p, B_p D_p\rangle \end{aligned}$$

$$
\begin{aligned}
&= \sum_{i,j} (X_1)_p)_{ij}^2 (W_1)_{ij}^2 - \sum_{i,j} (X_1)_{p+1})_{ij}^2 (W_1)_{ij}^2 + 2\sum_{i,j} (A_1)_{ij}(X_1)_{p+1} - X_1)_p)_{ij}^2 (W_1)_{ij}^2 \\
&+ \|(X_1)_p C_p\|_F^2 + \|(X_1)_{p+1} C_p\|_F^2 - 2\langle (A_2, ((X_1)_p - (X_1)_{p+1})C_p, \rangle \\
&+ 2\langle ((X_1)_p - (X_1)_{p+1})C_p, B_p D_p \rangle \\
&= \|(X_1)_p \odot W_1\|_F^2 - \|(X_1)_{p+1} \odot W_1\|_F^2 + 2\langle A_1 \odot W_1 \odot W_1, (X_1)_{p+1} - (X_1)_p \rangle \\
&+ \|(X_1)_p C_p\|_F^2 - \|(X_1)_p C_{p+1}\|_F^2 - 2\langle ((X_1)_p - (X_1)_{p+1})C_p, A_2 - B_p D_p \rangle. \qquad (4.50)
\end{aligned}
$$

Note that,

$$
(((X_1)_{p+1} - A_1) \odot W_1 \odot W_1) = (A_2 - (X_1)_{p+1} C_p - B_p D_p) C_p^T,
$$

as $(X_1)_{p+1}$ satisfies (4.44). Post multiplying both sides of the above relation by $((X_1)_p - (X_1)_{p+1})^T$ gives us

$$
(((X_1)_{p+1} - A_1) \odot W_1 \odot W_1)((X_1)_p - (X_1)_{p+1})^T = (A_2 - (X_1)_{p+1} C_p - B_p D_p) C_p^T ((X_1)_p - (X_1)_{p+1})^T,
$$

which is

$$
\begin{aligned}
&(A_1 \odot W_1 \odot W_1)((X_1)_{p+1} - (X_1)_p)^T - (A_2 - B_p D_p) C_p^T ((X_1)_p - (X_1)_{p+1})^T \\
&= (X_1)_{p+1} C_p C_p^T ((X_1)_{p+1} - (X_1)_p)^T - (((X_1)_{p+1} \odot W_1 \odot W_1)((X_1)_p - (X_1)_{p+1})^T
\end{aligned}
$$

This, together with (4.50), will lead us to

$$
\begin{aligned}
d_4 &= \|(X_1)_p \odot W_1\|_F^2 - \|(X_1)_{p+1} \odot W_1\|_F^2 - 2\langle (X_1)_{p+1} \odot W_1 \odot W_1, (X_1)_{p+1} - (X_1)_p \rangle \\
&\quad + \|(X_1)_p C_p\|_F^2 - \|(X_1)_p C_{p+1}\|_F^2 - 2\langle ((X_1)_p - (X_1)_{p+1})C_p, (X_1)_{p+1} C_p \rangle \\
&\quad = \sum_{i,j} \left(((X_1)_p)_{ij}^2 - ((X_1)_{p+1})_{ij}^2 - 2((X_1)_{p+1})_{ij} ((X_1)_{p+1} - (X_1)_p)_{ij} (w_1)_{ij}^2 \right) \\
&\quad + \|(X_1)_p C_p\|_F^2 + \|(X_1)_p C_{p+1}\|_F^2 - 2\langle ((X_1)_p C_p, (X_1)_{p+1})C_p \rangle \\
&\quad = \sum_{i,j} \left(((X_1)_p)_{ij}^2 + ((X_1)_{p+1})_{ij}^2 - 2((X_1)_{p+1} (X_1)_p)_{ij} (w_1)_{ij}^2 \right) \\
&\quad + \|((X_1)_p - (X_1)_{p+1}) C_p\|_F^2 \\
&\quad = \|((X_1)_p - (X_1)_{p+1}) \odot W_1\|_F^2 + \|((X_1)_p - (X_1)_{p+1}) C_p\|_F^2. \qquad (4.51)
\end{aligned}
$$

Similarly we find

$$\begin{cases} d_2 = \|(X_1)_{p+1}(C_p - C_{p+1})\|_F^2, \\ d_3 = \|(B_p - B_{p+1})D_p\|_F^2, \\ d_4 = \|B_{p+1}(D_p - D_{p+1})\|_F^2. \end{cases} \tag{4.52}$$

Combining them together we have the desired result. ■

Theorem 40 implies a lot of interesting convergence properties of the algorithm. For example, we have the following estimates.

Corollary 37. *We have*

(i) $m_p - m_{p+1} \geq \frac{1}{2}\|B_{p+1}D_{p+1} - B_pD_p\|_F^2$ *for all* p.

(ii) $m_p - m_{p+1} \geq \|((X_1)_p - (X_1)_{p+1}) \odot W_1\|_F^2$ *for all* p.

PROOF: *(i).* From (4.48) we can write, for all p,

$$\begin{aligned} & m_p - m_{p+1} \\ & \geq \|B_{p+1}(D_p - D_{p+1})\|_F^2 + \|(B_p - B_{p+1})D_p\|_F^2 \\ & = \frac{1}{2}(\|B_{p+1}D_{p+1} - B_pD_p\|_F^2 + \|2B_{p+1}D_p - B_{p+1}D_{p+1} - B_pD_p\|_F^2), \end{aligned}$$

by parallelogram identity. Therefore,

$$m_p - m_{p+1} \geq \frac{1}{2}\|B_{p+1}D_{p+1} - B_pD_p\|_F^2.$$

This completes the proof.

(ii). This follows immediately from (4.48). ■

We now can state some convergence results as a consequence of Theorem 40 and Corollary 37.

Theorem 38. *(i) We have the following:* $\sum_{p=1}^{\infty} \|B_{p+1}D_{p+1} - B_pD_p\|_F^2 < \infty$, *and* $\sum_{p=1}^{\infty} (\|((X_1)_p - (X_1)_{p+1}) \odot W_1\|) < \infty$.

(ii) If $\sum_{p=1}^{\infty}\sqrt{m_p - m_{p+1}} < +\infty$, *then* $\lim_{p\to\infty} B_pD_p$ *and* $\lim_{p\to\infty}(X_1)_p$ *and exist. Furthermore if we write* $L^* := \lim_{p\to\infty} B_pD_p$ *then* $\lim_{p\to\infty} B_{p+1}D_p = L^*$ *for all* p.

PROOF: *(i).* From Corollary 37 we can write, for $N > 0$,

$$2(m_1 - m_{N+1}) \geq \sum_{p=1}^{N}\left(\|B_{p+1}D_{p+1} - B_pD_p\|_F^2\right),$$

$$\text{and } \; m_1 - m_{N+1} \geq \sum_{p=1}^{\infty}\left(\|((X_1)_p - (X_1)_{p+1}) \odot W_1\|\right) \geq \lambda^2 \sum_{p=1}^{N}\|(X_1)_p - (X_1)_{p+1}\|_F^2.$$

Recall, $\lambda = \min_{\substack{1\leq i\leq m\\ 1\leq j\leq k}} (W_1)_{ij}$. Also note that, $\{m_p\}_{p=1}^{\infty}$ is a decreasing non-negative sequence. Hence the results follows.

(ii). Again using Corollary 37 we can write, for $N > 0$,

$$\begin{aligned}\frac{1}{\sqrt{2}}\left(\|\sum_{p=1}^{N}(B_{p+1}D_{p+1} - B_pD_p)\|_F\right) &\leq \frac{1}{\sqrt{2}}\sum_{p=1}^{N}\left(\|B_{p+1}D_{p+1} - B_pD_p\|_F\right)\\ &\leq \sum_{p=1}^{N}\sqrt{m_p - m_{p+1}},\end{aligned}$$

where the first inequality is due to triangle inequality and the second inequality follows from (i). So

$$\frac{1}{\sqrt{2}}\left(\|\sum_{p=1}^{N}(B_{p+1}D_{p+1} - B_pD_p)\|_F\right) \leq \sum_{p=1}^{N}\sqrt{m_p - m_{p+1}},$$

which implies $\sum_{p=1}^{\infty}(B_{p+1}D_{p+1} - B_pD_p)$ is convergent if $\sum_{p=1}^{\infty}\sqrt{m_p - m_{p+1}} < +\infty$. Therefore, $\lim_{N\to\infty} B_ND_N$ exists. Similarly,

$$\left(\|\sum_{p=1}^{N}(X_{p+1} - X_p)\|_F\right) \leq \sum_{p=1}^{N}\left(\|X_{p+1} - X_p\|_F\right) \leq \frac{1}{\sqrt{\lambda}}\sum_{p=1}^{N}\sqrt{m_p - m_{p+1}},$$

which implies $\sum_{p=1}^{N}(X_{p+1} - X_p)$ is convergent if $\frac{1}{\sqrt{\lambda}}\sum_{p=1}^{N}\sqrt{m_p - m_{p+1}} < \infty$. Therefore, we conclude $\lim_{p\to\infty}(X_1)_p$ exists.

Further, $\lim_{p\to\infty}\|B_{p+1}D_{p+1} - B_{p+1}D_p\|_F^2 = 0$, since $\{m_p\}_{p=1}^{\infty}$ converges. Therefore $\lim_{p\to\infty} B_{p+1}D_p$ exists and is equal to $\lim_{p\to\infty} B_pD_p = L^*$. This completes the proof. ■

From Theorem 38, we can only prove the convergence of the sequence $\{B_pD_p\}$ but not of $\{B_p\}$ and $\{D_p\}$ separately. We next establish the convergence of $\{B_p\}$ and $\{D_p\}$ with stronger assumption. Consider the situation when

$$\sum_{p=1}^{\infty} \sqrt{m_p - m_{p+1}} < +\infty. \tag{4.53}$$

Theorem 39. *Assume (4.53) holds.*

(i) If B_p is of full rank and $B_p^T B_p \geq \gamma I_{r-k}$ for large p and some $\gamma > 0$ then $\lim_{p\to\infty} D_p$ exists.

(ii) If D_p is of full rank and $D_p D_p^T \geq \delta I_{r-k}$ for large p and some $\delta > 0$ then $\lim_{p\to\infty} B_p$ exists.

(iii) If $X_1^* := \lim_{p\to\infty} (X_1)_p$ is of full rank, then $C^* := \lim_{p\to\infty} C_p$ exists. Furthermore, if we write $L^* = B^*D^*$, for $B^* \in \mathbb{R}^{m\times(r-k)}, D^* \in \mathbb{R}^{(r-k)\times(n-k)}$, then (X_1^*, C^*, B^*, D^*) will be a stationary point of F.

Proof: *(i).* Using (4.48) we have, for $N > 0$,

$$\begin{aligned}\sum_{p=1}^{N} \sqrt{m_p - m_{p+1}} &\geq \sum_{p=1}^{N} \|B_{p+1}(D_p - D_{p+1})\|_F \\ &= \sum_{p=1}^{N} \sqrt{\mathrm{tr}[(D_p - D_{p+1})^T B_{p+1}^T B_{p+1}(D_p - D_{p+1})]},\end{aligned}$$

where $\mathrm{tr}(X)$ denotes the trace of the matrix X. Note that, $B_p^T B_p \geq \gamma I_{r-k}$, and we obtain

$$\sum_{p=1}^{N} \sqrt{m_p - m_{p+1}} \geq \sqrt{\gamma} \sum_{p=1}^{N} \|D_p - D_{p+1}\|_F.$$

Therefore, for $N > 0$,

$$\sqrt{\gamma} \| \sum_{p=1}^{N} (D_p - D_{p+1})\|_F \leq \sqrt{\gamma} \sum_{p=1}^{N} \|D_p - D_{p+1}\|_F \leq \sum_{p=1}^{N} \sqrt{m_p - m_{p+1}},$$

which implies $\sum_{p=1}^{\infty}(D_p - D_{p+1})$ is convergent if (4.53) holds. Hence $\lim_{N\to\infty} D_N$ exists. Similarly we can prove (ii).

