Abstract

LANDI, AMANDA KIM. The Nonnegative Matrix Factorization: Methods and Applications. (Under the direction of Kazufumi Ito.)

In today's world, data continues to grow in size and complexity. Thus, big data analysis is an imperative process. A popular approach for big data analysis is the use of low-dimensional matrix representations which reduce data complexity and highlight significant patterns. One technique that has recently gained popularity and success is the Nonnegative Matrix Factorization (NMF). The Nonnegative Matrix Factorization is not an exact factorization, but a decomposition of data into low-rank components and residual components. It is a representation of a data array in the form of two low-rank factor matrices with nonnegative entries. In this thesis, we will investigate the NMF as a data analysis method for the general class of data, extend NMF analysis, and explore new applications.

First, we discuss the NMF as a reduced representation and describe the standard NMF algorithms by Seung and Lee. These are the algorithms that concretized the concept of NMF. However, the standard NMF are slow to converge, and may not reach a desirable solution. We develop an algorithm that finds a better and more accurate representation based on the primal-dual active set method. Second, a significant aspect of the NMF problem is determination of rank for the nonnegative factors. For this purpose, we develop a method that takes advantage of the concept of NMF-singular values, and we compare this method to the statistical Akaike Information Criterion.

In summary, we advance NMF analysis conceptually, algorithmically, and extend to new applications. Particularly, in the case of the convolution, the two factors have the clear roles: convolution kernel and signal. Atoms are the prior information that classify the convolution kernel. For the case of the point-spread function, atoms are the weights that describe the kernel. Using proper atoms, we develop a method for the blind deconvolution based on a NMF representation so that we obtain an estimate of the signal as well as the kernel. In addition,

we examine the triple NMF representation to increase the capability of the bilinear NMF for clustering. We advance the representation by incorporating sparsity on a third factor such that the nonzeros then highlight significant features inferring more meaning on clusters. Furthermore, we address the Principal Component Pursuit problem in terms of the NMF. That is, we develop an NMF method to find the decomposition that separates low-rank components and sparse components from data.

The Nonnegative Matrix Factorization: Methods and Applications

by
Amanda Kim Landi

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

Applied Mathematics

Raleigh, North Carolina

2015

APPROVED BY:

Ernest Stitzinger	Zhilin Li
Lorena Bociu	Kazufumi Ito Chair of Advisory Committee

Biography

Amanda Kim Landi was born and raised in New Rochelle, NY. She earned two Bachelor of Arts, one in Mathematics and the other in English, from North Carolina Wesleyan College in 2009. She began her graduate work at North Carolina State University in 2010, earning her Masters of Science in Mathematics from NCSU in 2012.

Acknowledgements

First, I would like to thank my thesis advisor, Dr. Kazufumi Ito, for his guidance throughout my research. Second, I would like to thank my committee members Dr. Ernest Stitzinger, Dr. Zhilin Li, and Dr. Lorena Bociu for their participation.

I would like to thank my mother, Kim Marie, for always being the strongest woman and the most compassionate person I know. I would like to thank my grandparents, Reverend Richard and Carole Drake, for all the phone calls and reminders to eat. Finally, I would like to thank my remaining family and my friends for their support.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this thesis, we investigate the Nonnegative Matrix Factorization (NMF) as a powerful data analysis method. The Nonnegative Matrix Factorization is a low-rank representation of a data array in the form of two factor matrices with nonnegative elements. The NMF is significant in data analysis because it reveals an underlying basis and its assignment. Due to the nonnegativity, the basis represents localized features and gives the notion of a parts-based representation.

We will give a basic understanding of the NMF as a reduced representation of data, and we will describe the standard NMF algorithms by Seung and Lee. We analyze pros and cons of these algorithms. They are shown to have descent, but in practice they do not give the desired solution. Thus, we attempt to develop an alternative algorithm based on the primal-dual active set method that obtains a better representation. We introduce the concept of the NMF-singular values and develop a method to choose middle rank. We then compare this method to the statistical Akaike Information Criterion.

In addition, we examine the NMF beyond the original formulation. We are interested in the NMF when prior information for the factors is assumed. We discuss the concept of atoms and their role in blind deconvolution, and we develop an NMF method that incorporates the atomic information on the convolution kernel. Next, we advance the concept of the triple NMF by establishing sparsity on a third factor such that the non zeros highlight essential components

in the other two factors. We also study the NMF as a method for finding a decomposition that separates sparse components and low-dimensional components.

1.1 Concept

The Nonnegative Matrix Factorization (NMF) is the representation

$$Y = AP + E. \tag{1.1}$$

Generally, it is assumed $Y \in \mathbb{R}^{n\times m}$ is a matrix of observed data with nonnegative elements which can be represented by $A \in \mathbb{R}^{n\times p}$ and $P \in \mathbb{R}^{p\times m}$, unknown nonnegative matrices, and $E \in \mathbb{R}^{n\times m}$ is the model error. We follow convention and use $Y \approx AP$ when referring to the representation. From this point on, we denote an element-wise nonnegative matrix with the inequality $Y \geq 0$. The NMF problem is formulated as

Problem 1.1. *Given a nonnegative matrix $Y \in \mathbb{R}^{n\times m}$ and a positive integer $p < \min(n, m)$, find nonnegative matrices $A \in \mathbb{R}^{n\times p}$ and $P \in \mathbb{R}^{p\times m}$ to solve*

$$\min_{A,P} \ \frac{1}{2}\,||AP - Y||_F^2, \tag{1.2}$$

$$\textit{subject to } A,\ P \geq 0.$$

Note that the cost function is the Frobenius norm, defined as $||AP-Y||_F^2 = \sum_{i,j} |(AP-Y)_{i,j}|^2$. We will refer to this function as the fidelity. That is, the fidelity function determines how true the representation AP is to the original data Y by some criteria. For the Frobenius norm, the criteria is the distance between AP and Y.

There is another measure for the fidelity known as the generalized Kullback-Leibler divergence:

$$D(Y||AP) = \sum_{i,j}(Y_{i,j}\log\frac{Y_{i,j}}{(AP)_{i,j}} - Y_{i,j} + (AP)_{i,j}), \tag{1.3}$$

Like the Frobenius norm, this is also lower bounded by zero, and vanishes if and only if $Y = AP$. We refer to the generalized Kullback-Leibler as just divergence or divergence criterion from henceforth. It has also been referred to as the I-divergence [76]. The criteria of the divergence is the gap between the information of AP and Y. When Y and AP are densities, i.e. $\sum_{i,j} Y_{i,j} = \sum_{i,j} (AP)_{i,j} = 1$, the divergence reduces to the Kullback-Leibler divergence, a significant information criterion [65]. We discuss this in more detail later.

The defining difference between the NMF representation and others, such as the Singular Value Decomposition and Principal Component Analysis [55, 56, 71], is the constraint of nonnegativity on all elements. The concept is first introduced in [61] as the Positive Matrix Factorization (PMF). The purpose of the PMF was to address the nonnegativity that commonly arises in the physical sciences, e.g. chemical experiment measurements are recorded as nonnegative values. However, the NMF was not solidified until Seung and Lee developed the simple multiplicative updates in [65]. We address these algorithms more in chapter 2.

To understand how the NMF is used in data analysis, we describe the example of molecular pattern analysis [17, 19, 57, 59, 78]. Consider the matrix $Y \in \mathbb{R}^{n \times m}$ where each row represents a different cell and each column represents time-series measurements. If we consider the representation $Y \approx AP$, then we can interpret each row of P as a pattern across time and the columns of A as the assignment of these patterns to the cells. That is, if we rewrite $Y \approx AP$ as $y \approx aP$, where y and a are corresponding rows of Y and A, then we see that each row vector y is approximated by a linear combination of rows of P. Hence, P is a basis (or patterns) for the data in Y and A is the assignment of the basis vectors to each y.

Observe, p is the number of patterns present in P. Since there are fewer basis vectors than data vectors, good factorization is only possible if the basis vectors retrieve significant features in the data. Therefore, p is the complexity that needs to be selected (chapter 4). Also notice, the nonnegativity allows the combination of features to be done in an additive manner. This is significant because by combining features, NMF can produce meaningful themes in P. As a result, NMF has been used in a variety of applications, e.g. sparse coding, classification, spectral

clustering, molecular pattern analysis [5, 17, 19, 22, 34, 58].

1.2 Variational formulation

In order to achieve a desired structure by the NMF method, we incorporate prior information on A and P, such as sparsity in the assignment A, near-orthogonality in the basis P, or smoothness of patterns in P [36, 39, 70]. We formulate a regularized least squares problem. For example,

Problem 1.2. *Given a nonnegative matrix $Y \in \mathbb{R}^{n\times m}$ and a positive integer $p < \min(n, m)$, find nonnegative matrices $A \in \mathbb{R}^{n\times p}$ and $P \in \mathbb{R}^{p\times m}$ to solve*

$$\min_{A,P} J(A, P) = \min_{A,P} \frac{1}{2}||AP - Y||_F^2 + \alpha \sum_{i,k} A_{i,k} + \frac{\beta}{2} \sum_{k,\bar{k}} (PP^T - I)_{k,\bar{k}} \tag{1.4}$$

subject to

$$A,\ P \geq 0,\ PP^T = I,$$

where $\alpha \geq 0$ is the regularization parameter for the sparsity in A and $\beta \geq 0$ is the regularization parameter for the near orthogonality of P. Notice that if $\alpha = 0$ and $\beta = 0$, then (1.4) reduces to the standard formulation (1.2). If $\alpha > 0$ increases, then the sum $\sum_{i,k} A_{i,k}$ should decrease. This is the ℓ_1-norm, where we omit the absolute values because of the nonnegativity property. If $\beta > 0$ increases, then the Frobenius norm $\sum_{k,\bar{k}} (PP^T - I)_{k,\bar{k}}$ is expected to decrease as the off-diagonals of PP^T decrease. This is significant because as the basis, we ask P to reveal individual components for modeling accuracy. We cannot have complete orthogonality because of the nonnegativity constraint. The best we can hope for is to partition the basis vectors. Note here we can omit the square on this term because nonnegativity on P and the constraint $PP^T = I$, which enforces the diagonal entries of PP^T to be 1, prevents negativity. The regularization parameters $\alpha \geq 0$ and $\beta \geq 0$ are chosen so that the fidelity term $\frac{1}{2}||AP - Y||_F^2$ and the regularization terms are balanced [36].

An example of an application that will require near-orthogonality on P is text clustering.

Given a document-by-term matrix Y, the NMF finds a P such that P is a topic-by-term matrix. We ask that P partitions the terms into topics. In addition, the sparsity on A gives a clearer assignment of these clusters for analysis. See chapter 6 for more on clustering.

As stated earlier, the regularization parameters can change depending on prior information. In the case of smoothness in basis, we develop the problem

Problem 1.3. *Given a nonnegative matrix $Y \in \mathbb{R}^{n\times m}$ and a positive integer $p < \min(n, m)$, find nonnegative matrices $A \in \mathbb{R}^{n\times p}$ and $P \in \mathbb{R}^{p\times m}$ to solve*

$$\min_{A,P} J(A, P) = \min_{A,P} \frac{1}{2}||AP - Y||_F^2 + \alpha \sum_{i,k} A_{i,k} + \frac{\gamma}{2} trace(PHP^T) \tag{1.5}$$

subject to

$$A,\, P \geq 0,$$

where $\alpha \geq 0$ is the regularization parameter for the sparsity in A and $\gamma \geq 0$ is the regularization parameter for the smoothness in P. The operator H controls the quadratic variation in the rows of P. For one-dimension, we have

$$H = \begin{bmatrix} 2 & -1 & & 0 \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & 2 \end{bmatrix}.$$

Notice that if $\alpha = 0$ and $\gamma = 0$, then (1.5) reduces to the standard formulation, just as (1.4). If $\gamma > 0$ increases, then the smoothness in the rows of P is increased. The regularization parameters $\alpha \geq 0$ and $\gamma \geq 0$ are chosen so that the fidelity term $\frac{1}{2}||AP - Y||_F^2$ and the regularization terms are balanced [36].

An example application in which smoothness of P is desired rather than partitions is molecular pattern analysis. Consider a dataset[1] containing the measurements of cell activity after the

[1]provided by Elana Fertig, Professor of Oncology at Johns Hopkins University

treatment of a solution imatinib mesylate (IM) [59]. If any of the cells share patterns, then we want to see these patterns in the basis P. We will use smoothness regularization in chapter 5.

While regularization provides structural properties to our A and P, it also improves performance of algorithms through stabilization and faster convergence. We will discuss these details more in chapter 2 and 3.