(iii). Note that, from (4.48) we have, for $N > 0$,

$$\begin{aligned}\sum_{p=1}^{N} \sqrt{m_p - m_{p+1}} &\geq \sum_{p=1}^{N} \|(X_1)_{p+1}(C_p - C_{p+1})\|_F \\ &= \sum_{p=1}^{N} \sqrt{\mathrm{tr}[(C_p - C_{p+1})^T (X_1)_{p+1}^T (X_1)_{p+1}(C_p - C_{p+1})]}.\end{aligned}$$

If $X_1^* := \lim_{p\to\infty} (X_1)_p$ is of full rank, it follows that, for large p, $(X_1)_{p+1}^T (X_1)_{p+1} \geq \eta I_k$, for some $\eta > 0$. Therefore, we have

$$\sum_{p=1}^{N} \sqrt{m_p - m_{p+1}} \geq \sqrt{\eta} \sum_{p=1}^{N} \|C_p - C_{p+1}\|_F.$$

Following the same argument as in the previous proof, we have, for $N > 0$,

$$\sqrt{\eta} \| \sum_{p=1}^{N} (C_p - C_{p+1})\|_F \leq \sqrt{\eta} \sum_{p=1}^{N} \|C_p - C_{p+1}\|_F \leq \sum_{p=1}^{N} \sqrt{m_p - m_{p+1}},$$

which implies $\sum_{p=1}^{\infty} (C_p - C_{p+1})$ is convergent if (4.53) holds. Finally, we can conclude $\lim_{p\to\infty} C_p = C^*$ exists if (4.53) holds. Recall from (4.44-4.47), we have,

$$\begin{cases} ((X_1)_{p+1} - A_1) \odot W_1 \odot W_1 - (A_2 - (X_1)_{p+1} C_p - B_p D_p) C_p^T = 0, \\ (X_1)_{p+1}^T (A_2 - (X_1)_{p+1} C_{p+1} - B_p D_p) = 0, \\ (A_2 - (X_1)_{p+1} C_{p+1} - B_{p+1} D_p) D_p^T = 0, \\ B_{p+1}^T (A_2 - (X_1)_{p+1} C_{p+1} - B_{p+1} D_{p+1}) = 0. \end{cases}$$

Taking limit $p \to \infty$ in above we have

$$\begin{cases} \frac{\partial}{\partial X_1} F(X_1^*, C^*, B^*, D^*) &= (X_1^* - A_1) \odot W_1 \odot W_1 + (B^* D^* + X_1^* C^* - A_2) C^{*T} = 0, \\ \frac{\partial}{\partial C} F(X_1^*, C^*, B^*, D^*) &= X_1^{*T} (A_2 - X_1^* C^* - B^* D^*) = 0, \\ \frac{\partial}{\partial B} F(X_1^*, C^*, B^*, D^*) &= (A_2 - X_1^* C^* - B^* D^*) D^{*T} = 0, \\ \frac{\partial}{\partial D} F(X_1^*, C^*, B^*, D^*) &= B^{*T} (A_2 - X_1^* C^* - B^* D^*) = 0. \end{cases}$$

Therefore (X_1^*, C^*, B^*, D^*) is a stationary point of F. This completes the proof. ■

4.5 Numerical Results

In this section, we will demonstrate numerical results of our weighted rank constrained algorithm and show the convergence to the solution given by Golub, Hoffman and Stewart when $\lambda \to \infty$ as predicted by our theorems in Section 4.2. All experiments were performed on a computer with 3.1 GHz Intel Core i7-4770S processor and 8GB memory.

4.5.1 Experimental Setup

To perform our numerical simulations we construct two different types of test matrix A. The first series of experiments were performed to demonstrate the convergence of the algorithm proposed in Section 4.4 and to validate the analytical result proposed in Theorem 25. To this end, we performed our experiments on three full rank synthetic matrices A of size 300×300, 500×500, and 700×700 respectively. We constructed A as low rank matrix plus Gaussian noise such that $A = A_0 + \alpha * E_0$, where A_0 is the low-rank matrix, E_0 is the noise matrix, and α controls the noise level. We generate A_0 as a product of two independent full-rank matrices of size $m \times r$ whose elements are independent and identically distributed (i.i.d.) $\mathcal{N}(0,1)$ random variables such that $\mathrm{r}(X_0) = r$. We generate E_0 as a noise matrix whose elements are i.i.d. $\mathcal{N}(0,1)$ random variables as well. In our experiments we choose $\alpha = 0.2 \max_{i,j}(X_{ij})$. The true rank of the test matrices are 10% of their original size but after adding noise they become full rank.

To compare the performance of our algorithm with the existing weighted low-rank approximation algorithms, we are interested in which A has a known singular value distribution. To address this, we construct A of size 50×50 such that $\mathrm{r}(A) = 30$. Note that, A has first 20 singular values distinct, and last 10 singular values repeated. It is natural to consider the cases where A has large and small condition number. That is, we demonstrate the performance comparison of WLR in two different cases: (i) $\frac{\sigma_{max}}{\sigma_{min}}$ is small, and (ii) $\frac{\sigma_{max}}{\sigma_{min}}$ is large, where the condition number of the matrix A is $\kappa(A) = \frac{\sigma_{max}}{\sigma_{min}}$.

4.5.2 Implementation Details

Let $A_{WLR} = (\hat{X}_1^* \quad \hat{X}_1^* C^* + B^* D^*)$ where $(\hat{X}_1^*, C^*, \ B^*, D^*)$ be a solution to (5.2). We denote $(A_{WLR})_p$ as our approximation to A_{WLR} at pth iteration. Recall that $(A_{WLR})_p = ((\hat{X}_1)_p \ \ (\hat{X}_1)_p C_p + B_p D_p)$. We denote $\|(A_{WLR})_{p+1} - (A_{WLR})_p\|_F = Error_p$ and as a measure of the relative error $\frac{Error_p}{\|(A_{WLR})_p\|_F}$ is used. For a threshold $\epsilon > 0$ the stopping criteria of our algorithm at the pth iteration is $Error_p < \epsilon$ or $\frac{Error_p}{\|(A_{WLR})_p\|_F} < \epsilon$ or if it reaches the maximum iteration. The algorithm performs the best when we initialize $\hat{X}_1$ and D as random normal matrices and B and C as zero matrices. Throughout this section we set r as the target low rank and k as the total number of columns we want to constrain in the observation matrix. The algorithm takes approximately 35.9973 seconds on an average to perform 2000 iterations on a 300×300 matrix for fixed r, k, and λ.

4.5.3 Experimental Results on Algorithm in Section 4.4

We first verify our implementation of the algorithm for computing A_{WLR} for fixed weights. We initialize our algorithm by random matrices. Throughout this subsection we set the target low-rank r as the true rank of the test matrix and $k = 0.5r$. To obtain the accurate result we run every experiment 25 times with random initialization and plot the average outcome in each case. A threshold equal to 2.2204×10^{-16} ("machine ϵ") is set for the experiments in this subsection. For Figure 4.3 and 4.4, we consider a nonuniform weight with entries in W_1 randomly chosen from the interval $[\lambda, \zeta]$, where $\min_{\substack{1 \le i \le m \\ 1 \le j \le k}} (W_1)_{ij} = \lambda$ and $\max_{\substack{1 \le i \le m \\ 1 \le j \le k}} (W_1)_{ij} = \zeta$ in the first block W_1 and $W_2 = \mathbb{1}$ and plot iterations versus relative error. Relative error is plotted in logarithmic scale along Y-axis.

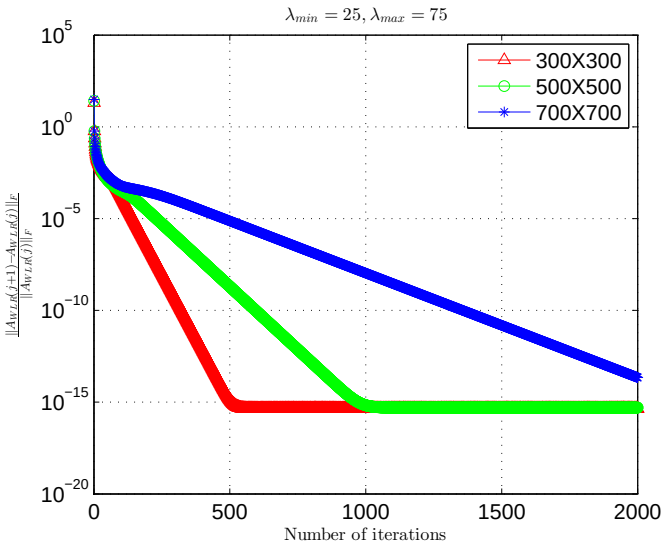

Figure 4.3: Iterations vs Relative error: $\lambda = 25, \zeta = 75$

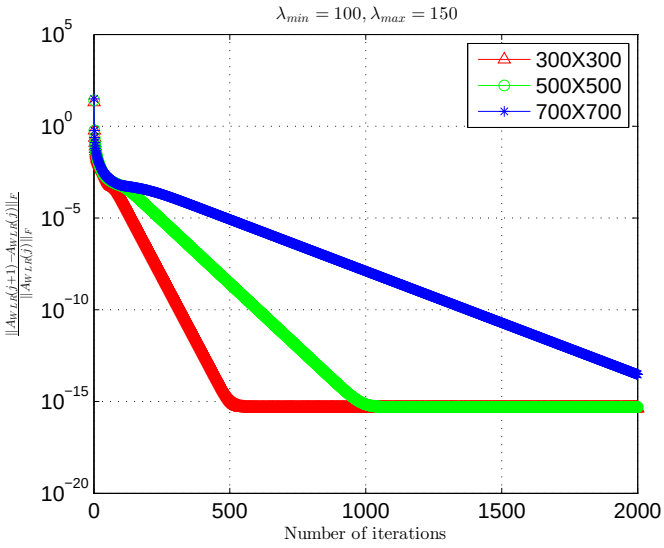

Figure 4.4: Iterations vs Relative error: $\lambda = 100, \zeta = 150$.

Next, we consider a uniform weight in the first block W_1 and $W_2 = \mathbb{1}$. Recall that, in this case the solution to problem (4.6) can be given in closed form by solving (3.4). That is, when $W_1 = \lambda\mathbb{1}$, the rank r solutions to (4.6) are $X_{SVD} = [\frac{1}{\lambda}\tilde{X}_1 \quad \tilde{X}_2]$, where $[\tilde{X}_1 \quad \tilde{X}_2]$ is

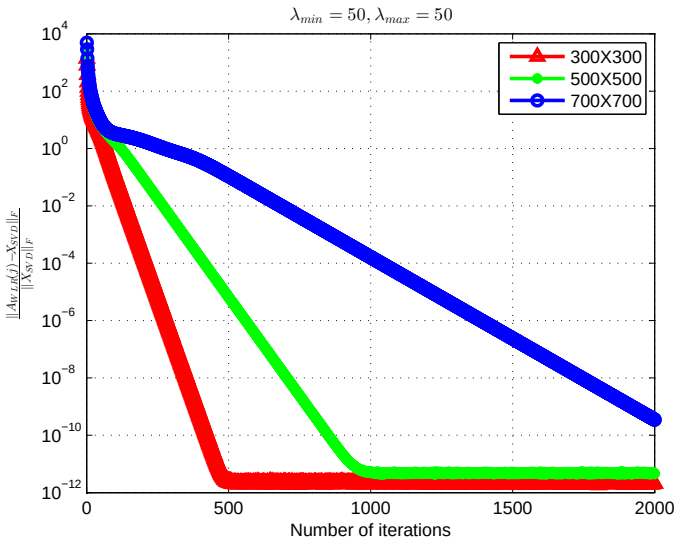

Figure 4.5: Iterations vs $\frac{\|(A_{WLR})_p - X_{SVD}\|_F}{\|X_{SVD}\|_F}$: $\lambda = 50$

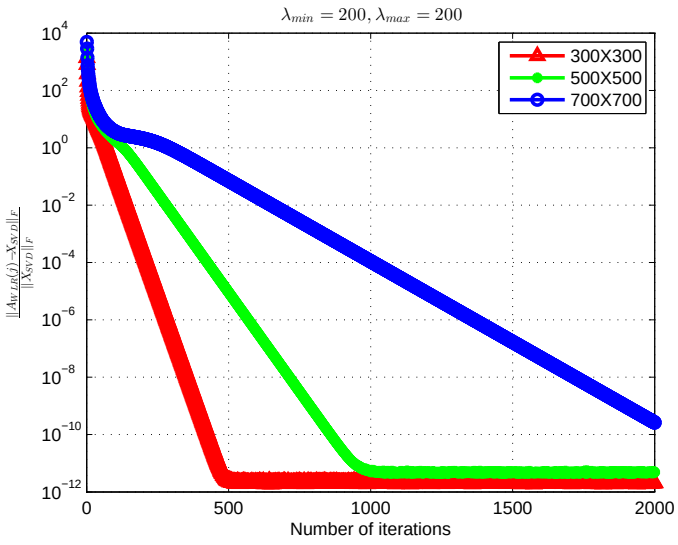

Figure 4.6: Iterations vs $\frac{\|(A_{WLR})_p - X_{SVD}\|_F}{\|X_{SVD}\|_F}$: $\lambda = 200$.

obtained in closed form using a SVD of $[\lambda A_1 \;\; A_2]$. In Figure 4.5 and 4.6, we plot iterations versus $\frac{\|(A_{WLR})_p - X_{SVD}\|_F}{\|X_{SVD}\|_F}$ in logarithmic scale. From Figures 4.3, 4.4, 4.5, and 4.6 it is clear that the algorithm in Section 4.4 converges. Even for the bigger size matrices the iteration count is not very high to achieve the convergence.