1.3 Thesis outline

The NMF is a versatile data analysis tool, and we expand NMF analysis in three different aspects: conceptually, algorithmically, and we explore new applications. These points are discussed according to the outline:

- Chapter 2: **The Nonnegative Matrix Factorization.** In this chapter, we analyze properties of the algorithms developed by Seung and Lee and compare their performance. In addition, we discuss the modifications by relaxation and by regularization.

- Chapter 3: **Primal-Dual Active Set Nonnegative Matrix Factorization.** In this chapter, we present a new NMF algorithm that uses a primal-dual active set method, and we compare to the standard NMF to show we get a better solution.

- Chapter 4: **Rank Determination.** In this chapter, we discuss how to choose a proper rank for the low-rank representation. We define the concept of the NMF-singular values in order to develop a method to choose the rank for A and P. We then compare to the statistical Akaike Information Criterion.

- Chapter 5: **Blind Deconvolution, Atoms, and NMF.** In this chapter, we describe the concept of atoms and their role in deconvolution. We develop a blind deconvolution method for the NMF incorporating atomic information for the convolution kernel.

- Chapter 6: **Triple NMF.** In this chapter, we discuss the triple NMF representation ASP. We describe the current application of triple NMF, and we advance this concept

by incorporating sparsity on the third factor S. The nonzeros extract detailed features for the reduced data. Furthermore, we discuss a symmetric NMF as a special case of the triple NMF.

- Chapter 7: **Low-rank and Sparse Component Extraction via Nonnegative Matrix Factorization.** In this chapter, we develop an NMF method to find the decomposition that separates low-rank components and sparse components from data.

- Chapter 8: **Conclusions and Future Work.** In this chapter, we summarize the contributions of this thesis and discuss potential future research.

We mention now that the thesis will use MATLAB language for functions, such as X =rand(n, m) for a random matrix X with uniform distribution size $n \times m$. We do explain functions, but omit explanations for notations such as A' where the apostrophe is Matlab notation for A^T.

Chapter 2

The Nonnegative Matrix Factorization

As indicated in chapter 1, the NMF concept was not popularized until Seung and Lee introduced the multiplicative algorithms in [65] for the two criterion: the Frobenius distance and the generalized Kullback-Leibler divergence. We will call the algorithms by Seung and Lee the standard NMF methods. We describe the element-wise gradient descent based derivation and discuss the properties, pros and cons of the NMF methods. We then compare the two algorithms in terms of performance. Moreover, based on our analysis of the standard NMF, we introduce a relaxation and a regularization to the standard NMF update.

2.1 The Standard NMF Updates

Recall the NMF model

$$Y \approx AP.$$

In [65], there are two sets of methods introduced that find a nonnegative matrix representation. The methods retain nonnegativity by a diagonal scale carefully chosen by Seung and Lee. We go into more detail.

The first algorithm, we call the Frobenius NMF, finds a representation for the Problem 1.1 with cost function (1.2). That is,

$$\min_{A,\,P\geq 0} J(A,P) = \min_{A,\,P\geq 0} \frac{1}{2}\,||AP-Y||_F^2 = \min_{A,\,P\geq 0} \frac{1}{2}\sum_{i,j}|(AP-Y)_{i,j}|^2$$

for the data Y and the representation AP. It is important to state here a solution to the NMF problem always exists. We have by definition, $J(A,P)$ is coercive since for any sequence $\{(A^i,P^i)\}$, $J(A^i,P^i)\to\infty$ as $||P^i||\to\infty$ and $||A^i||\to\infty$. In addition, $J(A,P)$ is continuous. We then have that a minimizer pair $(\bar{A},\bar{P})$ exists [13]. This gives existence of the Nonnegative Matrix Factorization. Note, we do not have uniqueness. For example, let $(\bar{A},\bar{P})$ be a pair such that $Y\approx\bar{A}\bar{P}$. It is also true that $Y\approx\hat{A}\hat{P}$ such that $\hat{A}=\bar{A}Q$ and $\hat{P}=Q^{-1}\bar{P}$ where $Q=\alpha I_{p\times p}$ and α a real, positive number.

Algorithm 2.1 Frobenius NMF

$$\begin{aligned} A &\leftarrow A.*(YP')./(APP') \\ P &\leftarrow P.*(A'Y)./(A'AP) \end{aligned}$$

We can understand the updates as a scaled element-wise gradient descent method in the form

$$A^+_{i,k} = A_{i,k} - \eta_{i,k}\frac{\partial J}{\partial A_{i,k}}. \tag{2.1}$$

We first find the gradient of $J(A,P)$ with respect to $A_{i,k}$

$$\frac{\partial J}{\partial A_{i,k}} = \sum_j (AP-Y)_{i,j}P_{k,j},$$

equivalently $\frac{\partial J}{\partial A} = (AP-Y)P' = APP' - YP'$.

We can represent the method (2.1) as the projected gradient method

$$A_{i,k} = \max(0, A_{i,k} - \eta_{i,k} \frac{\partial J}{\partial A_{i,k}}),$$

for any $\eta_{i,k}$, to ensure nonnegativity. Choosing a step-size, in general, is difficult. However, Seung and Lee choose the scale

$$\eta_{i,k} = \frac{A_{i,k}}{(APP')_{i,k}}$$

which preserves nonnegativity. Therefore, the projection to nonnegativity is unnecessary. If we substitute these into (2.1), we have

$$A^+_{i,k} = A_{i,k} - \frac{A_{i,k}}{(APP')_{i,k}} (APP' - YP')_{i,k}.$$

When we distribute, we see the $A_{i,k}$-term cancels. Hence, after simplification, we get

$$A^+_{i,k} = A_{i,k} \frac{(YP')_{i,k}}{(APP')_{i,k}}$$

for each (i, k) element of A^+.

Remark 1. Observe, for each iterate, $A_{i,k}$ will grow when $(YP')_{i,k}$ is larger than $(APP')_{i,k}$. Also, $A_{i,k}$ will get smaller when $(APP')_{i,k}$ is larger than $(YP')_{i,k}$. In addition, notice that if we begin with a positive initialization, $A_{i,k}$ stays positive. Likewise, if we begin with a 0-initialization, $A_{i,k}$ stays 0. □

We can rewrite this in terms of matrices, and we do this in Matlab notation to get the update

$$A \leftarrow A. * (YP')./(APP')$$

where .* represents element-wise multiplication and ./ represents element-wise division. This is exactly the update presented in Algorithm 2.1. We derive the scaled element-wise gradient

descent update for P similarly, where

$$P^{+}_{k,j} = P_{k,j} - \eta_{k,j} \frac{\partial J}{\partial P_{k,j}}.$$

Based on the cost function, we find

$$\frac{\partial J}{\partial P_{k,j}} = \sum_{i} A_{i,k}(AP - Y)_{i,j},$$

or equivalently $\frac{\partial J}{\partial P} = A'(AP - Y) = A'AP - A'Y$. Seung and Lee let

$$\eta_{k,j} = \frac{P_{k,j}}{(A'AP)_{k,j}}.$$

Substitution gives us the P-update

$$P \leftarrow P. * (A'Y)./(A'AP).$$

It is shown that after each update, the cost function monotonically decreases [65]. In summary, this proof follows that of the Expectation-Maximization algorithm [18] in the manner of choosing an auxiliary function that gives the nonincreasing property $J(A, P^{+}) \leq J(A, P)$. We discuss details of this proof in a later section.

Recall there is an alternative problem based on the divergence criterion (1.3). That is,

$$\min_{A,\, P \geq 0} \; J(A, P) = \min_{A,\, P \geq 0} \; \sum_{i,j} (Y_{i,j} \log \left(\frac{Y_{i,j}}{(AP)_{i,j}} \right) - Y_{i,j} + (AP)_{i,j}).$$

Note that the divergence criterion is continuous and coercive. Hence, there exists a minimizer pair $(\bar{A}, \bar{P})$ [13]. Once again, a solution is not unique.

Seung and Lee provide us with Algorithm 2.2. We can also understand the updates in Algorithm 2.2 as a scaled element-wise gradient descent method; we use the update (2.1) for

Algorithm 2.2 divergence NMF

$$A_{i,k} \leftarrow A_{i,k} \frac{\sum_j P_{k,j} \frac{Y_{i,j}}{(AP)_{i,j}}}{\sum_j P_{k,j}}$$

$$P_{k,j} \leftarrow P_{k,j} \frac{\sum_i A_{i,k} \frac{Y_{i,j}}{(AP)_{i,j}}}{\sum_i A_{i,k}}$$

the divergence NMF. From (1.3), we have the partial derivative

$$\frac{\partial J}{\partial A_{i,k}} = \sum_j P_{k,j} \frac{-Y_{i,j}}{(AP)_{i,j}} + \sum_j P_{k,j}.$$

We choose the scaling step-size with the same intent as in the Frobenius NMF derivation. That is, we aim to remove the term $A_{i,k}$ in (2.1) in order to obtain the multiplicative update with nonnegative preservation. Let

$$\eta_{i,k} = \frac{A_{i,k}}{\sum_j P_{k,j}}$$

to obtain

$$A^+_{i,k} = A_{i,k} - \frac{A_{i,k}}{\sum_j P_{k,j}} \left(\sum_j P_{k,j} \frac{-Y_{i,j}}{(AP)_{i,j}} + \sum_j P_{k,j} \right).$$

After simplification, we get

$$A_{i,k} \leftarrow A_{i,k} \frac{\sum_j P_{k,j} \frac{Y_{i,j}}{(AP)_{i,j}}}{\sum_j P_{k,j}}$$

for each (i,k) element of A^+. This is exactly the A-update presented in Algorithm 2.2. We can rewrite this in Matlab notation as

$$A \leftarrow A. * (\, (Y./AP) \, P') ./ (\, \text{repmat}(\, \text{sum}(P, 2)', n, 1) \,),$$

where 'repmat(x, n, 1)' takes $x \in \mathbb{R}^{1 \times m}$ and creates an matrix $X \in \mathbb{R}^{n \times m}$ by repeating x along

n rows and 'sum(P, 2)' is a $p \times 1$ vector of the sums along the rows of P.

We derive the update for P similarly, which can be represented in Matlab notation as

$$P \leftarrow P. * (A'(Y./AP))./(\text{repmat}(\text{sum}(A)', 1, m)).$$

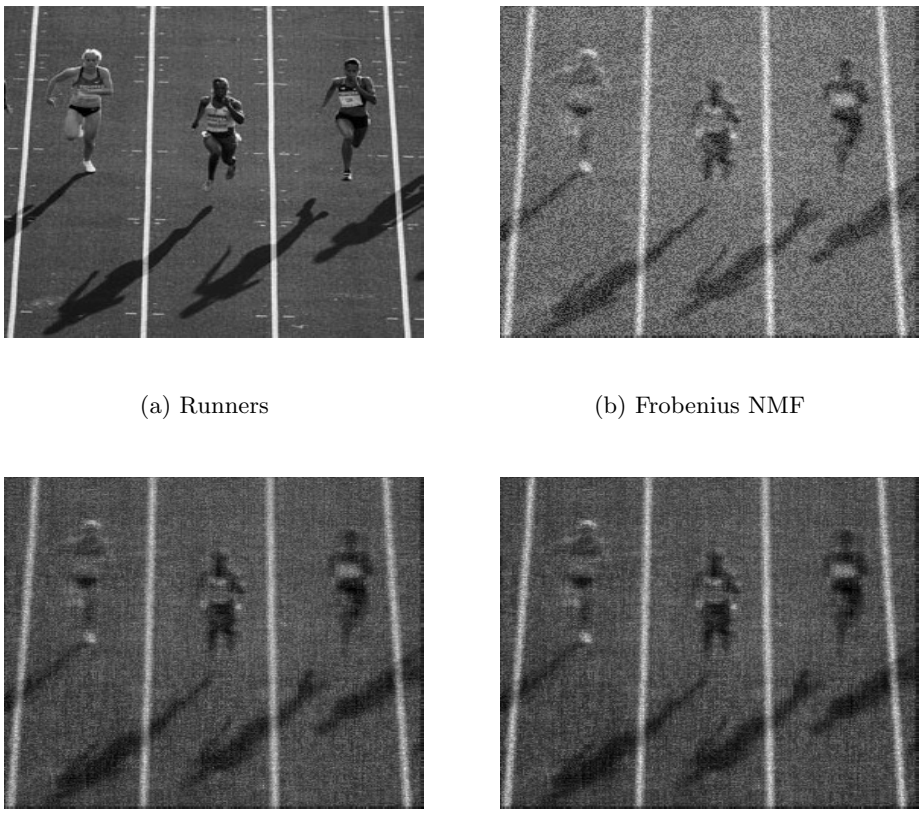
Seung and Lee show these updates are nonincreasing [65]. The proof similarly finds an auxiliary function that gives descent on J. In addition, we make the observation that the divergence NMF preserves density of Y in the density of AP after each update (see section 2.7) thanks to built-in scaling. In chapter 5, we discuss how the built-in scaling benefits in the application of deconvolution.