4.5.4 Numerical Results Supporting Theorem 25

We now demonstrate numerically the rate of convergence as stated in Theorem 2.1 when the block of weights in W_1 goes to ∞ and $W_2 = \mathbb{1}$. First we use an uniform weight $W_1 = \lambda\mathbb{1}$ and $W_2 = \mathbb{1}$. The algorithm in Section 4.4 is used to compute A_{WLR} and SVD is used for calculating A_G, the solution to (3.1) when $A = (A_1 \;\; A_2)$. We plot λ vs. $\lambda\|A_G - A_{WLR}\|_F$ where $\lambda\|A_G - A_{WLR}\|_F$ is plotted in logarithmic scale along Y-axis. We run our algorithm 20 times with the same initialization and plot the average outcome. A threshold equal to 10^{-7} is set for the experiments in this subsection. For Figure 4.7 and 4.8 we set $\lambda = [1 : 50 : 1000]$.

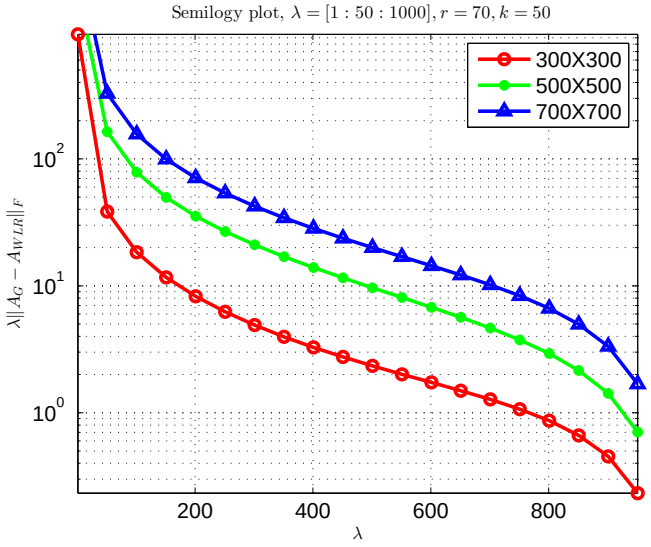

Figure 4.7: λ vs. $\lambda\|A_G - A_{WLR}\|_F$: $(r, k) = (70, 50)$

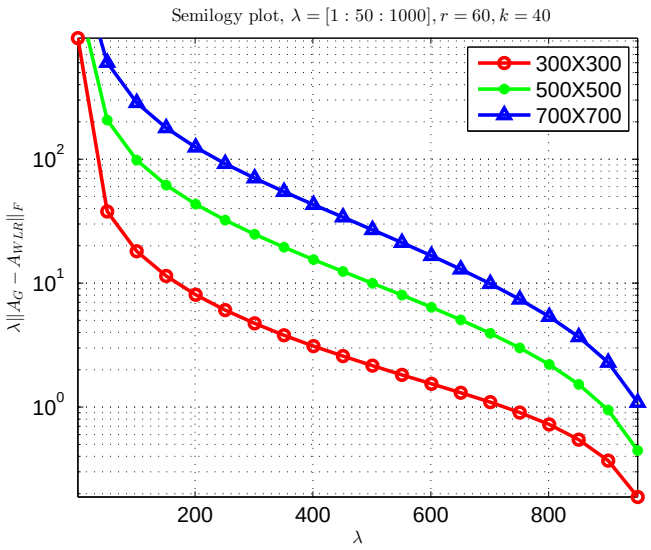

Figure 4.8: λ vs. $\lambda\|A_G - A_{WLR}\|_F$: $(r, k) = (60, 40)$.

The plots indicate for an uniform λ in W_1 the convergence rate is at least $O(\frac{1}{\lambda}), \lambda \to \infty$. Next we consider a nonuniform weight in the first block W_1 and $W_2 = \mathbb{1}$. We consider $\lambda = [2000 : 50 : 3000]$ such that $(W_1)_{ij} \in [2000, 2020], [2050, 2070], \cdots$, and so on. For Figure 4.9 and 4.10, $\lambda\|A_G - A_{WLR}\|_F$ is plotted in regular scale along Y-axis.

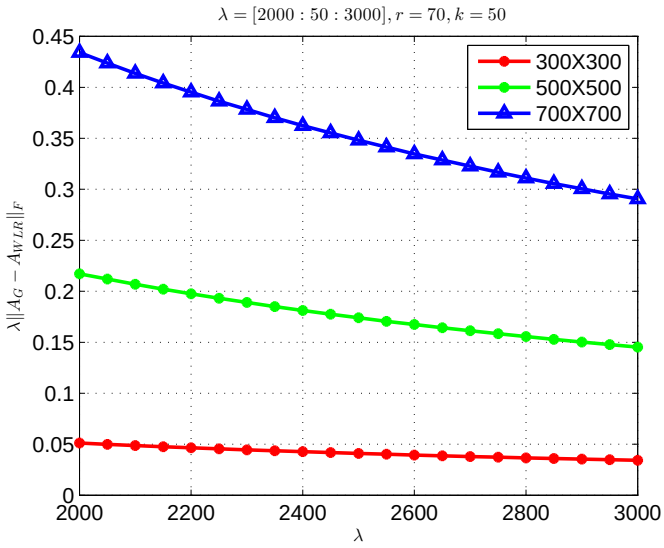

Figure 4.9: λ vs. $\lambda\|A_G - A_{WLR}\|_F$: $(r, k) = (70, 50)$

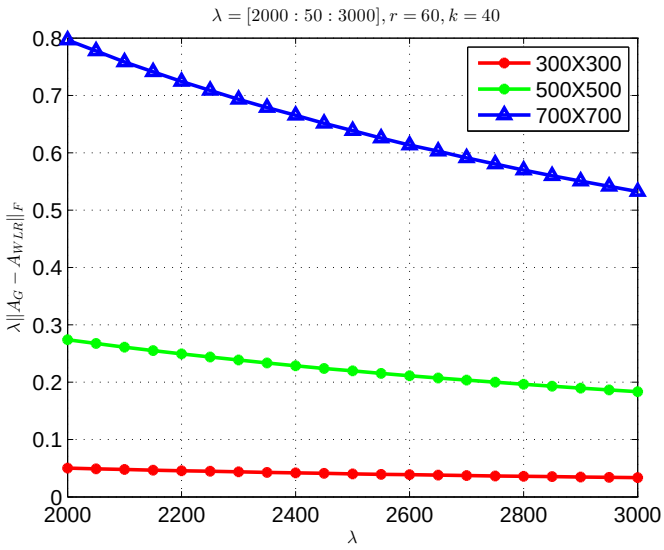

Figure 4.10: λ vs. $\lambda\|A_G - A_{WLR}\|_F$: $(r, k) = (60, 40)$.

The curves in Figure 4.9 and 4.10 are not always strictly decreasing but it is encouraging to see that they stay bounded. Figures 4.7, 4.8, 4.9, and 4.10 provide numerical evidence in supporting Theorem 25. As established in Theorem 25 the above plots demonstrate the convergence rate is at least $O(\frac{1}{\lambda}), \lambda \to \infty$.

4.5.5 Comparison with other State of the Art Algorithms

In this section, we will make an explicit connection of the algorithm proposed in Section 4.4 with the standard weighted total alternating least squares (WTALS) proposed in [23, 37] and expectation maximization (EM) method proposed by Srebro and Jaakkola [39] and compare their performance on synthetic data. We compare the performance of our algorithm with the standard alternating least squares and EM method [23, 39] for $k = 0$ case.

For the numerical experiments in this section, we are interested to see how the distribution of the singular values affects the performance of our algorithm compare to other state-of-the-art algorithms.

Performance Compare to other Weighted Low-Rank Approximation Algorithms We set $(W_1)_{ij} \in [50, 1000]$ and $W_2 = \mathbb{1}$. For WTALS, as specified in the software package, we consider `max_iter = 1000, threshold = 1e-10` [23].For EM, we choose `max_iter = 5000, threshold = 1e-10`, and for WLR, we set `max_iter = 2500, threshold = 1e-16`. As for the performance measure of the algorithms we use the root mean square error (RMSE) which is $\|A - \hat{A}\|_F/\sqrt{mn}$, where $\hat{A} \in \mathbb{R}^{m\times n}$ is the low-rank approximation of A obtained by using different weighted low-rank approximation algorithm. The `MATLAB` code for the EM method is written by the authors following the algorithm proposed in [39]. Note that for computational time of WLR and EM, the authors do not claim the optimized performance of their codes. However, the initialization of X plays a crucial role in promoting convergence of the EM method to a global, or a local minimum, as well as the speed with which convergence is attained. For the EM method, first we rescale the weight matrix to $W_{EM} = \frac{1}{\max_{ij}(W_1)_{ij}}(W_1 \;\; \mathbb{1})$. For a given threshold of weight bound ϵ_{EM}, we initialize X to a zero matrix if $\min_{ij}(W_{EM})_{ij} \leq \epsilon_{W_{EM}}$, otherwise we initialize X to A. Initialization for WLR is same as specified in Section 4.5.2. To obtain the accurate result we run each experiment 10 times and plot the average outcome in each case. Both RMSE and computational time are plotted in logarithmic scale along Y-axis. Figures 4.11, 4.12, 4.13, and 4.14 indicate that WLR is more efficient in handling bigger size matrices than WTALS [23] with the comparable performance measure. This can be attributed by the fact that WTALS uses a weight matrix of size $mn \times mn$ for the given input size $m \times n$, which is both memory and time inefficient. On the other hand, Figures 4.11, 4.12, 4.13, and 4.14 demonstrate the fact that as mentioned in [39], EM-inspired method is computationally effective, however in some cases might converge to a local minimum instead of global.

Performance Comparison for $k = 0$ For $k = 0$ we set the weight matrix as $W = \mathbb{1}$ for all weighted low-rank approximation algorithm. Moreover, we include the classic alternating least squares algorithm to compare between the accuracy of the methods. As specified in

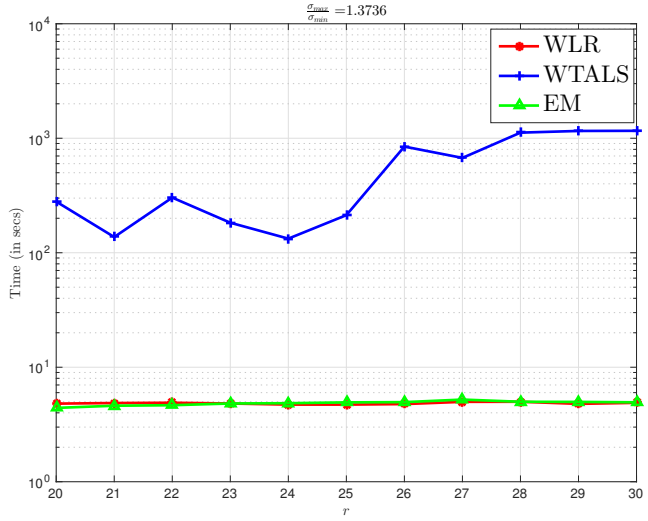

Figure 4.11: Comparison of WLR with other methods: r versus time. We have $\frac{\sigma_{max}}{\sigma_{min}} = 1.3736$, $r = [20 : 1 : 30]$, and $k = 10$.

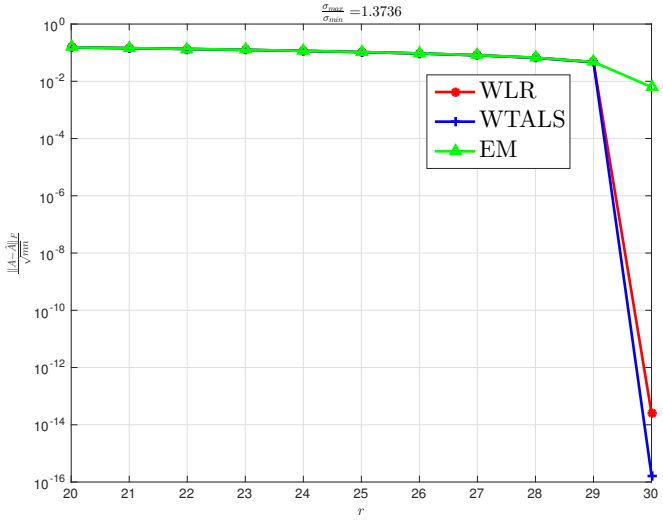

Figure 4.12: Comparison of WLR with other methods: r versus RMSE, $\frac{\sigma_{max}}{\sigma_{min}} = 1.3736$, $r = [20 : 1 : 30]$, and $k = 10$.

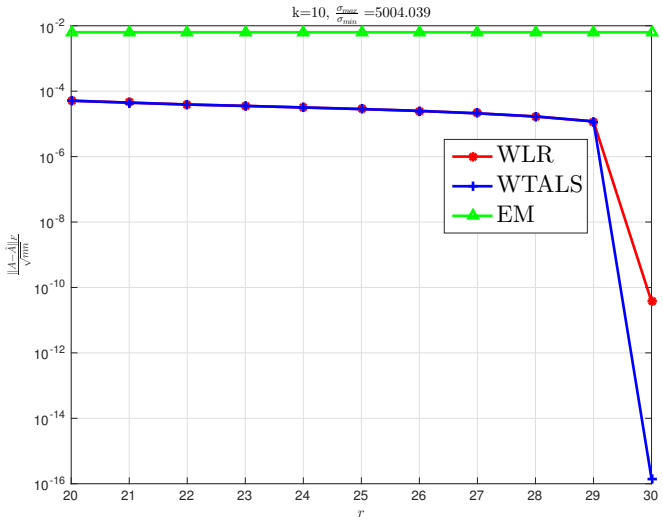

Figure 4.13: Comparison of WLR with other methods: r versus time. We have $\frac{\sigma_{max}}{\sigma_{min}} = 5.004 \times 10^3$, $r = [20 : 1 : 30]$, and $k = 10$.

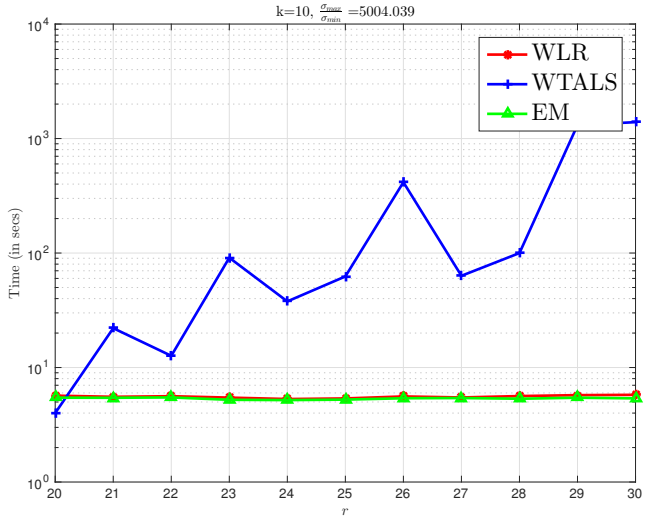

Figure 4.14: Comparison of WLR with other methods: r versus RMSE, $\frac{\sigma_{max}}{\sigma_{min}} = 5.004 \times 10^3$, $r = [20 : 1 : 30]$, and $k = 10$.

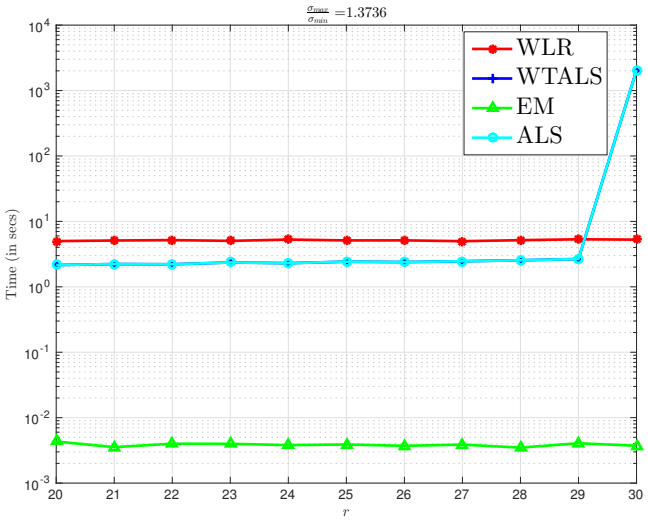

Figure 4.15: Comparison of WLR with other methods: r versus time. We have $\frac{\sigma_{max}}{\sigma_{min}} = 1.3736$, $r = [20 : 1 : 30]$, and $k = 0$.

the previous section, the stopping criterion for all weighted low-rank algorithms are kept the same and RMSE is used for performance measure. We run each experiment 10 times and plot the average outcome in each case. Figure 4.16 and 4.18 indicate that WLR has comparable performance in both cases, $\kappa(A)$ small and large. However from Figure 4.15 and 4.17 we see the standard ALS, WTALS, and EM method is more efficient than WLR, as for $W = \mathbb{1}$ case, each method uses SVD to compute the solution.

Performance Compare to Other Weighted Low-Rank Algorithms for the Limiting Case of Weights As mentioned in our analytical results, one can expect, with appropriate conditions, the solutions to (4.6) will converge and the limit is A_G, the solution to the constrained low-rank approximation problem by Golub-Hoffman-Stewart. We now show the effectiveness of our method compare to other state-of-the-art weighted low rank algorithms when $(W_1)_{ij} \to \infty$, and $W_2 = \mathbb{1}$. SVD is used for calculating A_G, the solution to (3.1), when $A = (A_1 \quad A_2)$, for varying r and fixed k. Considering A_G as the true solution we use the

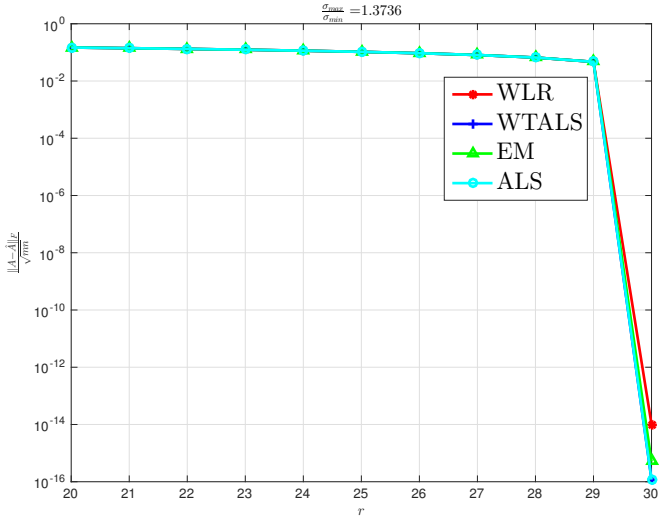

Figure 4.16: Comparison of WLR with other methods: r versus RMSE, $\frac{\sigma_{max}}{\sigma_{min}} = 1.3736$, $r = [20 : 1 : 30]$, and $k = 0$.

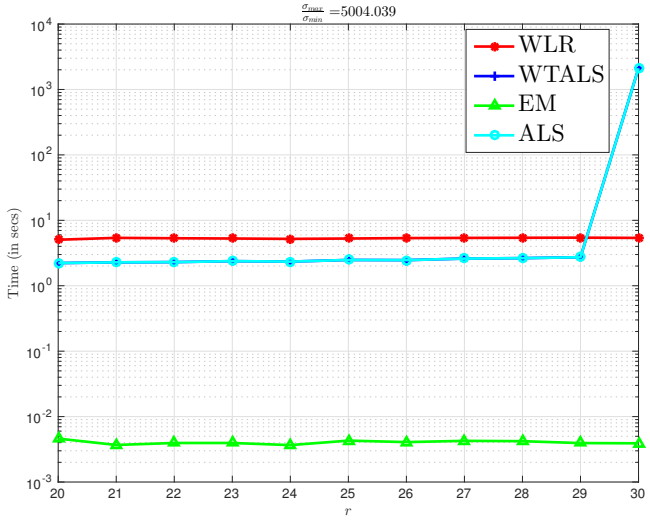

Figure 4.17: Comparison of WLR with other methods: r versus time. We have $\frac{\sigma_{max}}{\sigma_{min}} = 5.004 \times 10^3$, $r = [20 : 1 : 30]$, and $k = 0$.

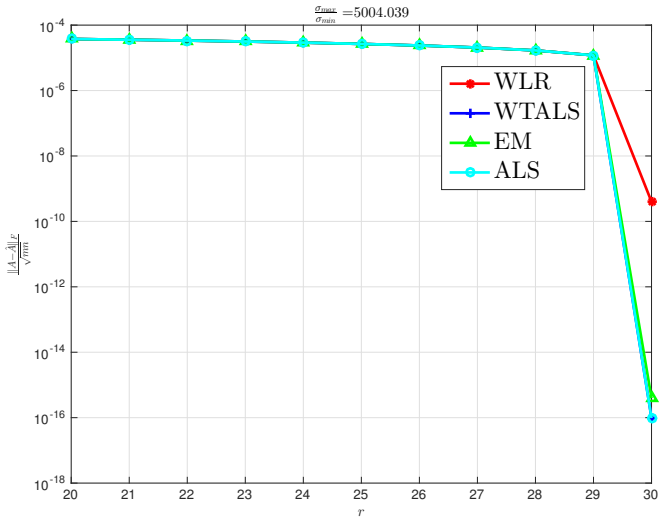

Figure 4.18: Comparison of WLR with other methods: r versus RMSE, $\frac{\sigma_{max}}{\sigma_{min}} = 5.004 \times 10^3$, $r = [20 : 1 : 30]$, and $k = 0$.

RMSE measure $\|A_G - \hat{A}\|_F/\sqrt{mn}$ as the performance measure metric for different algorithms, where $\hat{A} \in \mathbb{R}^{m\times n}$ is the low-rank approximation of A obtained by different weighted low-rank approximation algorithm. From Figure 4.19 and 4.20 it is evident that WLR has the superior performance compare to the other state-of-the-art weighted low-rank approximation algorithms, with computation time being as effective as EM method (see Table 4.1).

To conclude, WLR has comparable or superior performance compare to the state-of-the-art weighted low-rank approximation algorithms for the special case of weight with fairly less computational time. Even when the columns of the given matrix are not constrained, that is $k = 0$, its performance is comparable to the standard ALS. Additionally, WLR and EM method can easily handle bigger size matrices and easier to implement for real world problems (see Section 4.5.6 for detail). On the other hand, WTALS requires more computational time and is not memory efficient to handle large scale data. Another important feature of our algorithm is that it does not assume any particular condition about the matrix

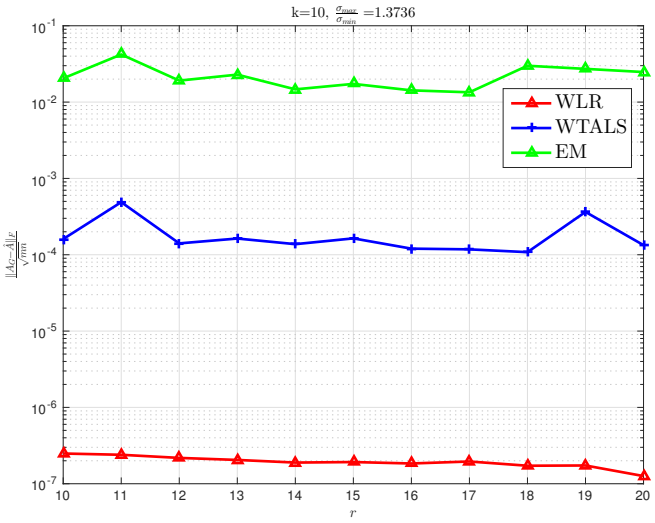

Figure 4.19: r vs $\|A_G - \hat{A}\|_F/\sqrt{mn}$ for different methods, $(W_1)_{ij} \in [500, 1000], W_2 = \mathbb{1}$, $r = 10 : 1 : 20$, and $k = 10$, $\frac{\sigma_{max}}{\sigma_{min}}$ is small.

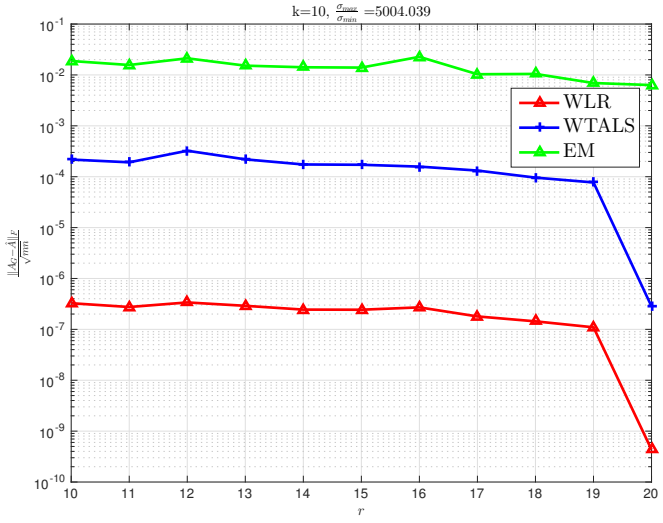

Figure 4.20: r vs $\|A_G - \hat{A}\|_F/\sqrt{mn}$ for different methods, $(W_1)_{ij} \in [500, 1000], W_2 = \mathbb{1}$, $r = 10 : 1 : 20$, and $k = 10$: $\frac{\sigma_{max}}{\sigma_{min}}$ is large.

Table 4.1: Average computation time (in seconds) for each algorithm to converge to A_G

$\kappa(A)$	WLR	EM	WTALS
1.3736	6.5351	6.1454	205.1575
5.004×10^3	8.8271	8.1073	107.0353

A and performs equally well in every occasion.

4.5.6 Background Estimation form Video Sequences [6]

In this section, we will present how our algorithm can be useful in the context of real world problems and handling large scale data matrix. For this purpose, we will demonstrate the qualitative performance of our algorithm on a classic computer vision application: background estimation from video sequences. We use the heuristic that the data matrix A can be considered of containing two blocks A_1 and A_2 such that A_1 mainly contains the information about the background frames and we want to find a low-rank matrix $X = (X_1 \;\; X_2)$ with compatible block partition such that, $X_1 \approx A_1$. In our experiments, we use the Stuttgart synthetic video data set [51]. It is a computer generated video sequence, that comprises both static and dynamic foreground objects and varying illumination in the background. We choose the first 600 frames of the *BASIC* sequence to capture the changing illumination and foreground object. The reader should note that, frame numbers 550 to 600 have static foreground.