2.2 Comparison

While both the Frobenius NMF and the divergence NMF find nonnegative representations, the characters of these cost functions are different. For example, minimizing the Frobenius cost function (1.2) results in minimizing the distance between the data Y and the representation AP. On the other hand, the divergence (1.3) is not a distance since it fails symmetry. It determines the gap between the density of Y and the density of AP.

Because of the inherent distinctions in the nature of these criterions, the choice between (1.2) and (1.3) is dependent on the desired outcome in model. However, we will give an objective comparison of the performance of the Algorithms 2.1 and 2.2.

We first compare the divergence NMF to the Frobenius NMF in terms of operation counts. Recall $A \in \mathbb{R}^{n \times p}$ and $P \in \mathbb{R}^{p \times m}$. Examining the matrix operations in Algorithm 2.1, we have three matrix multiplies and two element-wise operations for both the A- and P-updates. For the A-update, the matrix product PP' requires mp^2 operations and the matrix product of A to PP' requires np^2. The final matrix product YP' requires nmp. The two element-wise operations are np each. In totality, the A-update needs $mp^2 + np^2 + nmp + 2np$ operations, i.e. order nmp. We can determine the operation counts for the P-update similarly. In totality, the



(a) Runners

(b) Frobenius NMF

(c) Frobenius NMF

(d) Divergence NMF

Figure 2.1: Divergence NMF v. Frobenius NMF

P-update requires $np^2 + mp^2 + nmp + 2mp$, also order nmp. On the other hand, Algorithm 2.2 has two matrix multiplies, three element-wise operations, and a summation. For the A-update, the matrix product AP requires nmp operations and the product of P' to $(Y./AP)$ requires nmp since $(Y./AP)$ is $n \times m$. The element-wise operations are np each. Finally, the summation requires np operations. In totality, we have $2nmp+4np$. Similarly, the P-update requires a total of $2nmp + 4mp$. Both A- and P-update for divergence NMF have order nmp as well. Hence, the difference in computational time may be miniscule. This is confirmed in Table 2.1, where we see that as p increases, so does the speed of the divergence NMF over the Frobenius NMF. However, we have that the Frobenius NMF is slightly faster more often.

Table 2.1: Divergence NMF v. Frobenius NMF

algorithm	p	iterations	cpu time (in seconds)	$D(Y\|\|AP)$	$\frac{\|\|AP-Y\|\|_F}{\|\|Y\|\|_F}$
divergence NMF	50	50	0.2271	2.2339e5	0.1957
		100	0.4420	1.7634e5	0.1743
	150	50	0.5309	1.9137e5	0.1803
		100	0.9770	1.1977e5	0.1433
	250	50	0.7316	1.7908e5	0.1740
		100	1.4386	9.3393e4	0.1262
Frobenius NMF	50	50	0.1361	2.2578e5	0.1959
		100	0.2665	1.8078e5	0.1751
	150	50	0.4203	1.9546e5	0.1811
		100	0.7585	1.2653e5	0.1449
	250	50	0.7372	1.8430e5	0.1754
		100	1.4440	1.0176e5	0.1290

Consider the example in Figure 2.1b. This is a 276 pixel by 276 pixel image of runners on a track [64]. Often images observed will be corrupted in some manner in real applications. Therefore, we have added a 5×5 local Gaussian blur, standard deviation 3, and 39% relative noise. The separation of the corruption and the original image is known as the deconvolution problem, and we discuss this more in chapter 5.

In Table 2.1, we compare divergence NMF and Frobenius NMF with different number of iterations as well as different middle rank. The purpose is to compare the results for different sized A and P for different iterations. We discuss middle rank more in chapter 4. As one can see, the divergence NMF not only minimizes its own divergence criterion it also minimizes the Frobenius distance.

Remark 2. We have compared for several examples the divergence NMF and the Frobenius NMF for the same quantities as in Table 2.1. If we begin with the same initializations for A and P, we found that divergence NMF minimizes both criterion each time. □

Overall, in this objective situation, the divergence NMF has better performance. However, as we stated before, the algorithm one chooses is dependent on the desired result for the application at hand.

Remark 3. As we've already mentioned, the standard NMF method monotone decreases. However, they are slow. Moreover, if initializations are positive, the algorithm does not find an A and P with 0-entries. Similarly, if initializations have 0-entries, those entries will remain 0. Please see later section on Seung-Lee analysis. □

2.3 Relaxation

In this section, we introduce the relaxation update in Algorithm 2.3. There is a relaxation update for A with relaxation parameter $t \in (0, 2)$, and for P with relaxation parameter $s \in (0, 2)$. These parameters can be different values.

Algorithm 2.3 relaxed NMF

$$a \leftarrow \quad A. * (YP')./(APP'), \quad A \leftarrow \quad ta + (1-t)A$$

$$q \leftarrow \quad P. * (A'Y)./(A'AP), \quad P \leftarrow \quad sq + (1-s)P$$

We found that the relaxation update does not improve the standard NMF. However, when we use the relaxation as a one-sided update, e.g. we know A and want to find only P, we found improvement. This is addressed more in chapter 5 for deconvolution. We also discuss how this update can be used for a symmetric NMF in chapter 6. If $t = 1$, then we have the NMF update. If $t < 1$, then we have an under-relaxation. Moreover, for $t \leq 1$, the method is monotone decreasing. If $t > 1$, gives an over-relaxation update. However, for $t > 1$, nonnegativity is destroyed and we need to project onto nonnegativity.

2.4 Regularization

As mentioned in chapter 1, applications may desire a specific structure of the resulting representation. In addition, regularization improves the performance of the Seung-Lee algorithms in

terms of stability and faster convergence.

It is important enough here to recall that we consider the rows of P an approximating basis for Y and A as the assignment for P. This interpretation is significant in the formulation of least squares problems where we seek specific structure. Recall the two least squares problems in chapter 1. In this section, we will discuss the derivation of the algorithm for cost function (1.4). That is,

$$\min_{A,\,P\geq 0} \; J(A,P) = \min_{A,\,P\geq 0} \; \frac{1}{2}\,||AP-Y||_F^2 + \alpha \sum_{i,k} A_{i,k} + \frac{\beta}{2}\sum_{k,\bar{k}} (PP'-I)_{k,\bar{k}}$$

$$\text{subject to } A,\, P \geq 0,\, PP' = I$$

where $\alpha \geq 0$ is the sparsity parameter of A and $\beta \geq 0$ is the near-orthogonality parameter of P. We remind the reader that we measure the sparsity in A by the ℓ_1-norm, and since we deal with nonnegativity, we remove the absolute value. Furthermore, we measure the near-orthogonality in the rows of P by the Frobenius norm of the off-diagonals of PP' compared to the identity $I_{p\times p}$, where we omit the square on each term because of nonnegativity. As α increases, we expect the ℓ_1-norm to decrease. Similarly, as β increases, we expect the sum of the off-diagonals to decrease.

Algorithm 2.4 Regularized Frobenius NMF

E=ones(p); F=ones(size(A));

$$\begin{aligned} A &\leftarrow A.*(YP')./(APP'+\alpha F), \\ P &\leftarrow P.*(A'Y)./(A'AP+\beta(EP-P)) \\ P &\leftarrow \sqrt{\text{diag}(\,\text{diag}(PP')\,)}^{-1} P \end{aligned}$$

The Algorithm 2.4 can be derived using the element-wise gradient descent (2.1) method for

both A and P. We give you the scaling step-sizes, chosen in the same manner as before,

$$\eta_{A_{i,k}} = \frac{A_{i,k}}{(APP')_{i,k} + \alpha}, \quad \eta_{P_{k,j}} = \frac{P_{k,j}}{(A'AP)_{k,j} + \beta(\sum_{\bar{k}\neq k} P_{\bar{k},j})}$$

and the partial derivatives

$$\frac{\partial J}{\partial A} = (AP - Y)P' + \alpha F, \quad \frac{\partial J}{\partial P} = A'(AP - Y) + \beta(\sum_{\bar{k}\neq k} P_{\bar{k},j}),$$

where F is a matrix of ones size of A. Notice, $\sum_{\bar{k}\neq k} P_{\bar{k},j} = EP - P$, where E is a $p \times p$ matrix of ones. With substitution into (2.1), we obtain the first two steps of Algorithm 2.4 in Matlab language.

Remark 4. Recall earlier we observed for the A-update, if $(APP')_{i,k}$ is greater than $(YP')_{i,k}$, then $A_{i,k}$ will get smaller. Now observe for the regularized A-update that α enhances this reduction and causes $A_{i,k}$ to go to 0 more quickly. Thus, improving convergence. □

In the last step, we consider the case $PP' = I$, or in Matlab notation diag(diag(PP')) $= I$. That is, the L^2-projection of P onto the constraint set $\{P : \text{diag}(\text{diag}(PP')) = I\}$. Note, the function diag$(\cdot)$ takes the diagonal entries of a matrix and makes a vector, or it takes a vector and turns it into a diagonal matrix. We look at diag(diag(PP')) instead of just PP' because our goal is to make the diagonals of PP' equal to 1 and the off-diagonals as close to 0 as possible in order to make P nearly orthogonal. The matrix P is *nearly* orthogonal because the nonnegative constraint prevents complete orthogonality. One can also think of this as partitioning the basis.

We will use this variational formulation in chapter 3 in comparison to a developed primal-dual active set method.

2.5 Conclusion

In summary, we have developed enough basic understanding of the NMF concept and the standard NMF algorithms in order to build upon these details. Furthermore, the adjustments

we've presented may also be extended to the divergence NMF. In the remaining thesis, we will use the ideas presented in this chapter and develop more advanced concepts and methods.

2.6 Analysis of Seung-Lee Proof

In this section, we discuss the convergence property of the standard Frobenius NMF updates in [65]. We then make observations on strict descent.

Recall the Problem 1.1 for $Y \in \mathbb{R}^{n \times m}$

$$\min_{A,\, P \geq 0} \; J(A,P) = \min_{A,\, P \geq 0} \; \frac{1}{2} \, ||AP - Y||_F^2.$$

Notice this cost function is not convex in both A and P, but it is convex (quadratic) in A or in P. Therefore, Seung and Lee use the alternating updates. That is, hold A constant and update P by minimizing $J(A,P)$ over $P \geq 0$.

Let p represent a column of P. Thus, we have the column-wise separated problem

$$\min_{p \geq 0} \; J(p) = \min_{p \geq 0} \; \frac{1}{2} \, ||Ap - y||_F^2$$

where y represents a corresponding column. Let p^i be p at the i-th iteration. We define the function

$$G(p, p^i) = J(p^i) + (p - p^i)' \nabla J(p^i) + \frac{1}{2}(p - p^i)' K(p^i)(p - p^i), \tag{2.2}$$

where

$$K(p^i) = \delta_{k\ell} \frac{(A'Ap^i)_k}{p_k^i} \tag{2.3}$$

and

$$\nabla J(p^i) = A'Ap^i - A'y. \tag{2.4}$$

The function G is an auxilary function for $J(p)$ if it satisfies

$$(a)\ \ G(p,p) = J(p)\ \forall\ p, \quad (b)\ \ G(p,p^i) \geq J(p)\ \forall\ p. \tag{2.5}$$

Given this function, Seung-Lee prove the following theorem:

Theorem 1. *Suppose the function $G(p,p^i)$ is an auxilary function for $J(p) = \frac{1}{2}\,||Ap - y||_F^2$. Then $J(p)$ is nonincreasing under the update*

$$p^{i+1} = \arg\min_p G(p,p^i). \tag{2.6}$$

That is, $J(p^{i+1}) \leq G(p^{i+1},p^i) \leq G(p^i,p^i) = J(p^i)$.