Given the sequence of 600 test frames, each frame in the test sequence is resized to 64×80; originally they were 600×800. Each resized frame is stacked as a column vector of size 5120×1 and we form the test matrix A. Next, we use the method described in [3], to choose the set S of correct frame indexes with least foreground movement. In our experiments, for the Stuttgart video sequence, we empirically choose $k = \left\lceil |S|/2 \right\rceil$, where $|S|$ denotes the

cardinality of the set S. We set $r = k + 1$. However, such assumptions do not apply to all practical scenarios.

Algorithm 4: Background Estimation using WLR

Input : $A = (A_1 \;\; A_2) \in \mathbb{R}^{m\times n}$ (the given matrix);
$W = (W_1 \;\; W_2) \in \mathbb{R}^{m\times n}, W_2 = \mathbb{1} \in \mathbb{R}^{m\times(n-k)}$ (the weight), threshold $\epsilon > 0$,
$i_1, i_2 \in \mathbb{N}$;

Run WSVT with $W = I_n$ to obtain: $A = B_{In} + F_{In}$;

Plot image histogram of F_{In} and find threshold ϵ_1;

Set $F_{In}(F_{In} \leq \epsilon_1) = 0$ and $F_{In}(F_{In} > \epsilon_1) = 1$ to obtain a logical matrix LF_{In};

Set $B_{In}(B_{In} \leq \epsilon_1) = 0$ and $B_{In}(B_{In} > \epsilon_1) = 1$ to obtain a logical matrix LB_{In};

Find $\epsilon_2 = \text{mode}(\{\frac{\sum_i (LF_{IN})_{i1}}{\sum_i (LB_{IN})_{i1}}, \frac{\sum_i (LF_{IN})_{i2}}{\sum_i (LB_{IN})_{i2}}, \cdots, \frac{\sum_i (LF_{IN})_{in}}{\sum_i (LB_{IN})_{in}}\})$;

Denote $S = \{i : (\frac{\sum_i (LF_{IN})_{i1}}{\sum_i (LB_{IN})_{i1}}, \frac{\sum_i (LF_{IN})_{i2}}{\sum_i (LB_{IN})_{i2}}, \cdots, \frac{\sum_i (LF_{IN})_{in}}{\sum_i (LB_{IN})_{in}}) \leq \epsilon_2\}$;

Set $k = \left\lceil |S|/i_1 \right\rceil, r = k + i_2$;

Rearrange data: $\tilde{A}_1 = (A(:, i))_{m\times k}$, $i \in S$ randomly chosen and $\tilde{A}_2 = (A(:, i'))_{m\times(n-k)}$, $i \neq i'$;

Apply Algorithm 1 on $\tilde{A} = (\tilde{A}_1 \; \tilde{A}_2)$ to obtain $\tilde{X}$;

Rearrange the columns of $\tilde{X}$ similar to A to find X;

Output : X.

Therefore, we argue that, in practical scenarios, the choices of r and k are problem-dependant and highly heuristic. We rearrange the columns of our original test matrix A as follows: Form $\tilde{A}_1 = (A(:, i))_{m\times k}$ such that $i \in S$ and $1 \leq i \leq k$, and using the remaining columns of the matrix A form the second block $\tilde{A}_2$. With the rearranged matrix $\tilde{A} = (\tilde{A}_1 \;\; \tilde{A}_2)$, we run our algorithm for 200 iterations and obtain a low-rank estimation $\tilde{X}$.

Finally, we rearrange the columns of $\tilde{X}$ as they were in the original matrix A and form X. The algorithm takes approximately 72.5 seconds to run 200 iterations on a matrix of size

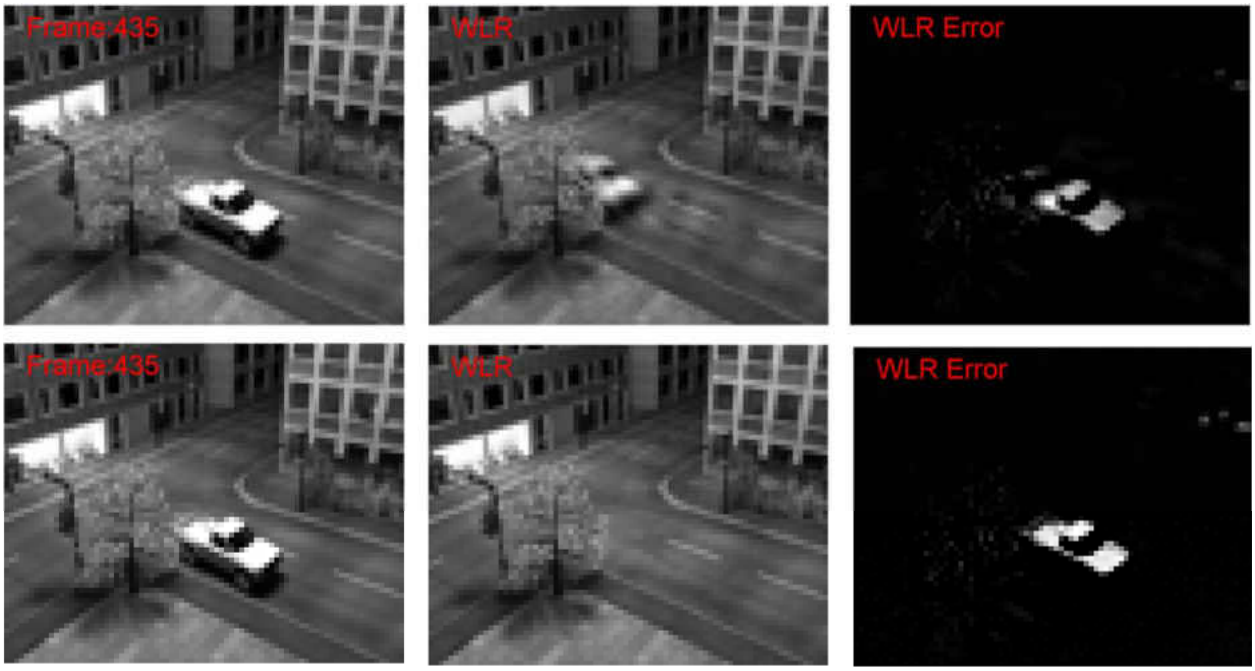

Figure 4.21: Qualitative analysis: On Stuttgart video sequence, frame number 435. From left to right: Original (A), WLR low-rank (X), and WLR error ($A - X$). Top to bottom: For the first experiment we choose $(W_1)_{ij} \in [5, 10]$ and for the second experiment $(W_1)_{ij} \in [500, 1000]$.

5120×600 for a fixed choice of r, k, and W_1. We show the qualitative analysis of our weighted low-rank approximation algorithm in background estimation in Figure 4.21 and 4.22. The results in Figure 4.21 suggest the fact that the choice of weight makes a significant difference in the performance of the algorithm. Indeed, our weighted low-rank algorithm can perform reasonably well in background estimation with proper choice of weight. On the other hand, the experimental result in Next, in In Figure 4.22, we present frame number 210 and 600 of the *Basic* scenario.

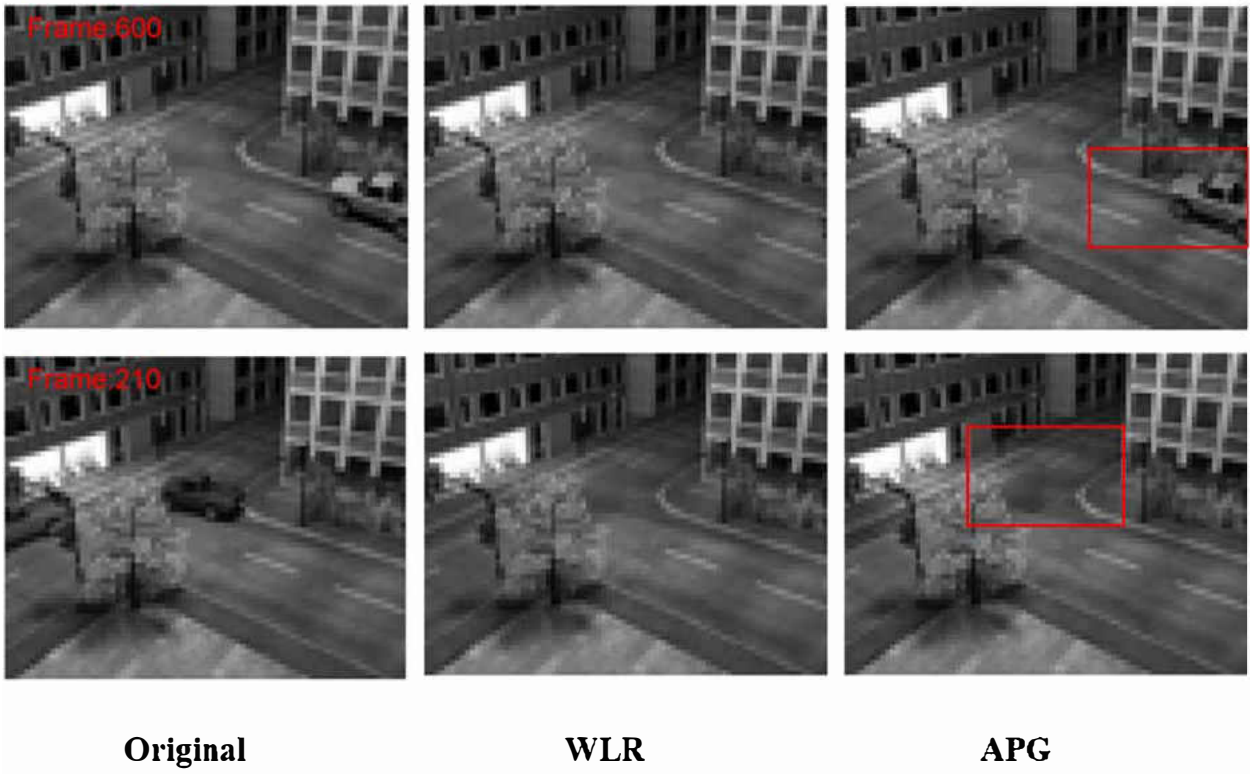

Figure 4.22: Qualitative analysis of the background estimated by WLR and APG on the *Basic* scenario. Frame number 600 has static foreground. APG can not remove the static foreground object from the background. On the other hand, in frame number 210, the low-rank background estimated by APG has still some black patches. In both cases, WLR provides a substantially better background estimation than APG.

The performance of APG on frame 210 is comparable with WLR, but on frame 600 WLR clearly outperforms APG. Even when the foreground is static, with the proper choice of W, r, and k our algorithm can provide a good estimation of the background by removing the static foreground object, in our case the static car at the bottom right corner. On the other hand, the performance of the RPCA algorithms in background estimation when there is static foreground is not good [3, 49].

CHAPTER FIVE: AN ACCELERATED ALGORITHM FOR WEIGHTED LOW RANK MATRIX APPROXIMATION FOR A SPECIAL FAMILY OF WEIGHTS

In Chapter 4, we have verified the limit behavior of the solution to (4.6) when $(W_1)_{ij} \to \infty$ and $W_2 = \mathbb{1}$, the matrix whose entries are equal to 1, both analytically and numerically in [2]. As mentioned in our analytical results, one can expect, with appropriate conditions, the solutions will converge and the limit is A_G, the solution to the constrained low-rank approximation problem by Golub-Hoffman-Stewart. In this chapter we design two numerical algorithms by exploiting an interesting property of the solution to the problem (4.6). Our new algorithms are capable of achieving the desired accuracy faster compare to the algorithm we proposed in [2, 6] when $(W_1)_{ij}$ is large.

The rest of the chapter is organized as follows. In section 5.1, we state an important property of the solution to (4.6) and based on it we propose two accelerated algorithms to solve problem (4.6). Numerical results demonstrating their performance are given in Section 5.2.

5.1 Algorithm [4]

In this section we propose a numerical algorithm to solve (4.6). Recall that (4.6) is a weighted low rank approximation problem which does not have a closed form solution in general [39]. As in [2, 39, 40, 41, 23], our new algorithm is not based on matrix factorization to address the rank constraint. But we exploit the dependence of X_2 on X_1, instead of factoring $X = PQ$. We could take advantage of the special types of weight when $W_2 = \mathbb{1}$ (or even $W_2 \to \mathbb{1}$) to explicitly express X_2 in terms of X_1. We address this property in our next theorem.