We will show G is an auxilary function. The first condition is simple to show, i.e.

$$G(p,p) = J(p) + (p-p)'\nabla J(p) + \frac{1}{2}(p-p)'K(p)(p-p) = J(p).$$

To show condition (2.5)(b), consider the second Taylor approximation of $J(p)$ at $p = p^i$:

$$J(p) = J(p^i) + (p-p^i)'\nabla J(p) + \frac{1}{2}(p-p^i)'(A'A)(p-p^i).$$

We compare $G(p,p^i)$ to this Taylor approximation:

$$\begin{aligned} G(p,p^i) - J(p) &= \tfrac{1}{2}(p-p^i)'K(p^i)(p-p^i) - \tfrac{1}{2}(p-p^i)'(A'A)(p-p^i) \\ &= \tfrac{1}{2}(p-p^i)'[K(p^i) - A'A](p-p^i). \end{aligned}$$

Condition (2.5)(b) requires $G(p,p^i) - J(p) \geq 0 \implies \frac{1}{2}(p-p^i)'[K(p^i) - A'A](p-p^i) \geq 0$. We

now show $K(p^i) - A'A$ is positive semidefinite; let $\mathbf{0} \neq x \in \mathbb{R}^{p\times 1}$. Thus,

$$\begin{aligned} x'[K(p^i) - A'A]x &= \textstyle\sum_{k\ell} x_k [K(p^i) - A'A]_{k\ell}\, x_\ell \\ &= \textstyle\sum_{k\ell} \left(x_k \left[\frac{\delta_{k\ell}(p^i A'A)_k}{p^i_k} \right] x_\ell - x_k [A'A]_{k\ell} x_\ell \right) \\ &= \textstyle\sum_{k\ell} \left(\left[\frac{p^i_k (A'A)_{k\ell}}{p^i_k} \right] x_k^2 - x_k [A'A]_{k\ell} x_\ell \right) \\ &= \textstyle\sum_{k\ell} (A'A)_{k\ell} \left(\frac{1}{2} x_k^2 + \frac{1}{2} x_\ell^2 - x_k x_\ell \right) \\ &= \textstyle\sum_{k\ell} (A'A)_{k\ell} (x_k - x_\ell)^2 \geq 0. \end{aligned}$$

The function $G(p, p^i)$ does satisfy the conditions, therefore it is an auxilary function of $J(p)$. By Theorem 1, $J(p)$ then has a nonincreasing update. To find the minimum argument of $G(p, p^i)$, we set $\frac{\partial G}{\partial p^i} = 0$. That is,

$$\frac{\partial G}{\partial p^i} = \nabla J(p^i) + (p - p^i) K(p^i) = 0.$$

By this, we obtain the element-wise update

$$p_k^{i+1} = p_k^i - \frac{p_k^i}{(A'Ap^i)_k} ((A'Ap^i)_k - (A'y)_k),$$

which coincides with the standard Frobenius NMF update by Seung and Lee.

Seung and Lee state this is a nonincreasing update. That is,

$$J(p^{i+1}) \leq G(p^{i+1}, p^i) \leq G(p^i, p^i) = J(p^i),$$

where $G(p^i, p^i) = J(p^i)$ comes from the first auxilary function condition and $J(p^{i+1}) \leq G(p^{i+1}, p^i)$ comes from the second auxilary function condition. However, we examine the strict descent. That is,

Corollary 1. *If $G(p, p^i)$ is an auxilary function for $J(p) = \frac{1}{2}||Ap - y||_F^2$, then the strict descent*

$$J(p^{i+1}) < J(p^i)$$

occurs if $G(p^{i+1}, p^i) < G(p^i, p^i) = J(p^i)$.

Observe, from the above proof we have

$$\begin{aligned} J(p^{i+1}) - J(p^i) &= J(p^{i+1}) - G(p^{i+1}, p^i) + G(p^{i+1}, p^i) - G(p^i, p^i) \\ &= -\tfrac{1}{2}x'[K(p^i) - A'A]x + x'\nabla J(p^i) + \tfrac{1}{2}x'K(p^i)x \end{aligned}$$

where $x = p^{i+1} - p^i$. From the update (2.6), we have $\nabla J(p^i) = -K(p^i)x$. Thus,

$$J(p^{i+1}) - J(p^i) = -\frac{1}{2}x'[K(p^i) - A'A]x - \frac{1}{2}x'K(p^i)x.$$

Note, if p^{i+1} and p^i do not differ by a constant, then $x'[K(p^i) - A'A]x > 0$. In addition, $x'K(p^i)x > 0$ if $p_k^i > 0 \ \forall \ k$.

Thus, if we have these assumptions, we have strict descent

$$J(p^{i+1}) < J(p^i)$$

with

$$J(p^{i+1}) + \frac{1}{2}x'[K(p^i) - A'A]x + \frac{1}{2}x'K(p^i)x \leq J(p^i). \tag{2.7}$$

Given this strict descent holds for A and P, we will now show a sequence $\{(A^i, P^i)\}_{i=0}^N$ is bounded, i.e. it has a convergent subsequence (to a local minimum). Note we assume P^0 is not a local minimum.

Let (A^i, P^i) be the estimate at the i-th iterate. We assume

$$\min(\text{diag}(A'A)^i) \geq \hat{k} \quad \text{for } \hat{k} > 0,$$

and

$$\min(\text{diag}(PP')^i) \geq \hat{k} \quad \text{for } \hat{k} > 0.$$

Observe when $k = \ell$,

$$\frac{1}{2}x'[K(p^i) - A'A]x + \frac{1}{2}x'K(p^i)x \geq \frac{1}{2}\hat{k}|x_k|^2.$$

Thus, it follows from (2.7) that

$$J(p^{i+1}) + \frac{1}{2}\hat{k}|p^{i+1} - p^i|^2 \leq J(p^i).$$

Moreover, over all columns and iterations, and given (A^i, P^i), we have

$$J(A^n, P^n) + \sum_{i=1}^{n} \frac{1}{2}\bar{k}\left(||P^i - P^{i-1}||^2 + ||A^i - A^{i-1}||^2\right) \leq J(A^0, P^0)$$

Thus, A^n and P^n are bounded uniformly in n and $||A^n - A^{n-1}|| + ||P^n - P^{n-1}|| \to 0$ uniformly in n. Hence, there exists a subsequence $(A^{n_k}, P^{n_k}) \to (\bar{A}, \bar{P})$.

Next, we show that $(\bar{A}, \bar{P})$ is a minimizer of $J(A, P)$ over nonnegative matrices. We will again use column-separation. Therefore, recall p is a column of P.

As shown in the above analysis, convergence gets slower as i gets larger. Thus, we show the asymptotic complementarity property is satisfied:

$$\bar{p}_j \geq 0 \implies \nabla J(p^{n_k})_j \to 0 \quad \text{and} \quad p_j^{n_k} \to 0 \implies \nabla J(\bar{p})_j \geq 0$$

Let $p_j^{n_k}$ be a subsequence such that

$$p_j^{n_k} \to \bar{p}_j$$

where $\bar{p}_j \geq 0$. Since

$$p_j^{n_k+1} = p_j^{n_k} - \eta\, \nabla J(p^{n_k})_j \tag{2.8}$$

with

$$\eta = \frac{p_j^{n_k}}{(A'Ap^{n_k})_j} \geq \frac{1}{\bar{k}} > 0,$$

we have $\nabla J(p^{n_k})_j \to 0$ and $\nabla J(\bar{p})_j = 0$. Next, suppose

$$p_j^{n_k} \to 0.$$

Let $c = \nabla J(\bar{p})_j < 0$. Then, for a sufficiently large n_k

$$\nabla J(p^{n_k})_j < -\frac{c}{2}.$$

From (2.8)

$$p_j^{n_k+1} - p_j^{n_k} > \frac{c\eta}{2},$$

which contradicts the fact that $p_j^{n_k} \to 0$. Therefore, we have

$$\nabla J(\bar{p})_j \geq 0.$$

In summary, $\bar{p}$ satisfies the asymptotic complementarity condition. Note that as p_j gets smaller, convergence slows. There is a 0-determination strategy such that when $p_j^{n_k}$ gets significantly small we set $p_j^{n_k} = 0$. However, there is no recovery for this element to positivity.

Remark 5. Although we focus on the Frobenius NMF in this section, descent for the divergence NMF is shown by Seung and Lee in [65]. In addition, the analysis on strict descent can be extended to the divergence criterion. □

2.7 Density Preservation of the Divergence NMF

We observe that the divergence NMF preserves at each step $\sum_{i,j}(AP)_{i,j} = \sum_{i,j} Y_{i,j}$, despite the fact that the initialization (A^0, P^0) does not. In Table 2.2, we see this is true. The algorithm then generates minimizing interests for the K-L divergence. Hence, if Y is normalized by $\sum_{i,j} Y_{i,j}$,

then at each step $\sum_{i,j}(AP)_{i,j}$ is also equal to 1. The results in the tables are from the time-series dataset. Observe in the presence of regularization, however, the sum preservation is not exact, but significantly closes the gap compared to the initializations, see Table 2.2.

Table 2.2: Density Preserved and Density Unpreserved

$\sum_{i,j} Y_{i,j}$	α	γ	$\sum_{i,j}(A_0P_0)_{i,j}$	$\sum_{i,j}(AP)_{i,j}$
1	0	0	5.0400e4	1.0000
	0.5	0	1.3342e4	0.8457
	1	0	1.3224e4	0.7438
	0	0.2	1.2368e4	0.7417
	0	0.3	1.0931e4	0.7441
1.7740e4	0	0	1.1367e4	1.7740e4
	0.1	0	1.3301e4	1.7089e4
	1	0	1.4615e4	1.3171e4
	0	0.2	1.1254e4	1.7741e4
	0	0.3	1.3794e4	1.7742e4

To prove the density preservation, recall the divergence cost function (1.3), and the element-wise gradient descent method for P is

$$P^+ = P - \eta_P \frac{\partial J}{\partial P}.$$

If we choose $\eta_{P_{k,j}} = \frac{P_{k,j}}{\sum_i A_{i,k}}$ using the same choice method we did for the A-update. Then,

$$P^+_{k,j} = P_{k,j} - \frac{P_{k,j}}{\sum_i A_{i,k}} \left(\frac{-Y_{i,j}}{(AP)_{i,j}} \sum_i A_{i,k} + \sum_i A_{i,k} \right).$$

Now we multiply $A_{\hat{i},k}$ on the left-hand side of both sides of the equation to get

$$\begin{aligned} A_{\hat{i},k}P^+_{k,j} &= A_{\hat{i},k}P_{k,j} - A_{\hat{i},k}\left(\frac{P_{k,j}}{\sum_i A_{i,k}}\left(\frac{-Y_{i,j}}{(AP)_{i,j}}\sum_i A_{i,k} + \sum_i A_{i,k}\right)\right) \\ &= A_{\hat{i},k}P_{k,j} + A_{\hat{i},k}P_{k,j}\frac{Y_{i,j}}{(AP)_{i,j}} - A_{\hat{i},k}P_{k,j} \\ &= A_{\hat{i},k}P_{k,j}\frac{Y_{i,j}}{(AP)_{i,j}} \end{aligned}$$

Then, we sum over the k's

$$\begin{aligned} (AP^+)_{\hat{i},j} &= \sum_k A_{\hat{i},k}P^+_{k,j} \\ &= \sum_k \left(A_{\hat{i},k}P_{k,j}\frac{Y_{i,j}}{(AP)_{i,j}}\right) \\ &= \left(\sum_k A_{\hat{i},k}P_{k,j}\right)\frac{Y_{i,j}}{(AP)_{i,j}} \\ &= (AP)_{\hat{i},j}\frac{Y_{i,j}}{(AP)_{i,j}} \end{aligned}$$

Summing over both the $\hat{i}$'s and the j's,

$$\begin{aligned} \textstyle\sum_j (AP^+)_{\hat{i},j} &= \textstyle\sum_j \left((AP)_{\hat{i},j}\frac{Y_{i,j}}{(AP)_{i,j}}\right) \\ \textstyle\sum_{\hat{i}}\sum_j (AP^+)_{\hat{i},j} &= \textstyle\sum_{\hat{i}}\sum_j \left((AP)_{\hat{i},j}\frac{Y_{i,j}}{(AP)_{i,j}}\right) \\ \textstyle\sum_{\hat{i},j} (AP^+)_{\hat{i},j} &= \textstyle\sum_{\hat{i},j} Y_{i,j} = \sum_{i,j} Y_{i,j} \end{aligned}$$

Thus, $\sum_{i,j}(AP^+)_{i,j} = \sum_{i,j} Y_{i,j}$ after the P-update. Similarly for the A-update.

Therefore, the NMF by generalized Kullback-Leibler divergence preserves the density from Y in the density of AP.

Chapter 3

The Primal-Dual Active Set Nonnegative Matrix Factorization

The standard NMF methods by Seung and Lee are simple to implement because of the element-wise operations, and they are still used because of their success in finding a nonnegative matrix factorization. However, as we discussed previously, the nature of the multiplicative updates causes slow convergence and may not find a desired solution. In this chapter, we develop an algorithm that finds a better representation by applying the well-established primal-dual active set (PDAS) method to the NMF problem. We then numerically validate the benefit of the PDAS-NMF method over standard NMF method.

3.1 Primal-dual Active Set Method

3.1.1 Unregularized

Recall the regularized least squares problem,

Problem 1.1. *Given a nonnegative matrix* $Y \in \mathbb{R}^{n\times m}$ *and a positive integer* $p < \min(n, m)$,

find nonnegative matrices $A \in \mathbb{R}^{n\times p}$ *and* $P \in \mathbb{R}^{p\times m}$ *to solve*

$$\min_{A,P}\ J(A,P) = \min_{A,P}\ \frac{1}{2}\,||AP - Y||_F^2 + \alpha \sum_{i,k} A_{i,k} + \frac{\beta}{2}\sum_{k,\bar{k}} (PP^{'} - I)_{k,\bar{k}}$$

subject to

$$A,\ P \geq 0,\ PP^{'} = I$$

where $\alpha > 0$ is the regularization parameter for the sparsity in A and $\beta > 0$ is the regularization parameter for the near-orthogonality of P. Recall, if $\alpha > 0$ increases, then the sum $\sum_{i,k} A_{i,k}$ should decrease; if $\beta > 0$ increases, then the sum of off-diagonal entries of PP' decreases. Moreover, the regularization parameters $\alpha \geq 0$ and $\beta \geq 0$ are chosen so that the fidelity term $\frac{1}{2}||AP - Y||_F^2$ and the regularization terms are balanced [36].

In this section, we develop a Newton-like method for solving the unregularized (1.4). To this end, we first state the necessary optimality for the constrained minimization. We define the Lagrangian function

$$L(A, P, \Lambda_1, \Lambda_2) = J(A,P) - \langle \Lambda_1, A\rangle_F - \langle \Lambda_2, P\rangle_F, \quad \Lambda_1,\ \Lambda_2 \geq 0,$$

where $\Lambda_1 \in \mathbb{R}^{n\times p}$ and $\Lambda_2 \in \mathbb{R}^{p\times m}$ are the Lagrange Multipliers for $A \geq 0$ and $P \geq 0$, respectively, and the Frobenius products are defined $\langle \Lambda_1, A\rangle_F = \sum_{i,k}([\Lambda_1]_{i,k} \cdot A_{i,k})$ and $\langle \Lambda_2, P\rangle_F = \sum_{k,j}([\Lambda_2]_{k,j} \cdot P_{k,j})$. Lagrange Multiplier Theory states the necessary optimality is given by

$$\begin{cases} \frac{\partial L}{\partial A} = (AP - Y)P' - \Lambda_1 = 0, \quad \Lambda_1 \geq 0 \text{ and } \langle \Lambda_1, A\rangle_F = 0 \\ \\ \frac{\partial L}{\partial P} = A'(AP - Y) - \Lambda_2 = 0, \quad \Lambda_2 \geq 0 \text{ and } \langle \Lambda_2, P\rangle_F = 0 \end{cases}$$

The complementarity condition in [30,38] then determines the primal-dual active set method by

$$\Lambda_1 = \max(0, \Lambda_1 - cA), \text{ element-wise,}$$

where $c > 0$. Similarly,

$$\Lambda_2 = \max(0, \Lambda_2 - cP), \text{ element-wise.}$$

For the PDAS-NMF, we let $c = 1$. We make the following observations about this condition. First, if $(\Lambda_1 - A)_{i,k} < 0$, then $[\Lambda_1]_{i,k} = 0$ and $A_{i,k} > 0$. Second, if $(\Lambda_1 - A)_{i,k} > 0$, then $[\Lambda_1]_{i,k} > 0$ and $A_{i,k} = 0$.

Algorithm 3.1 Primal-Dual Active Set (PDAS) NMF Algorithm

1. Evaluate $\Lambda_1 = (AP - Y)P'$ and let

$$\mathcal{A}_1 = \{(i,k) : (\Lambda_1 - A)_{i,k} > 0\}, \quad \mathcal{I}_1 = \{(i,k) : (\Lambda_1 - A)_{i,k} \leq 0\}.$$

2. Let $A = 0$ on $\mathcal{A}_1$ and set $S = PP'$ and $G = YP'$. For each $1 \leq i \leq n$

$$A(i,k) = G(i,k)S(k,k)^{-1}, \quad (i,k) \in \mathcal{I}_1$$

3. Evaluate $\Lambda_2 = A'(AP - Y)$ and let

$$\mathcal{A}_2 = \{(k,j) : (\Lambda_2 - P)_{k,j} > 0\}, \quad \mathcal{I}_2 = \{(k,j) : (\Lambda_2 - P)_{k,j} \leq 0\}.$$

4. Let $P = 0$ on $\mathcal{A}_2$ and set $T = A'A$ and $G = A'Y$. For each $1 \leq j \leq m$

$$P(k,j) = T(k,k)^{-1}G(k,j), \quad (k,j) \in \mathcal{I}_2$$

Based on this fact, we develop the Primal-Dual Active Set (PDAS) method [30, 37, 38]. It uses the current $(\Lambda_1 - A)$ and determines the active index set $\mathcal{A}_1 = \{(i,k) : (\Lambda_1 - A)_{i,k} > 0\}$ where $A^+_{i,k} = 0$ for all $(i,k) \in \mathcal{A}_1$, and an inactive index set $\mathcal{I}_1 = \{(i,k) : (\Lambda_1 - A)_{i,k} \leq 0\}$ where $(\Lambda_1)^+_{i,k} = 0$ for all $(i,k) \in \mathcal{I}_1$. The PDAS algorithm then solves the system

$$(A^+P - Y)P' - \Lambda_1^+ = 0 \tag{3.1}$$

for (A^+, Λ_1^+) with $A^+ = 0$ on $\mathcal{A}_1$ and $\Lambda_1^+ = 0$ on $\mathcal{I}_1$.

Similarly, we have the active set $\mathcal{A}_2 = \{(k,j) : (\Lambda_2 - P)_{k,j} > 0\}$ where $P^+_{k,j} = 0$ for all $(k,j) \in \mathcal{A}_2$, and the inactive set $\mathcal{I}_2 = \{(k,j) : (\Lambda_2 - P)_{k,j} \leq 0\}$ where $(\Lambda_2)^+_{k,j} = 0$ for all

$(k, j) \in \mathcal{I}_2$. The PDAS algorithm then solves the system

$$A'(AP^+ - Y) - \Lambda_2^+ = 0$$

for (P^+, Λ_2^+) with $P^+ = 0$ on $\mathcal{A}_2$ and $\Lambda_2^+ = 0$ on $\mathcal{I}_2$.

In summary, we have Algorithm 3.1. Steps 1 and 2 are applied for the A-update, and steps 3 and 4 are for the P-update. For each i-th row of A^+ and j-th column of P^+, these steps require matrix-vector solutions of principle minors of T and S.

Remark 1. Notice that the linear system solve in step 2 is executed row-wise for n rows and column-wise for m columns in step 4. These steps can be costly depending on the size of the inactive sets. □

Remark 2. Recall for the Frobenius distance cost function, we have the property of coercivity that gives the existence of a minimizer pair $(\bar{A}, \bar{P})$. Lagrange Multiplier theory states that since $(\bar{A}, \bar{P})$ is a minimizer for $J(A, P)$, then there exists the dual variables Λ_1 and Λ_2 such that $(\bar{A}, \bar{P}, \Lambda_1, \Lambda_2)$ is a stationary point for $L(A, P, \Lambda_1, \Lambda_2)$. □

3.1.2 Regularized

Next, we consider the case of (1.4) when $\alpha > 0$ and $\beta > 0$. We extend Algorithm 3.1 to the Primal-Dual Active Set method for the regularized case (Algorithm 3.2). We need to redefine the necessary optimality for this case

$$\begin{cases} (AP - Y)P' + \alpha F - \Lambda_1 = 0, & \Lambda_1 \geq 0, \quad \langle \Lambda_1, A \rangle_F = 0 \\ \\ A'(AP - Y) + \beta(EP - P) - \Lambda_2 = 0, & \Lambda_2 \geq 0, \quad \langle \Lambda_2, P \rangle_F = 0 \end{cases}$$

The complementarity condition in [30, 38] then determines the primal-dual active set method as explained earlier.

Algorithm 3.2 PDAS-NMF Algorithm for Regularized (1.4)

1. Evaluate $\Lambda_1 = (AP - Y)P' + \alpha F$ and let

$$\mathcal{A}_1 = \{(i,k) : (\Lambda_1 - A)_{i,k} > 0\}, \quad \mathcal{I}_1 = \{(i,k) : (\Lambda_1 - A)_{i,k} \leq 0\}.$$

2. Let $A = 0$ on $\mathcal{A}_1$ and set $S = PP'$ and $G = YP' + \alpha F$. For each $1 \leq i \leq n$

$$A(i,k) = G(i,k)S(k,k)^{-1}, \quad (i,k) \in \mathcal{I}_1$$

3. Evaluate $\Lambda_2 = A'(AP - Y) + \beta (EP - P)$ and let

$$\mathcal{A}_2 = \{(k,j) : (\Lambda_2 - P)_{k,j} > 0\}, \quad \mathcal{I}_2 = \{(k,j) : (\Lambda_2 - P)_{k,j} \leq 0\}.$$

4. Let $P = 0$ on $\mathcal{A}_2$ and set $T = A'A + \beta(E - I_{p\times p})$ and $G = A'Y$. For each $1 \leq j \leq m$

$$P(k,j) = T(k,k)^{-1}G(k,j), \quad (k,j) \in \mathcal{I}_2$$

For the regularized case, however, the PDAS algorithm solves the system

$$(A^+P - Y)P' + \alpha F - \Lambda_1^+ = 0$$

for (A^+, Λ_1^+) with $A^+ = 0$ on $\mathcal{A}_1$ and $\Lambda_1^+ = 0$ on $\mathcal{I}_1$. In addition, it solves the system

$$A'(AP^+ - Y) + \beta(EP^+ - P^+) - \Lambda_2^+ = 0$$

for (P^+, Λ_2^+) with $P^+ = 0$ on $\mathcal{A}_2$ and $\Lambda_2^+ = 0$ on $\mathcal{I}_2$.

Remark 3. The PDAS method can be applied to any regularized problem, such as Problem (1.4) and Problem (1.5), as long as the necessary optimality system

$$\begin{cases} \frac{\partial J}{\partial A} - \Lambda_1 = 0, \quad \Lambda_1 \geq 0 \text{ and } \langle \Lambda_1, A \rangle_F = 0 \\ \\ \frac{\partial J}{\partial P} - \Lambda_2 = 0, \quad \Lambda_2 \geq 0 \text{ and } \langle \Lambda_2, P \rangle_F = 0 \end{cases}$$

can be efficiently solved. This algorithm is successful when $\frac{\partial J}{\partial A}$ and $\frac{\partial J}{\partial P}$ are linear. □

3.2 Numerical Validation

In this section, we test the PDAS-NMF algorithm. Recall α is the sparsity enhancement parameter; as α increases, the sum of the elements of A should decrease because of the increase of 0-elements. Also recall that β enhances the near-orthogonality of P; as β increases, the sum of the off-diagonal entries of PP' should decrease because of the increase of orthogonality. Consider Figure 3.1 [11]. We will use the matrix representation of this image, where each element of the matrix is a value in the range $0 : 255$, representing pixel intensity. We will compare standard NMF to PDAS-NMF for both unregularized and regularized cases.

Figure 3.1: Tree

We will examine several regularized cases. That is, we'll give results for different values of α and β. Furthermore, the Figure 3.1 is 400-by-600 pixels. Therefore, the choice in p can range from 1 to 400. We look at a few different p to see how this influences the results of the regularization. The same initializations for A and P are used within each p-case. In addition, we will run both the standard NMF and the PDAS-NMF for ten iterations.

In the Tables 3.1 and 3.2, "w.o." represents results found by standard NMF/ regularized standard NMF; "w." represents results found with the PDAS-NMF method. In Table 3.1, we examine the cases in which there is only sparsity regularization, only near-orthogonality

regularization, and both. In the cases $\alpha \neq 0$ and $\beta = 0$, we see the degree of sparsity of PDAS-NMF is significantly higher than regularized NMF. Similarly, in the cases $\alpha = 0$ and $\beta \neq 0$, PDAS-NMF increases the degree of partitions in the rows of P. In the case of both regularizations, PDAS-NMF finds a representation with a better balance. Therefore, for the individual matrices A and P, PDAS-NMF finds better results. We examine the fidelity and the balance of the representation in Table 3.2, a continuation of Table 3.1. This table displays the fidelity of the representations found in Table 3.1. Clearly, it can be seen that PDAS-NMF representations also have a better fidelity.