Theorem 40. *Assume $r > k$. For $(W_1)_{ij} > 0$ and $(W_2)_{ij} = 1$, if $(\hat{X}_1(W), \hat{X}_2(W))$ is a*

solution to (4.6), then

$$\hat{X}_2(W) = P_{\hat{X}_1(W)}(A_2) + H_{r-k}\left(P^{\perp}_{\hat{X}_1(W)}(A_2)\right).$$

Proof. Note that,

$$\begin{aligned}
&\|(A_1 - \hat{X}_1(W)) \odot W_1\|_F^2 + \|A_2 - \hat{X}_2(W)\|_F^2 \\
=& \min_{\substack{X_1, X_2 \\ \mathrm{r}(X_1 \;\; X_2) \le r}} \left(\|(A_1 - X_1) \odot W_1\|_F^2 + \|A_2 - X_2\|_F^2\right) \\
\le& \|(A_1 - \hat{X}_1(W)) \odot W_1\|_F^2 + \|A_2 - X_2\|_F^2,
\end{aligned}$$

for all $(\hat{X}_1(W) \;\; X_2)$ such that $\mathrm{r}(\hat{X}_1(W) \;\; X_2) \le r$. Therefore,

$$(\hat{X}_1(W) \;\; \hat{X}_2(W)) = \arg \min_{\substack{X_1 = \hat{X}_1(W) \\ \mathrm{r}(X_1 \;\; X_2) \le r}} \|(\hat{X}_1(W) \;\; A_2) - (X_1 \;\; X_2)\|_F^2. \tag{5.1}$$

Therefore, by Theorem 17, $\hat{X}_2(W) = P_{\hat{X}_1(W)}(A_2) + H_{r-k}\left(P^{\perp}_{\hat{X}_1(W)}(A_2)\right)$. $\square$

We will use Theorem 40 to device an iterative process to solve (4.6) for the special case of weight. We assume that $\mathrm{r}(X_1) = k$. Then any X_2 such that $\mathrm{r}(X_1 \;\; X_2) \le r$ can be given in the form

$$X_2 = X_1 C + D,$$

for some arbitrary matrices $C \in \mathbb{R}^{m \times (n-k)}$ and $D \in \mathbb{R}^{m \times (n-k)}$, such that $\mathrm{r}(D) \le r - k$. Therefore, for $W_2 = \mathbb{1}$, (4.6) becomes an constrained weighted low-rank approximation problem:

$$\min_{\substack{X_1, C, D \\ \mathrm{r}(D) \le r-k}} \left(\|(A_1 - X_1) \odot W_1\|_F^2 + \|A_2 - X_1 C - D\|_F^2\right). \tag{5.2}$$

Denote $F(X_1, C, D) = \|(A_1 - X_1) \odot W_1\|_F^2 + \|A_2 - X_1 C - D\|_F^2$ as the objective function. If X_1 has QR decomposition: $X_1 = QR$, then using Theorem 40 we find

$$P_{X_1(W)}(A_2) = QQ^T A_2 = X_1 C,$$

which implies $Q^T A_2 = RC$ and we obtain (assuming X_1 is of full rank)

$$C = R^{-1} Q^T A_2.$$

Next we claim

$$H_{r-k}\left(P^{\perp}_{X_1(W)}(A_2)\right) = D,$$

that is,

$$H_{r-k}((I_m - QQ^T)A_2) = D,$$

which can be shown using the following argument: If $P^{\perp}_{X_1(W)}(A_2)$ has a singular value decomposition $U\Sigma V^T$ then the above expression reduces to

$$H_{r-k}((I_m - QQ^T)A_2) = U\Sigma_{r-k}V^T,$$

To conclude, for a given X_1, we have:

$$C = R^{-1}Q^T A_2 \text{ and } U\Sigma_{r-k}V^T = D,$$

and altogether

$$X_2 = X_1 C + D,$$

such that $\mathrm{r}(D) \leq r - k$. We are only left to find X_1 via the following iterative scheme:

$$(X_1)_{p+1} = \arg\min_{X_1} F(X_1, C_p, D_p). \tag{5.3}$$

We will update X_1 row-wise. Therefore, we will use the notation $X_1(i,:)$ to denote the i-th row of the matrix X_1. We set $\frac{\partial}{\partial X_1} F(X_1, C_p, B_p)|_{X_1=(X_1)_{p+1}} = 0$ and obtain

$$-(A_1 - (X_1)_{p+1}) \odot W_1 \odot W_1 - (A_2 - (X_1)_{p+1}C_p - D_p)C_p^T = 0.$$

Solving the above expression for X_1 sequentially along each row produces

$$(X_1(i,:))_{p+1} = (E(i,:))_p(\mathrm{diag}(W_1^2(i,1) \;\; W_1^2(i,2) \cdots W_1^2(i,k)) + C_p C_p^T)^{-1},$$

where $E_p = A_1 \odot W_1 \odot W_1 + (A_2 - D_p)C_p^T$. Therefore, we have the following algorithm.

Algorithm 5: Accelerated Exact WLR Algorithm

Input : $A = (A_1 \;\; A_2) \in \mathbb{R}^{m\times n}$ (the given matrix); $W = (W_1 \;\; \mathbb{1}) \in \mathbb{R}^{m\times n}$, (the weight); threshold $\epsilon > 0$;

Initialize: $(X_1)_0$;

while *not converged* **do**

$(X_1)_p = Q_p R_p$, $(I_m - Q_p Q_p^T)A_2 = U_p \Sigma_p V_p^T$;

$C_p = R_p^{-1} Q_p^T A_2$;

$D_p = U_p (\Sigma_p)_{r-k} V_p^T$;

$E_p = A_1 \odot W_1 \odot W_1 + (A_2 - D_p)C_p^T$;

$(X_1(i,:))_{p+1} = (E(i,:))_p (\mathrm{diag}(W_1^2(i,1) \;\; W_1^2(i,2) \cdots W_1^2(i,k)) + C_p C_p^T)^{-1}$;

$p = p + 1$;

end

Output : $(X_1)_p, (X_1)_p C_p + D_p$.

Remark 41. *Recall that the update rule for our numerical procedure in Algorithm 5 is*

$$X_{p+1} = ((X_1)_p \;\; (X_1)_p C_p + D_p),$$

such that $\mathrm{r}((X_1)_p) = k$, $\mathrm{r}((X_1)_p C_p) = k$, $\mathrm{r}(D_p) = r - k$, *and* $\max\{\mathrm{r}((X_1)_p C_p + D_p)\} = r$. *We use* $(X_1)_{p+1}$ *to compute* $(X_2)_{p+1}$ *in the next iteration.*

Instead if we use the update rule

$$X_{p+1} = ((X_1)_{p+1} \;\; (X_1)_p C_p + D_p),$$

then $\mathrm{r}((X_1)_{p+1}) = k$, *and* $\mathrm{r}((X_1)_{p+1} C_p) = k$, $\mathrm{r}(D_p) = r - k$.*But we might face a challenge in keeping the rank of* X_{p+1} *less than equal to* r *at the begining, when the entries of* $(W_1)_{ij}$ *are small, and, consequently, the algorithm will take a huge number of iterations to converge. But for larger weight this phenomenon work as a boon. We give the following justification: If*

for a given $\epsilon > 0$, $\|(X_1)_{p+1} - (X_1)_p\| > \epsilon$, *then* $(X_1)_p C_p \notin R((X_1)_{p+1})$*; where* $R(A)$ *denotes the column space of* A*, and as a consequence* $\mathrm{r}(X_{p+1}) = r + k$*. But as* $\|(X_1)_{p+1} - (X_1)_p\| < \epsilon$*, then* $(X_1)_p C_p \in R((X_1)_{p+1})$ *and we obtain* $\mathrm{r}(X_{p+1}) = r$*, as desired.*

Algorithm 6: Accelerated Inexact WLR Algorithm

Input : $A = (A_1 \;\; A_2) \in \mathbb{R}^{m\times n}$ (the given matrix); $W = (W_1 \;\; \mathbb{1}) \in \mathbb{R}^{m\times n}$, (the weight); threshold $\epsilon > 0$;

Initialize: $(X_1)_0$;

while *not converged* **do**

$(X_1)_p = Q_p R_p$, $(I_m - Q_p Q_p^T) A_2 = U_p \Sigma_p V_p^T$;

$C_p = R_p^{-1} Q_p^T A_2$;

$D_p = U_p (\Sigma_p)_{r-k} V_p^T$;

$E_p = A_1 \odot W_1 \odot W_1 + (A_2 - D_p) C_p^T$;

$(X_1(i,:))_{p+1} = (E(i,:))_p (\mathrm{diag}(W_1^2(i,1) \;\; W_1^2(i,2) \cdots W_1^2(i,k)) + C_p C_p^T)^{-1}$;

$p = p + 1$;

end

Output : $(X_1)_{p+1}, (X_1)_p C_p + D_p$.

5.2 Numerical Experiments

In this section, we will demonstrate numerical results of our weighted rank constrained algorithm on synthetic data and show the convergence to the solution given by Golub, Hoffman and Stewart when $\lambda \to \infty$ as proposed by our main results in Chapter 4. The motivations behind performing the numerical experiments were twofold: one is to support the convergence and efficiency of the algorithm, and the second one is to verify the analytical property of the solution from Chapter 4. The choices of r, k, and W are not made purposefully to support any real world example. All experiments were performed on a computer with 3.1 GHz Intel Core i7-4770S processor and 8GB memory.

5.2.1 Experimental Setup

Following the experimental setup in Chapter 4, we construct a full rank matrix A as $A = A_0 + \alpha * E_0$, where A_0 is the low-rank matrix, E_0 is the gaussian noise matrix, and α controls the noise level. In our experiments we choose $\alpha = 0.2 \max_{i,j}(A_{ij})$. The true rank of the test matrices are 10% of their original size but after adding noise they become full rank.

5.2.2 Implementation Details

Throughout this section we set r as the target low rank and k as the total number of columns we want to constrain in the observation matrix. Let $X_{WLR} = (X_1^* \quad X_1^*C^* + D^*)$ where (X_1^*, C^*, D^*) be a solution to (5.2). We denote $(X_{WLR})_p$ as our approximation to X_{WLR} at pth iteration. Recall that $(X_{WLR})_p = ((X_1)_{p+1} \quad (X_1)_p C_p + D_p)$. We denote $\|(X_{WLR})_{p+1} - (X_{WLR})_p\|_F = Error_p$ and use $\frac{Error_p}{\|(X_{WLR})_p\|_F}$ as a measure of the relative error. For a threshold $\epsilon > 0$ the stopping criteria of the exact accelerated WLR algorithm at $(p+1)$th iteration is $Error_p < \epsilon$ or $\frac{Error_p}{\|(X_{WLR})_p\|_F} < \epsilon$ or if it reaches the maximum iteration count. But, for a threshold $\epsilon > 0$ the stopping criteria of the inexact accelerated WLR algorithm at $(p+1)$th iteration is $Error_p < \epsilon$ or $\frac{Error_p}{\|(X_{WLR})_p\|_F} < \epsilon$ or $\mathrm{r}((X_{WLR})_{p+1}) \leq r$ (see Remark 41). For both algorithms we initialize X_1 as a random matrix and a threshold equal to 2.2204×10^{-16} ("machine ϵ") is set to perform all numerical experiments.

5.2.3 Experimental Results on Algorithm 6

We first show the power of the inexact accelerated algorithm in computing X_{WLR} for fixed weights. Throughout this subsection we set the target low-rank r as the true rank of the test matrix and $k = 0.5r$. We initialize our algorithm by random matrices. To obtain the accurate result we run every experiment 25 times with random initialization and plot the average outcome in each case.

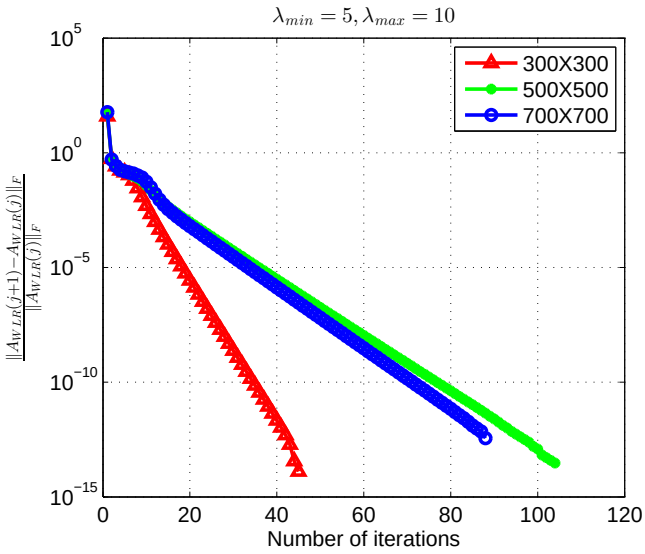

Figure 5.1: Iterations vs Relative error: $\lambda = 5, \zeta = 10$

For Figure 5.1 and 5.2, we consider a nonuniform weight with entries in W_1 randomly chosen from the interval $[\lambda, \zeta]$, where $\min_{\substack{1\le i\le m\\ 1\le j\le k}}(W_1)_{ij} = \lambda$ and $\max_{\substack{1\le i\le m\\ 1\le j\le k}}(W_1)_{ij} = \zeta$ and $W_2 = \mathbb{1}$ and plot iterations versus relative error.