Remark 4. We have investigated several examples, and PDAS-NMF provides better results over standard NMF. □

Remark 5. When both α and β are non-zero, we see that we achieve the results we would like, but an increase in sparse-ness of A or degree of orthogonality of P means an increase in the relative norm. If one cares more about structural properties, then the fidelity has little meaning. In this example, we want regularization parameters based on a balance among the three terms [36]. □

Remark 6. Notice we'll measure sparsity of the standard NMF by the number of elements below 10^{-3} since 0-values are not actually attained. □

The final piece of this comparison is the cost function

$$J(A,P) = \frac{1}{2}\,||AP - Y||_F^2 + \alpha \sum_{i,k} A_{i,k} + \frac{\beta}{2} \sum_{k,\bar{k}} (PP' - I)_{k,\bar{k}}.$$

After examination, it can be seen that in every case, the PDAS-NMF method finds a representation that minimizes the cost function. This is significant given that PDAS-NMF results are more sparse and more nearly-orthogonal. Hence, it finds a better balance than standard NMF among the three criteria.

We give visual results in Figure 3.2 in order to verify that PDAS-NMF does indeed find the better result. We use the arguments from the final case in the Tables above. That is, we choose

Table 3.1: Near-orthogonality and Sparsity: without and with PDAS

p	α	β	w.o. $\frac{\sum_{k,\bar{k}}(PP'-I)_{k,\bar{k}}}{p^2}$	w. $\frac{\sum(PP'-I)_{k,\bar{k}}}{p^2}$	w.o. $\frac{A_{i,j}<1e-3}{np}$	w. $\frac{A_{i,j}==0}{np}$
100	0	0	162.0577	5.3618e3	1.7500e-04	0.4445
	6	0	162.6786	5.3615e3	1.7500e-04	0.4448
	0	6	0.7356	0.2322	2.5000e-05	0.4105
	3	6	0.7356	0.2363	2.5000e-05	0.4214
	6	3	0.7356	0.2625	2.5000e-05	0.4489
150	0	0	162.2677	2.4138e3	3.5000e-04	0.4497
	6	0	162.8027	2.4127e3	3.5000e-04	0.4504
	0	6	0.7341	0.1776	5.0000e-05	0.4034
	3	6	0.7341	0.1822	5.0000e-05	0.4314
	6	3	0.7341	0.2090	5.0000e-05	0.4707
200	0	0	160.6696	1.3637e3	4.5000e-04	0.4511
	6	0	161.1590	1.3584e3	4.5000e-04	0.4557
	0	6	0.7329	0.1356	3.7500e-05	0.3925
	3	6	0.7329	0.1412	3.7500e-05	0.4420
	6	3	0.7329	0.1720	3.7500e-05	0.5044

$(p, \alpha, \beta) = (200, 6, 3)$. After 100 iterations of Frobenius NMF, we do not have a representation as good as PDAS-NMF for 10 iterations, where we have fidelity 0.1423, "sparsity" 0.0031, and near-orthogonality 0.3310 after 100 iterations. This is verified by Figure 3.2. Note, the standard NMF may be used as a preprocessing step in order to better initialize A and P for the PDAS-NMF method.

3.3 Conclusion

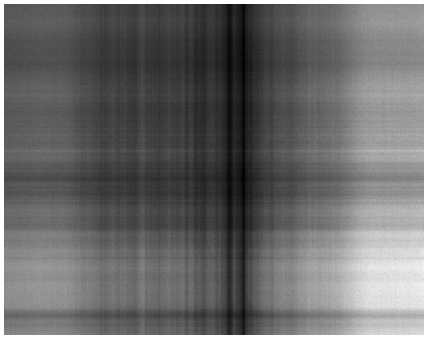
In summary, we have presented an algorithm that solves the Nonnegative Matrix Factorization problem (Problem 1.1) as a constraint problem. In addition, the PDAS-NMF removes the issue of standard NMF by giving elements in A and P the ability to attain 0-value and allows them to regain positivity if the algorithm decides. Furthermore, the PDAS-NMF method has been extended to the regularized NMF problem in which we desire the following structural properties: near-orthogonality of the basis P and sparsity of the assignment A.

In the example above, we have shown the success of PDAS-NMF over the standard NMF.

Table 3.2: Fidelity Term and Final Cost Function Value: without and with PDAS

p	α	β	w.o. $\frac{\|Y-AP\|_F}{\|Y\|_F}$	with $\frac{\|AP-Y\|_F}{\|Y\|_F}$	w.o. $J(A,P)$	w. $J(A,P)$
100	0	0	0.3158	0.1219	3.1463e8	0.4688e8
	6	0	0.3158	0.1219	3.1533e8	0.4703e8
	0	6	0.3159	0.1219	3.1485e8	0.4689e8
	3	6	0.3159	0.1218	3.2001e8	0.5555e8
	6	3	0.3159	0.1222	3.2514e8	0.6377e8
150	0	0	0.3138	0.1021	3.1065e8	0.3289e8
	6	0	0.3138	0.1021	3.1136e8	0.3314e8
	0	6	0.3140	0.1023	3.1110e8	0.3303e8
	3	6	0.3140	0.1021	3.1627e8	0.4293e8
	6	3	0.3140	0.1024	3.2141e8	0.5176e8
200	0	0	0.3138	0.0896	3.1065e8	0.2533e8
	6	0	0.3138	0.0893	3.1136e8	0.2553e8
	0	6	0.3140	0.0912	3.1114e8	0.2626e8
	3	6	0.3140	0.0883	3.1632e8	0.3588e8
	6	3	0.3140	0.0883	3.2145e8	0.4484e8

However, if n, m, p, large enough, the system solve can be costly. In chapter 5, we develop a primal-dual active set method where we use the conjugate gradient method as an incomplete solver for system 3.1. This allows us to use the PDAS method for large systems. We implement the algorithm for the application of deconvolution. This adaptation can be applied to standard AP representation.



(a) Frobenius NMF after 10 iterations

(b) Frobenius NMF after 50 iterations

(c) Frobenius NMF after 100 iterations

(d) PDAS-NMF after 10 iterations

Figure 3.2: For $(p, \alpha, \beta) = (200, 6, 3)$

Chapter 4

Rank Determination

The Nonnegative Matrix Factorization is a low-rank representation that highlights important patterns while reducing complexity. This begs the question how to best choose the middle rank p for the representation since it determines the amount of complexity in the representation. Typically, the middle rank is chosen based on prior knowledge, but intuition can be wrong. In this chapter, we develop a strategy to select the middle rank for the NMF using the concept of NMF-singular values and compare with the Akaike Information Criterion.

4.1 Introduction

Consider the set of data points

$$(x_1, y_1),\ (x_2, y_2),\ \ldots,\ (x_m, y_m)$$

in $\mathbb{R}^2$. Furthermore, consider the polynomial $\hat{p}_n(x)$ and the process of curve fitting. The goodness of fit, or fidelity, is the degree with which $\hat{p}_n(x)$ accurately fits the data points. By increasing the order n, one may also increase the accuracy. However, the polynomial may be quite complex, and can be highly oscillatory. The goal of this process, then, is to determine the complexity balance. That is, find the order n such that polynomial $\hat{p}_n(x)$ is both well-behaved and a good

fit to the data points.

For the case of NMF, complexity translates to the choice of p. The choice determines the trade-off between the fidelity and the amount of information present in the representation. In other words, a balance is necessary because if p is chosen to be small, then we reduce the complexity of the data present in our representation but this increases the fidelity. If p is chosen to be larger, we achieve a smaller fidelity but we may not have reduced enough complexity present in the original data. The method we develop will determine the balancing point of these two trade-off quantities.

4.2 NMF-Singular Values

In order to introduce the NMF-singular values, we first recall the Singular Value Decomposition (SVD) of a rank r matrix $Y_{n\times m}$ in the decomposition

$$Y = U\Sigma V',$$

where $U \in \mathbb{R}^{n\times k}$ and $V \in \mathbb{R}^{m\times \bar{k}}$ are orthonormal and $\Sigma \in \mathbb{R}^{k\times \bar{k}}$ is the matrix with diagonal entries

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0,$$

known as singular values, and 0-entries elsewhere. We will assume $k = \bar{k}$. We have also assumed U and V are real; note that they can be unitary. This is not the case in this chapter since we deal with real data.

The SVD is used in many applications as a reduced-rank representation of Y. Typically, this factorization is reduced in the following manner

$$Y_m = U_m\Sigma_m V_m^{'},$$

where Σ_m is the diagonal matrix whose diagonal elements are the m dominant singular values

of Y. The truncation at σ_m is, by some method of determination, the threshold between the most significant information and the least significant information. One such method assumes $\sigma_{m+1}, \ldots, \sigma_r$ do not hold necessary information if they are significantly smaller in magnitude than σ_m. Another method assumes these singular values are not significant if there is minimal change in value across $\sigma_m, \ldots, \sigma_r$.

Remark 1. The rank m should be related to p since the rank of Y_m is m, and the rank is exactly the complexity of our interests. □

We would like to develop a similar system for data reduction. Thus, we relate the NMF to the SVD. In the SVD, the rows of V' are the orthonormal basis of Y, ordered by the corresponding singular values; as we stated earlier, the rows of P are the basis of Y in the NMF. Recall that A is the assignment of P; in this same manner, $U\Sigma$ is the assignment of those basis vectors in V'. Observe that each column of $U\Sigma$ contains the information from the singular values. Therefore, the columns of $U\Sigma$ determine the contribution scale of the rows of V'.

Remark 2. If we find a reduced rank representation of Y by either the SVD or the NMF, we don't have an exact factorization of Y. The SVD and NMF are different by nature. A clear advantage of the NMF over the SVD is that the factors are nonnegative. □

We can extend this to the NMF by considering the NMF-singular value defined as

Definition 1. *Given $Y \in \mathbb{R}^{n\times m}$ and a Nonnegative Matrix Factorization AP, where $A \in \mathbb{R}^{n\times \bar{p}}$ and $P \in \mathbb{R}^{\bar{p}\times m}$ with $\bar{p} = \min(n, m)$, the NMF-singular values are defined*

$$s_k = ||A(:, k)||_1, \quad k = 1, \ldots, \bar{p}$$

Thus, the s_k-values describe the magnitude of contribution for the rows of P. In this manner, we may consider the s_k values as generalized singular values, or NMF-singular values. Furthermore, since each s_k determines the contributions of each basis vector $P_{k,*}$, we can use their information to indicate the choice in p just as singular values are used to truncate the SVD.

In order to find the s_k-values, we do need to find the NMF representation for $\bar{p} = \min(n, m)$. We make note here that we can do this because the following examples are not significantly large. However, if we were dealing with a much larger dataset, choosing $\bar{p} = \min(n, m)$ is not feasible. In this case, we would want to choose a $\bar{p}$ that is largest possible such that the NMF can be calculated in a reasonable amount of time.

In addition to this choice in $\bar{p}$, we do not use regularized NMF for this chapter. We found that $(\alpha, \beta) \neq (0, 0)$ did not change the pattern of the NMF-singular values, which is what we need for our analysis.

4.3 Akaike Information Criterion

As we stated earlier, we want to determine the p such that we have a balance between fidelity and the complexity in representation. The statistical model selection criterion Akaike Information Criterion (AIC) [10, 12, 42, 75] expresses the relationship between these two quantities.

Recall the Kullback-Leibler (K-L) divergence is defined

$$D(Y||AP) = \sum_{i,j} \left(Y_{i,j} \log \left(\frac{Y_{i,j}}{(AP)_{i,j}} \right) \right), \tag{4.1}$$

where Y and AP are probability densities. This measures the gap between the density Y and the density AP. That is, the K-L divergence represents the information lost when the model AP is used to represent the original data Y. The best model AP will be the one that minimizes $D(Y||AP)$. This divergence, however, requires knowing the true distribution of Y; we do not have this information. Therefore, we need to estimate.

Akaike states the Kullback-Leibler divergence is asymptotically estimated by the formulation known as the Akaike Information Criterion (AIC). This is defined

$$AIC = -2 \log L(AP; Y) + 2K, \tag{4.2}$$

where L is the likelihood function, the probability of Y given AP, and K is the total number of elements in A and P. In the case of NMF, Gaussian distribution is assumed. Therefore, $-\log L(AP;Y)$ is $||AP-Y||_F^2$. Also, since the parameters we are estimating are every entry in A and in P, then we use $K = n \times p + m \times p = (n+m) \times p$. The preferred representation is the one with the minimum AIC value.

It is significant to observe (4.2) does indeed express the trade-off relationship between fidelity and complexity. If p remains small, K remains small but $-\log L(AP;Y)$ will be large since there will be less data representing Y. Thus, AIC will be overpowered by the fidelity. However, as p increases, $-\log L(AP;Y)$ will decrease but K will increase. Thus, AIC will be overpowered by the complexity. Therefore, the minimum value of AIC occurs when both the fidelity and the complexity are evenly matched.

4.4 Comparison Study

We now develop our method using the NMF-singular values by three examples: a time-series dataset provided by Dr. Elana Fertig [59], an image of the well-known Lena, and a term-by-document dataset. We will compare the results to the choice by AIC.

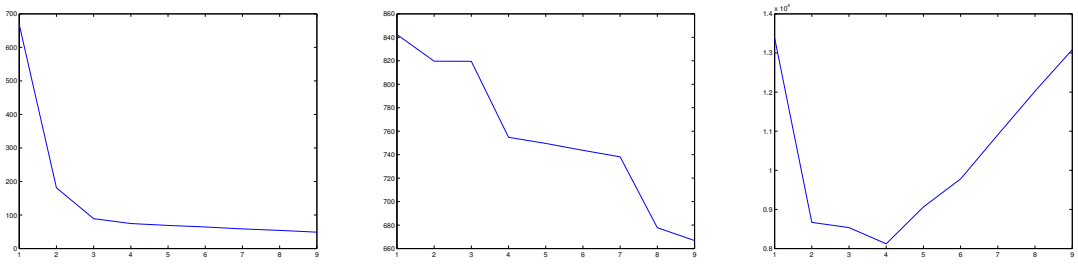
The first example is time-series data with 1363 different cells represented and 9 time points. The data are samples of gastrointestinal stromel tumor cells at 9 time points after being treated with 10 μmol/L of imatinib mesylate (IM), which has had success in treating chronic myelogeneous leukemia. We refer you to Figure 4.1.

In Figure 4.1a, we display the SVD singular values. If we truncate by the SVD singular values, we will choose $p = 3$. As we stated earlier, we aim to find a similar systematic method using the NMF.

For our method, we calculate the s_k-values from Definition 1, for $k = 1, \ldots, 9$. In Figure 4.1b, we plot the s_k-values. Similar to the plot of the SVD singular values, along the y-axis is the value of the s_k-values and the x-axis gives k. We want to examine the curvature. At the point of maximum curvature, there occurs a significant turning point between the fidelity and

the complexity [29]. This is the turning point we use in order to determine the best-fit rank. The greatest change in curvature is contained in the range $p = 3 : 7$. We claim the optimal p is in this range. More specifically, we claim the optimal is $p = 4$.

We'll examine the AIC for the time-series data set for $p = 1 : 9$. The time-series data set gives us the fidelity term $||(AP - Y)./sqrt(\Sigma)||_F^2$, where Σ is the co-variance matrix for the data element-wise. In Figure 4.1c, we've plotted the AIC for each p. We see that the minimal AIC occurs when $p = 4$. Hence, this choice in rank provides the best balance between information lost and reduced complexity. Moreover, AIC has confirmed our analysis of the s_k-curve and the assertion $p = 4$.



(a) σ_k-values (b) s_k-values (c) AIC values

Figure 4.1: Time-Series Dataset

Consider the second example, a gray-scale image 'Lena' in Figure 5.1a. The matrix representation is 512 pixels by 512 pixels [33]. Each element represents pixel intensity. In Figure 4.2a, we have the curve for the log-SVD singular values in order to provide visuals. Previously, it traced the graph's border. The curve indicates a significant change in values very early. We find the s_k-values for $k = 1, \ldots, 512$. Figure 4.2b displays the s_k-values. We examine the curvature of the s_k-curve and determine that the best range is $p = 20 : 80$.

Now, we compute the AIC (4.2) for each p in order to verify our assertion. For the image, we must take into consideration the pixel intensity and scale accordingly so fidelity will not

outweigh complexity too significantly. We take the first term $||AP - Y||_F^2$ and we divide by (4*256). Figure 4.2c shows the AIC values we found. The minimum AIC value gives $p = 32$, confirming the range selection by s_k-curve analysis as containing the optimal selection for p.

Figure 4.3c displays the representation for our Y. We can visually verify that the optimal p chosen by our method does indeed maintain important features in the face of the woman, while features that may not matter, such as details in the hair strands, are fuzzy. In Figure 4.3, we display two images, $p = 10$ and $p = 90$. We've chosen these p's based on their position of the s_k-curve. The rank $p = 10$ is positioned where the large quantity of fidelity overpowers complexity, and rank $p = 90$ is positioned where the complexity is larger in value than fidelity. Table 4.1 confirms this assertion.

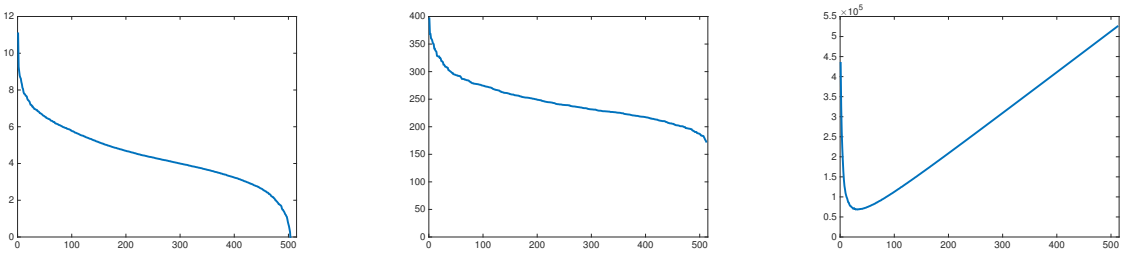
Table 4.1: Fidelity-Complexity Balance for Lena

p	scaled fidelity	K
10	99315	10240
32	35784	32768
90	11804	92160

Figure 4.3d shows that $p = 10$ gives an image which is much fuzzier than the optimal image, and thus we lose features. This clearly implies that more information is lost. Figure 4.3b gives an image for $p = 90$, which is much clearer than the optimal woman (there is more information). However, this does not provide the best balance (See Table 4.1). Clearly, at $p = 32$ we have the better balance, which is indicated by the range chosen via NMF-singular values.

Remark 3. We use the s_k-curve analysis to highlight a range of possible p. This allows flexibility in the choice of p since the balance displayed by $p = 32$ in Table 4.1 may not significantly change within the highlighted range. In addition, choosing a range of feasible p reduces the amount of computations necessary. □

Finally, we include an example of a term-by-document data set, that is, the medlars dataset



(a) $\log(\sigma_k)$-values (b) s_k-values (c) AIC values

Figure 4.2: Lena Analysis

of medical abstracts[1]. The dataset contains 1,033 documents and 5,831 terms. The dataset can be found in a file with format Harwell-Boeing for sparse data. Figure 4.4a is the curve of the SVD singular values, and Figure 4.4b is the curve of the s_k-values for $k = 1, \ldots, 1033$.

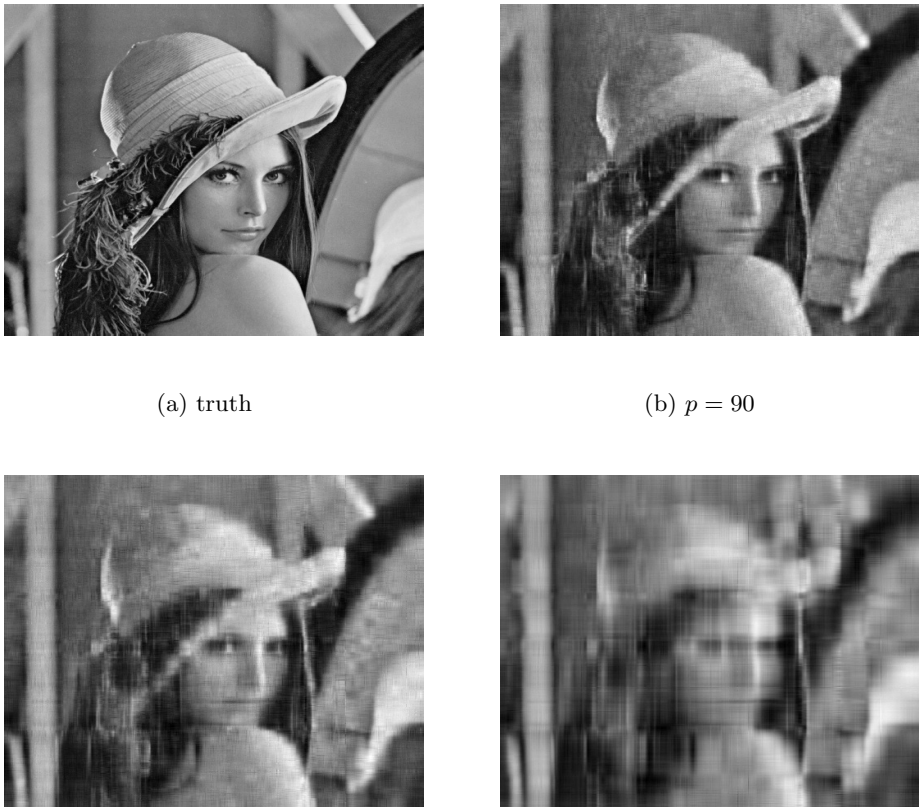
Figure 4.4b displays the s_k-values for the medlar dataset, found through a factorization $Y \approx AP$, where $\bar{p} = 1033$. Again, examining the s_k-curve, we determine that p will approximately be in the range of 10 to 20. We calculate the AIC values to confirm this judgement.

Scaling the fidelity is a necessity in the first two cases. For this database, we also need to scale. We need to consider sparsity when scaling. In the data Y, the sparsity is 99.14%. Then, the remaining percentage 0.86% implies the nonzero elements of Y. Using this value, we scale the first term of (4.2). That is,

$$AIC = 2\left(\frac{||AP - Y||_F^2}{0.0086} + K\right).$$

In Figure 4.4c, we calculate the AIC values for $p = 1 : 5 : 1033$. We see that the minimum AIC is indeed achieved in the first 20 values. We take a closer look the AIC for values $p = 1 : 20$, see Figure 4.4d. The minimum AIC occurs at $p = 15$. This is the best-fit rank according to our method.

[1] http://web.eecs.utk.edu/research/lsi/



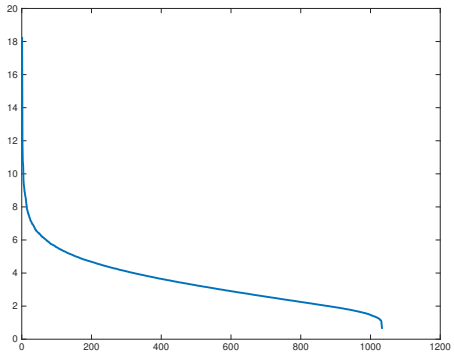
(a) truth (b) $p = 90$

(c) $p = 32$ (d) $p = 10$

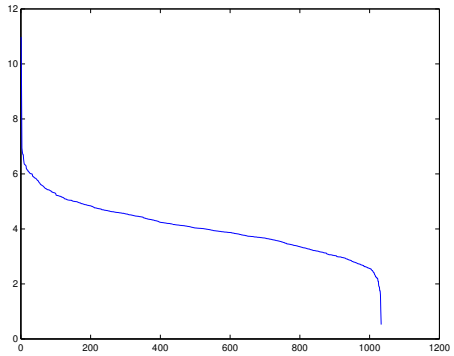
Figure 4.3: Woman

4.5 Conclusion

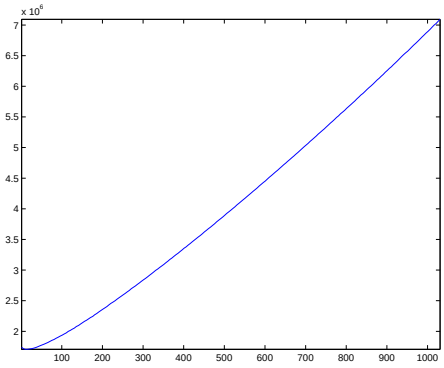
In summary, we have defined the NMF-singular values based on the "connection" between the SVD and the NMF. In addition, we have developed a systematic method to find a proper middle rank p. We find the NMF-singular values, then we identify a neighborhood of p which contains the point of maximum curvature on the NMF-singular value curve. This point defines the balance between fidelity and complexity. Furthermore, we compared the estimated range by s_k-curve to an established information criterion AIC. We show the p selected by AIC is contained in the selected range by the NMF-singular value curve analysis. Thus, confirming the



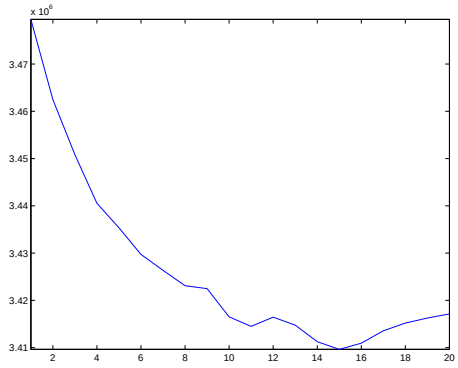
(a) σ_k-values



(b) s_k-values



(c) AIC values $p = 1 : 5 : 1033$



(d) AIC values $p = 1 : 20$

Figure 4.4: Medlar Dataset

NMF-singular values indicate a proper p. In general, the choice of the complexity is application dependent, and we've used our method for three examples in different applications to show we successfully generalized this choice.