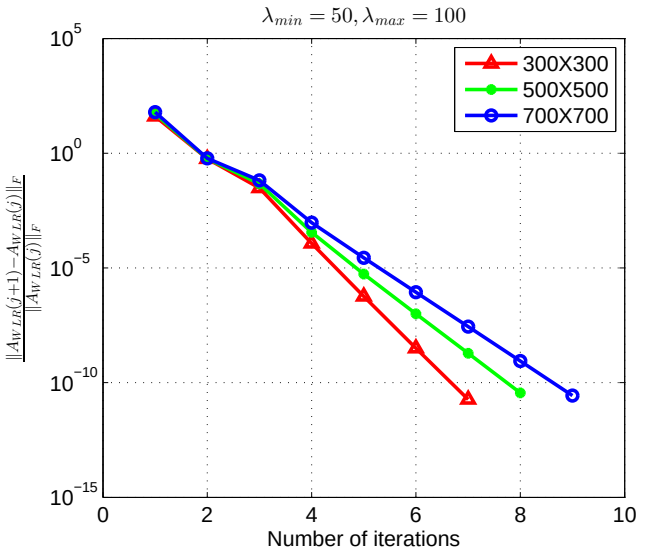

Figure 5.2: Iterations vs Relative error $\lambda = 50, \zeta = 100$.

Relative error is plotted in logarithmic scale along Y-axis. Next, we consider a uniform

weight in the first block W_1 and $W_2 = \mathbb{1}$. Recall that, in this case the solution to problem (4.6) can be given in closed form by solving (3.4).

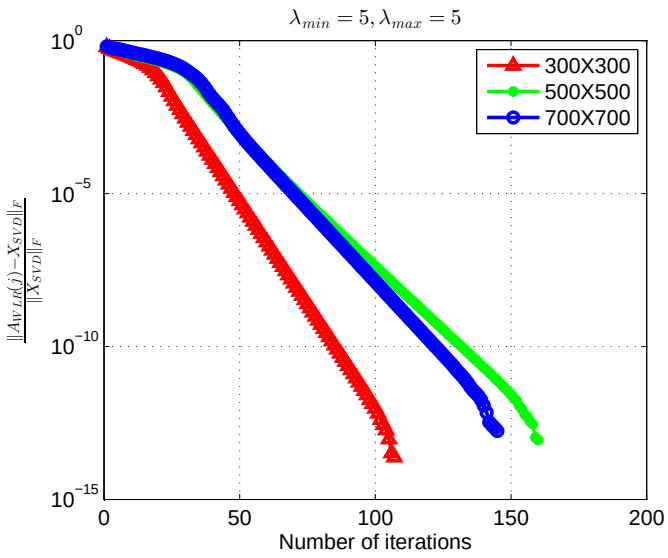

Figure 5.3: Iterations vs $\frac{\|X_{WLR}(p) - X_{SVD}\|_F}{\|X_{SVD}\|_F}$: $\lambda = 5$.

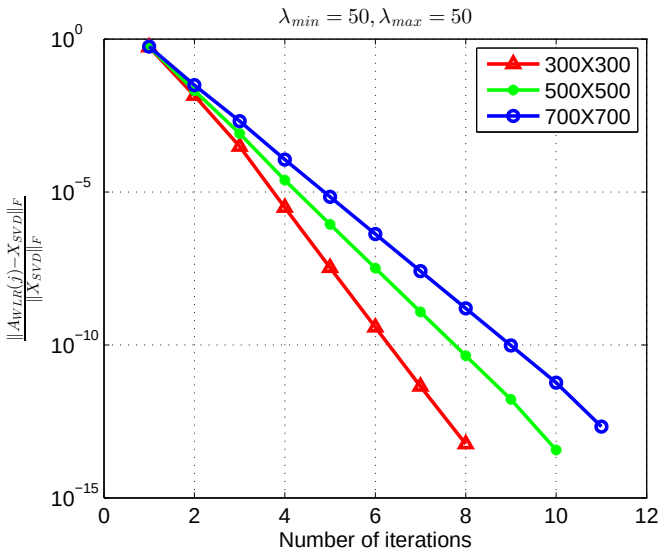

Figure 5.4: Iterations vs $\frac{\|X_{WLR}(p) - X_{SVD}\|_F}{\|X_{SVD}\|_F}$: $\lambda = 50$.

That is, when $W_1 = \lambda\mathbb{1}$, the rank r solutions to (4.6) are $X_{SVD} = [\frac{1}{\lambda}\tilde{X}_1 \quad \tilde{X}_2]$, where $[\tilde{X}_1 \quad \tilde{X}_2]$ is obtained in closed form using a SVD of $[\lambda A_1 \quad A_2]$. In Figure 5.3 and 5.4,

we plot iterations versus $\frac{\|A_{WLR(p)}-X_{SVD}\|_F}{\|X_{SVD}\|_F}$ in logarithmic scale. From Figures 5.1-5.4 it is clear that the inexact accelerated WLR algorithm in Section 5.1 converges. Even for bigger size matrices the iteration count is not very high to achieve the convergence. As claimed in Remark 41, it is clear from Figures 5.1,5.2,5.3, and 5.4, inexact accelerated WLR takes almost 1/10 of iterations when the weights in the first block increase. Hence for bigger weights in W_1, the algorithm takes significantly less time to converge.

5.2.4 Comparison between WLR, Exact Accelerated WLR, and Inexact Accelerated WLR

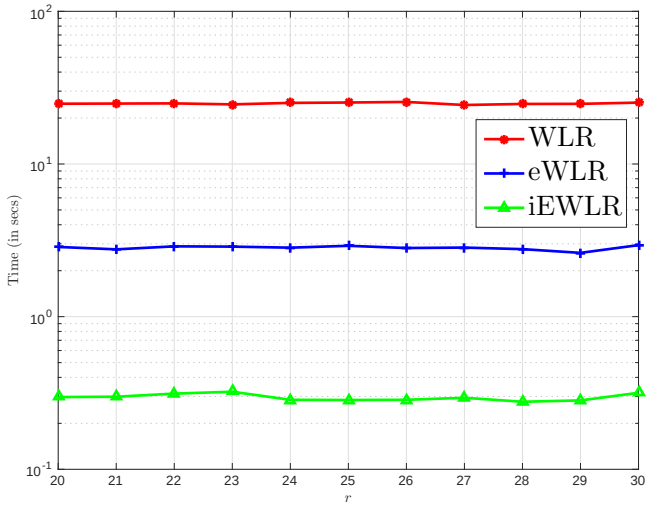

Figure 5.5: Rank vs. computational time (in seconds) for different algorithms. Inexact accelerated WLR takes the least computational time.

In this section, we compare the performance of WLR, exact accelerated WLR, and inexact accelerated WLR on a full rank synthetic test matrix of size 300×300. For the performance measure of the algorithms, we use the root mean square error (RMSE) which is $\|A-\hat{A}\|_F/\sqrt{mn}$, where $\hat{A}\in\mathbb{R}^{m\times n}$ is the low-rank approximation of A obtained by using different weighted

low-rank approximation algorithm. We set $r = 20 : 1 : 30$, $k = 10, \lambda = 50, \zeta = 1000$, and to obtain the accurate result we run every experiment 10 times with random initialization and plot the average outcome in each case. We set the number of iterations for WLR and exact accelerated WLR as 2500 and 100, respectively.

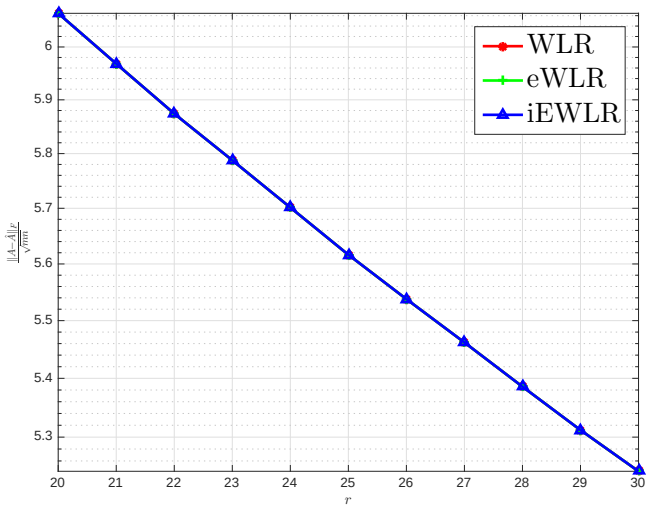

Figure 5.6: Rank vs. RMSE for different algorithms.
All three algorithms have same precision.

From Figure 5.5 and 5.6, we can conclude both exact and inexact accelerated WLR algorithms can recover a low rank matrix as precisely as the regular WLR algorithm in significantly less time.

5.2.5 Numerical Results Supporting Theorem 25

Finally, we numerically demonstrate the rate of convergence as stated in Theorem 25 when the block of weights in W_1 goes to ∞ and $W_2 = \mathbb{1}$. First we use an uniform weight $W_1 = \lambda\mathbb{1}$ and $W_2 = \mathbb{1}$. We use inexact accelerated WLR algorithm to compute A_{WLR} and SVD is used for calculating A_G, the solution to (3.1) when $A = (A_1 \;\; A_2)$. We plot λ vs. $\lambda\|A_G - A_{WLR}\|_F$ where $\lambda\|A_G - A_{WLR}\|_F$ is plotted in logarithmic scale along Y-axis. We run our algorithm

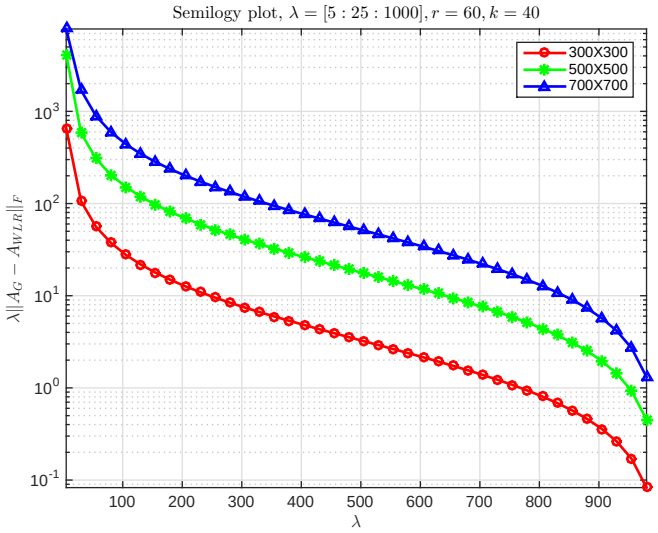

Figure 5.7: λ vs. $\lambda\|A_G - A_{WLR}\|_F$: Uniform λ in the first block, $(r, k) = (60, 40)$.

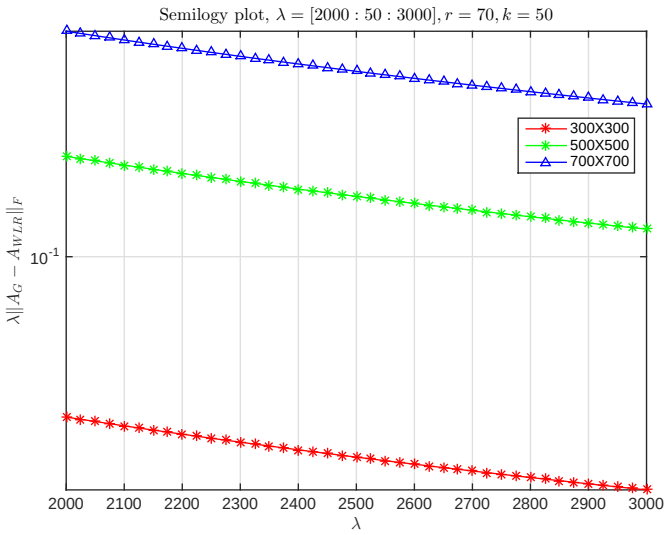

Figure 5.8: λ vs. $\lambda\|A_G - A_{WLR}\|_F$: non-uniform λ in the first block, $(r, k) = (70, 50)$.

25 times with the same initialization and plot the average outcome. For Figure 5.7 we set $\lambda = [5 : 25 : 1000]$. For Figure 5.8, we consider a nonuniform weight in the first block W_1 and $W_2 = \mathbb{1}$. We consider $\lambda = [2000 : 25 : 3000]$ such that $(W_1)_{ij} \in [2000, 2010], [2025, 2035]$ and so on.

Figure 5.7 and 5.8 provide numerical evidence in supporting Theorem 25. As established in Theorem 25, for both uniform and nonuniform weights in W_1 and $W_2 = \mathbb{1}$, the above Plots demonstrate the convergence rate is at least $O(\frac{1}{\lambda}), \lambda \to \infty$.

LIST OF REFERENCES

[1] G. H. Golub, A. Hoffman, and G. W. Stewart , *A generalization of the Eckart-Young-Mirsky matrix approximation theorem*, Linear Algebra and its Applications, 88-89 (1987), pp. 317–327.

[2] A. Dutta and X. Li, *On a problem of weighted low-rank approximation of matrices*, SIAM Journal on Matrix Analysis and Applications, 2016, Revision submitted.

[3] A. Dutta, X. Li, B. Gong, and M. Shah, *Weighted Singular Value Thresholding and its Applications in Computer Vision*, Journal of Machine Learning research, 2016, submitted.

[4] A. Dutta and X. Li, *An Accelerated Algorithm for Weighted Low-Rank Matrix Approximation for a Special Family of Weights*, preprint.

[5] T. Boas, A. Dutta, X. Li, K. Mercier, and E. Niderman, *Shrinkage Function and Its Applications in Matrix Approximations,* Electronic Journal of Linear Algebra, 2016, submitted.

[6] A. Dutta and X. Li, *Background Estimation from Video Sequences Using Weighted Low-Rank Approximation of Matrices*, IEEE 30th Conference on Computer Vision and Pattern Recognition, 2017, submitted.

[7] O. Oreifej, X. Li, and M. Shah, *Simultaneous Video Stabilization and Moving Object Detection in Turbulence*, IEEE Transaction on Pattern Analysis and Machine Intelligence, 35-2 (2013), pp. 450–462.

[8] I. T. Jolliffee, *Principal Component Analysis*, Second edition, Springer-Verlag, 2002, doi:10.1007/b98835.

[9] Z. Lin, M. Chen, and Y. Ma, *The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices*, arXiv preprint arXiv1009.5055, 2010.

[10] Per-Åke Wedin, *Perturbation bounds in connection with singular value decomposition*, BIT Numerical Mathematics, 12-1(1972), pp. 99–111. doi:10.1007/BF01932678.

[11] C. Eckart and G. Young, *The approximation of one matrix by another of lower rank*, Psychometrika, 1-3 (1936), pp. 211–218. doi:10.1007/BF02288367.

[12] N. Srebro and T. Jaakkola, *Weighted low-rank approximations*, 20th International Conference on Machine Learning (2003), pp. 720–727.

[13] G.W. Stewart, *A second order perturbation expansion for small singular values*, Linear Algebra and its Applications, 56 (1984), pp. 231–235, doi:10.1016/0024-3795(84)90128-9.

[14] C. Davis and W. Kahan, *The rotation of eigenvectors by a perturbation III.*, SIAM Journal on Numerical Analysis, 7 (1970), pp. 1–46.

[15] T. Wiberg, *Computation of principal components when data are missing*, In Proceedings of the Second Symposium of Computational Statistics (1976), pp. 229–336.

[16] N. Srebro, J. D. M. Rennie, and T. S. Jaakola, *Maximum-margin matrix factorization*, In Proc. of Advances in Neural Information Processing Systems, 18 (2005), pp. 1329–1336.

[17] T. Hastie, R. Mazumder, J. Lee, and R. Zadeh, *Matrix completion and low-rank SVD via fast alternating least squares*, arXiv preprint arXiv1410.2596, 2014.

[18] M. Udell, C. Horn, R. Zadeh, and S. Boyd, *Generalized low-rank models*, arXiv preprint arXiv:1410.0342, 2014.

[19] S. Boyd L. and Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[20] J. Hansohm, *Some properties of the normed alternating least squares (ALS) algorithm*, Optimization, 19-5 (1988), pp. 683–691.

[21] A. M. Buchanan and A. W. Fitzgibbon, *Damped Newton algorithms for matrix factorization with missing data*, In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2 (2005), pp. 316–322, doi: 10.1109/CVPR.2005.118.

[22] H. Liu, X. Li, and X. Zheng, *Solving non-negative matrix factorization by alternating least squares with a modified strategy*, Data Mining and Knowledge Discovery, 26-3 (2012), pp. 435–451, doi: 10.1007/s10618-012-0265-y.

[23] I. Markovsky, J. C. Willems, B. De Moor, and S. Van Huffel, *Exact and approximate modeling of linear systems: a behavioral approach*, Number 11 in Monographs on Mathematical Modeling and Computation, SIAM, 2006.

[24] I. Markovsky, *Low-rank approximation: algorithms, implementation, applications*, Communications and Control Engineering. Springer, 2012.

[25] S. Van Huffel and J. Vandewalle, *The total least squares problem: computational aspects and analysis*, Frontiers in Applied Mathematics 9 , SIAM, Philadelphia, 1991.

[26] K. Usevich and I. Markovsky, *Variable projection methods for affinely structured low-rank approximation in weighted 2-norms*, Journal of Computational and Applied Mathematics 272 (2014), pp. 430–448.

[27] G.W. Stewart, *On the asymptotic behavior of scaled singular value and QR decompositions*, Mathematics of Computation, 43-168 (1984), pp. 483–489.

[28] J. H. Manton, R. Mehony, and Y. Hua, *The geometry of weighted low-rank approximations*, IEEE Transactions on Signal Processing, 51-2 (2003), pp. 500–514.

[29] W. S. Lu, S. C. Pei, and P. H. Wang, *Weighted low-rank approximation of general complex matrices and its application in the design of 2-D digital filters*, IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 44-7 (1997), pp.650–655, doi: 10.1109/81.596949.

[30] D. Shpak, *A weighted-leats-squares matrix decomphod with application to the design of 2-D digital filters*, In Proceedings of IEEE 33rd Midwest Symposium on Circuits and Systems, (1990), pp. 1070–1073.

[31] K. Usevich and I. Markovsky, *Optimization on a Grassmann manifold with application to system identification*, Automatica, 50-6 (2014), pp. 1656–1662.

[32] E. J. Candès, X. Li, Y. Ma, and J. Wright, *Robust principal component analysis?*, Journal of the Association for Computing Machinery, 58-3 (2011), pp. 11:1–11:37.

[33] E. J. Candès and Y. Plan, *Matrix completion with noise*, Proceedings of the IEEE, 98-6 (2009), pp. 925–936.

[34] A. L. Chistov and D. Yu. Grigor'ev, *Complexity of quantifier elimination in the theory of algebraically closed fields*, Mathematical Foundations of Computer Science's Lecture Notes in Computer Science, 176 (1984), pp. 17–31.

[35] I. T. Jolliffee, *Principal component analysis*, Second ed., Springer-Verlag, 2002.

[36] A. Edelman, T. A. Arias, S. T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM Journal on Matrix Analysis and Applications, 20 (1998), pp. 303-353.

[37] J. H. Manton, R. Mehony, and Y. Hua, *The geometry of weighted low-rank approximations*, IEEE Transactions on Signal Processing, 51-2 (2003), pp. 500–514.

[38] C. Eckart and G. Young, *The approximation of one matrix by another of lower rank*, Psychometrika, 1-3 (1936), pp. 211–218.

[39] N. S. Srebro and T. S. Jaakkola, *Weighted low-rank approximations*, 20th International Conference on Machine Learning, 2003, pp. 720–727.

[40] T. Okatani and K. Deguchi, *On the Wiberg algorithm for matrix factorization in the presence of missing components*, International Journal of Computer Vision, 72-3 (2007), pp. 329–337.

[41] T. Wiberg, *Computation of principal components when data are missing*, In Proceedings of the Second Symposium of Computational Statistics, 1976, pp. 229–336.

[42] B. Xin, Y. Tian, Y. Wang, and W. Gao, *Background subtraction via generalized fused lasso foreground modeling*, IEEE Computer Vision and Pattern Recognition (2015), pp. 4676–4684.

[43] N. Srebro, J. D. M. Rennie, and T. S. Jaakola, *Maximum-margin matrix factorization*, Advances in Neural Information Processing Systems, 17 (2005), pp. 1329–1336.

[44] M. Tao and X. Yuan, *Recovering low-rank and sparse components of matrices from incomplete and noisy observations*, SIAM Journal on Optimization, 21 (2011), pp. 57–81.

[45] A. M. Buchanan and A. W. Fitzgibbon, *Damped Newton algorithms for matrix factorization with missing data*, IEEE Computer Vision and Pattern Recognition, 2 (2005), pp. 316–322.

[46] G.A. Watson, *Characterization of the subdifferential of some matrix norms*, Linear Algebra and its Applications, 170 (1992), pp. 33– 45.

[47] A. Eriksson and A. v. d. Hengel, *Efficient computation of robust weighted low-rank matrix approximations using the ℓ_1 norm*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 34-9 (2012), pp. 1681–1690.

[48] J. Cai, E. J. Candès, and Z. Shen, *A singular value thresholding algorithm for matrix completion*, *SIAM Journal on Optimization*, 20 (2010), pp. 1956–1982.

[49] J. Wright, Y. Peng, Y. Ma, A. Ganseh, and S. Rao, *Robust principal component analysis: exact recovery of corrputed low-rank matrices by convex optimization*, Advances in Neural Information Processing systems 22, (2009), pp. 2080–2088.

[50] N. Oliver, B. Rosario, and A. Pentland, *A Bayesian Computer Vision System for Modeling Human Interactions*, International Conference on Computer Vision Systems, pp. 255-272.

[51] S. Brutzer, B. Höferlin, and G. Heidemann, *Evaluation of background subtraction techniques for video surveillance*, IEEE Computer Vision and Pattern Recognition (2011), pp. 1937–1944.

[52] P. Lyman, H. Varian, *How much information 2003?*, Technical Report, 2004. Available at http://www2.sims.berkeley.edu/research/projects/how-much-info-2003/printable_report.pdf.

[53] R. Basri and D. Jacobs, *Lambertian reflection and linear subspaces*, IEEE Transaction on Pattern Analysis and Machine Intelligence, 25-3 (2003), pp. 218233.

[54] A. Georghiades, P. Belhumeur, and D. Kriegman, *From few to many: Illumination cone models for face recognition under variable lighting and pose*, IEEE Transaction on Pattern Analysis and Machine Intelligence, 23-6 (2001), pp. 643–660.

[55] G. W. Stewart, *On the early history of the singular value decomposition*, SIAM Review, 35 (1993), pp. 551–566.

[56] G. W. Strang, *Introduction to Linear Algebra*, 3rd ed., Wellesley-Cambridge Press, 1998.

[57] D. L. Donoho and I. M. Johnstone, *Ideal spatial adaptation by wavelet shrinkage*, Biometrika, 81 (1994), pp. 425–455.

[58] R. Tibshirani, *Regression shrinkage and selection via the LASSO*, Journal of the Royal statistical society, series B, 58 (1996), pp.267–288.

[59] K. Bryan and T. Leise, *Making do with less: an introduction to compressed sensing*, SIAM Review, 55 (2013), pp. 547–566.

[60] W. Yin, E. Hale, and Y. Zhang, *Fixed-point continuation for l_1-minimization: methodology and convergence*, SIAM Journal on Optimization, 19 (2008), pp. 1107–1130.

[61] E. J. Candès, J. Romberg, and T. Tao, *Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information*, IEEE Transactions on Information Theory, 52 (2006), pp. 489–509.

[62] M. R. Osborne, B. Presnell, and B. A. Turlach, *On the LASSO and its dual*, Journal of Computational and Graphical Statistics, 9 (1999), pp.319–337.

[63] R. J. Tibshirani and J. Taylor, *The solution path of the generalized LASSO*, The Annals of Statistics, 39-3 (2011), pp. 1335–1371.

[64] S. Q. Ma and D. Goldfarb and L. F. Chen, *Fixed point and Bregman iterative methods for matrix rank minimization*, Math. Prog. Ser. A, 2009.

[65] X. Yuan and J. Yang, *Sparse and low-rank matrix decomposition via alternating direction methods*, Technical report available from http://www.optimization-online.org/DBFILE/2009/11/2447.pdf, Department of Mathematics, Hong Kong Baptist University, 2009.

[66] M. Fazel, *Matrix Rank Minimization with Applications*, Ph.D. dissertation, Department of Electrical Engineering, Stanford University, 2002.

[67] T. Okatani, T.Yoshida, and K. Deguchi, *Efficient Algorithm for Low-rank Matrix Factorization with Missing Components and Performance Comparison of Latest Algorithms*, Proceedings of International Conference on Computer Vision(ICCV) 2011, pp. 1–8.

[68] K. Mitra, S. Sheorey, and R. Chellappa, *Large-scale matrix factorization with missing data under additional constraints*, In Proceedings of Advances in Neural Information Processing Systems (NIPS), 2010, pp. 1651–1659.

[69] C. Tomasi and T. Kanade, *Shape and motion from image streams under orthography: a factorization method*, International Journal of Computer Vision, 9-2 (1992), pp. 137–154.

[70] D. Martinec and T. Pajdia, *3d reconstruction by fitting low-rank matrices with missing data*, In Proceedings of Computer Vision and Pattern Recognition, 2005, pp. 198–205.

[71] N. Guilbert, A. Bartoli, and A. Heyden, *Affine approximation for direct batch recovery of euclidian structure and motion from sparse data*, International Journal of Computer Vision, 69 (2006), pp. 317–333.

[72] K. Zhao and Z. Zhang, *Successively alternate least square for low-rank matrix factorization with bounded missing data*, Computer Vision and Image Understanding, 114 (2010), pp. 1084–1096.

[73] Y. Nesterov, *Smooth Minimization of Non-smooth Functions*, Mathematical Programmming, 103-1 (2005), pp. 127–152.

[74] N. S. Aybat, D. Goldfarb, and S. Ma, *Efficient algorithms for robust and stable principal component pursuit problems*, Computational Optimization and Applications, 58-1 (2014), pp. 1–29.

[75] L. Li, W. Huang, I.H. Gu, and Q. Tian, *Statistical modeling of complex backgrounds for foreground object detection*, IEEE Transaction on Image Processing,13-11 (2004), pp. 1459–1472.