Future work will be to examine this method when the generalized Kullback-Leibler divergence is used as a fidelity term in place of the Frobenius norm.

Chapter 5

Blind Deconvolution, Atoms, and NMF

In this chapter, we describe the concept of atoms and how they classify the convolution kernel. More specifically, we use the point-spread function kernel. Furthermore, we develop for the blind deconvolution a divergence NMF that incorporates atomic information. In addition, as alternatives, we develop a coordinate-descent NMF method and a primal-dual active set method for deconvolution.

5.1 Introduction

Consider the signal processing of images formed from signals received via satellite, i.e. satellite imagery. We have the convolution

$$f(x) = \int_{\Omega} k(x, y)p(y), \tag{5.1}$$

where $f(x)$ is the measured signal, $k(x, y)$ is the kernel function for the convolution, and $p(y)$ is the pattern of the original signal and density of Ω. In this case of satellite imagery, the kernel function will be the point-spread function (PSF).

In signal processing, *deconvolution* is the process of separating the true signal p when we know both $f(x)$ and $k(x, y)$. In the case when we only know $f(x)$, we aim to estimate both k and p from f. This is called *blind deconvolution*. However, we commonly have prior knowledge of k. For example, if a measured signal is received from a telescope, we can assume the point-spread function has properties of the Airy disc, a well-known PSF associated with lens objects such as cameras. The prior knowledge of k gives us the associated atoms, the weights that describe the convolution kernel.

To understand the concept of atoms and their role in the convolution, consider the discretized convolution

$$f = Ap, \tag{5.2}$$

where $f \in \mathbb{R}^{N\times 1}$ is the measured signal vector, $A \in \mathbb{R}^{N\times N}$ is the convolutional operator in the atomic class $\mathbb{Q}$ of kernels, and $p \in \mathbb{R}^{N\times 1}$ is the vector representation of the clean signal.

In the case where the PSF has local Gaussian distribution with 3 distinct elements, $a_1,\ a_2,\ a_3$, and a_1 the center:

$$\begin{bmatrix} a_3 & a_2 & a_3 \\ a_2 & a_1 & a_2 \\ a_3 & a_2 & a_3 \end{bmatrix},$$

we then have

$$f_\ell = a_2(p_{\ell+1} + p_{\ell-1} + p_{\ell-N} + p_{\ell+N}) + a_1 p_\ell + a_3(p_{\ell-N-1} + p_{\ell-N+1} + p_{\ell+N-1} + p_{\ell+N+1}), \tag{5.3}$$

where $a_1,\ a_2,\ a_3$ are the weights for 9-point neighborhood of ℓ. The atoms in $\mathbb{Q}$ represent the weights in $a = (a_1, a_2, a_3)$, as in (5.3). In this way, one may write $A = A(a)$ for the atom $a \in \mathbb{Q}$.

Remark 1. In terms of statistics, the atoms refer to the prior distribution of $a \in \mathbb{Q}$ for $A(a)$. As well, the order of a may be constrained, e.g. the diagonally dominant PSF, diagonal-to-tail decaying PSF, or the local Gaussian symmetry in (5.3). □

In this chapter, we compare divergence NMF to the deconvolution method of Richardson-

Lucy. We will show the built-in scaling by divergence NMF makes it the better performance algorithm. Secondly, we develop a blind deconvolution NMF based on the concept of the atoms. Finally, we will also develop a coordinate-descent method and a primal-dual active set method for the deconvolution problem.

5.2 NMF and Deconvolution

In this section, we will discuss the connection of divergence NMF to the Richardson-Lucy method for the deconvolution problem. Furthermore, we introduce relaxation and regularization to the divergence NMF.

Reconsider the integral (5.1). The Richardson-Lucy (R-L) algorithm is an iterative technique for the deconvolution problem [6]. Richardson proposed a method for the restoration of images in the presence of Poisson distributed noise [63]. Note, because of this assumption of Poisson distribution the algorithm does not normalize. The resulting method is the approximating update

$$p_i \leftarrow p_i \sum_{\ell} \frac{k_{i,\ell} f_\ell}{\sum_j k_{j,\ell} p_j}, \tag{5.4}$$

where p_i is the current iterate of the i-th element of p and the initialization of the i-th element is estimated.

Remark 2. In applications, the convolution kernel is symmetric, i.e. $k(x, y) = k(y, x)$. In addition, it is assumed by the method (5.4) the kernel convolution k is density. That is, $\int_\Omega k(x, y)dx = 1$. In addition, the atoms of the kernel are often nonnegative. □

Lucy discusses this same method [51]. He shows the convergence of this method as $r \to \infty$ provided

$$\int_\Omega p(y)dy = 1, \quad p \geq 0.$$

The method converges to a solution which maximizes the Poisson probability density function

$$L(k \otimes p; f) = \prod_x \frac{(k \otimes p)^{f(x)} e^{-(k \otimes p)(x)}}{f(x)!},$$

where $k \otimes p$ is the tensor-notation for the convolution [43, 51]. Note, this is equivalent to minimizing the negative log-likelihood function.

We now relate this method to the divergence NMF, originally presented in Algorithm 2.2. Recall the discretized convolution equation (5.2)

$$f = Ap,$$

where $f \geq 0$, $A = A(a) \geq 0$, and $p \geq 0$. Therefore, we can represent the signal f as a nonnegative matrix factorization. We can find this representation by optimizing the divergence criterion for the NMF. We formulate the NMF deconvolution problem

$$\min_{p \geq 0} \; J(p) = \min_{p \geq 0} \; \sum_i \left(f_i \log \frac{f_i}{(A(a)p)_i} - f_i + (A(a)p)_i \right).$$

We give the deconvolution NMF in Algorithm 5.1, where A represents the convolution operator.

Algorithm 5.1 deconvolution NMF - unregularized

$$p \leftarrow p. * (A' (f./(Ap)))./(\sum_i A_{i,k}) \tag{5.5}$$

Remark 3. The construction of A and the computation of Ap as described in (5.2) will increase in cost as the size of p increases. Therefore, we replace this construction and multiplication with the Matlab function 'imfilter($\cdot$)', where this function implements the convolution operation. □

The R-L algorithm assumes symmetry for the convolution kernel. We will assume the same for divergence NMF. However, we do not assume normalization, thus making the built-in scaling

Table 5.1: Deconvolution Comparison

Algorithm	$D(f\|\|Ap)$	$D(p_0\|p)$
RL	6.6927e5	3.2273e6
NMF	6.4655e5	2.5636e6
rlxd. NMF	9.3741e5	1.3342e6
reg. NMF	3.9613e6	5.4541e5

a significant characteristic of divergence NMF. This difference in algorithm gives the divergence NMF an advantage over Richardson-Lucy algorithm. We will show that this scaling not only produces a better representation, but it also produces a better result for the ground truth p.

Matlab has a built-in Richardson-Lucy algorithm called 'deconvlucy(·)'. We compare results of this method and our deconvolution NMF. Our test image is Figure 5.1a, the well-known test image of Lena [33]. We convolve the image with a 9×9 point-spread function of local Gaussian distribution standard deviation 3 (15 atoms), created by Matlab's 'fspecial(·)' and 'imfilter(·)'. The 'fspecial(·)' function creates point-spread functions. Then, we add 42% relative noise to get Figure 5.1b.

What matters in these results is the ground truth estimate since that is the goal of the deconvolution problem. We see from Table 5.1 that the deconvolution NMF finds an estimated p with a smaller divergence gap to the truth than the Richardson-Lucy estimate. Thus, we have a smaller divergence gap in regards to the representation Ap as well. In Figure 5.1c and 5.1d, we show the image found for each of the algorithms. Visually, we can verify the deconvolution NMF obtains an image with better pixel intensity.

Remark 4. It is important to state here that we used 10 iterations of deconv-NMF for the results in Table 5.1. This is the same number of iterations we use for deconvlucy(·). □

Furthermore, we've relaxed the deconvolution NMF, i.e.

$$p^+ = (1-t)p + tq, \tag{5.6}$$

where q = (5.5). We use this update in addition to the deconvolution NMF. We found for

(a) Truth

(b) Convoluted and Noise

(c) Richardson-Lucy via Matlab

(d) Deconvolution NMF

(e) Relaxed Deconvolution NMF $t = 0.2$

(f) Deconvolution NMF $(\alpha, \gamma) = (0.25, 10^{-6})$

Figure 5.1: Lena and Deconvolution

the particular case of Lena $t = 0.2$ produces better results. The relaxed update finds a p that minimizes the divergence gap more so than the NMF update without relaxation. Observe that as we minimize the gap between the truth and our results, the gap between the measured data and our results widens.

In chapters 2 and 3, we described the importance of regularization in attaining a desired structure for the representation AP (as well as the benefits of stabilization and convergence). This structure is highly dependent on what is needed for a particular application. When discussing the deconvolution of measured signals, we often deal with noise in signals. Therefore, we introduce regularization of the ground truth in order to denoise.

Consider the problem

$$\min_{p\geq 0} J(p) = \min_{p\geq 0} \sum_i \left(f_i \log \frac{f_i}{(Ap)_i} - f_i + (Ap)_i \right) + \alpha \sum_i p_i + \frac{\gamma}{2} \langle p, Hp \rangle_{\mathbb{R}^N}$$

where $A = A(a)$, $\alpha, \gamma \geq 0$ are regularization parameters on sparsity and smoothness on p, respectively. We measure sparsity on p by the ℓ_1-norm; as α increases, the sparsity of p is expected to increase, and thus $\sum_i p_i$ is expected to decrease. We measure smoothness on p by the H^1-regularization. These regularization terms are also discussed in chapter 1. We choose smoothness to reduce noise and sparsity for background recovery.

We then have the regularized deconvolution NMF with the steps

$$p = p. * (A' (f \,./\, (Ap)) \,./\, (\textstyle\sum_i A_{i,k} + \alpha e)),$$
$$p = (\, speye(N) + \gamma * h \,) \backslash p.$$

where 'speye(N)' creates a sparse identity matrix size $N \times N$, e is a vector of ones, and h is the finite difference operator.

In Table 5.1 and Figure 5.1f, we use the regularization parameters $(\alpha, \gamma) = (0.25, 10^{-6})$ based on the amount of noise reduced and the minimization of the divergence gap. We can see in Table 5.1 the regularized result does in fact find a representing p with a smaller divergence

Table 5.2: Lena: Deconvblind v. NMF Blind

Algorithm	$D(psf0\|\|psf)$	$D(p_0\|\|p)$	$D(f\|\|Ap)$
deconvblind	0.0556	3.5695e6	6.0347e5
NMF	0.0372	2.5895e6	6.4835e5
reg. NMF	0.0165	4.8962e5	4.0077e6

gap.

5.3 NMF and Blind Deconvolution

In this section, we examine the case of blind deconvolution in which we have prior information on the kernel convolution, i.e. atomic information.

Reconsider the equation (5.2). In this example, we have the prior information: three atoms with local Gaussian distribution. In order to solve for these elements of $A(a)$ without creating A at each step, we create a matrix representation for the atomic system instead, i.e. $aP = f$:

$$\begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix} \begin{bmatrix} p_1 & p_2 & p_3 & \dots & p_N \\ \sum_1(p_1) & \sum_1(p_2) & \dots & \dots & \sum_1(p_N) \\ \sum_2(p_1) & \sum_2(p_2) & \dots & \dots & \sum_2(p_N) \end{bmatrix} = \begin{bmatrix} f_1 & f_2 & f_3 & \dots & f_N \end{bmatrix}, \tag{5.7}$$

where, in this case,

$$\begin{aligned} \textstyle\sum_1(p_\ell) &= p_{\ell+1} + p_{\ell-1} + p_{\ell+N} + p_{\ell-N} \\ \textstyle\sum_2(p_\ell) &= p_{\ell+N+1} + p_{\ell-N+1} + p_{\ell-N-1} + p_{\ell+N-1}. \end{aligned}$$

This matrix system can be extended to point-spread functions with more than 3 elements and their corresponding sums.

Traditional blind deconvolution algorithms have an expectation-maximization approach [44]. For example, Matlab has a blind deconvolution algorithm called 'deconvblind(·)'. We compare our results to Matlab's using the same example as in a previous section: a blurred Lena (using a 9×9 local Gaussian blur with 3 standard deviation – 15 atoms) with 42% noise. We use the same

(a) Deconvblind (b) Blind NMF (c) Blind NMF $(\alpha, \gamma) = (0.5, 10^{-6})$

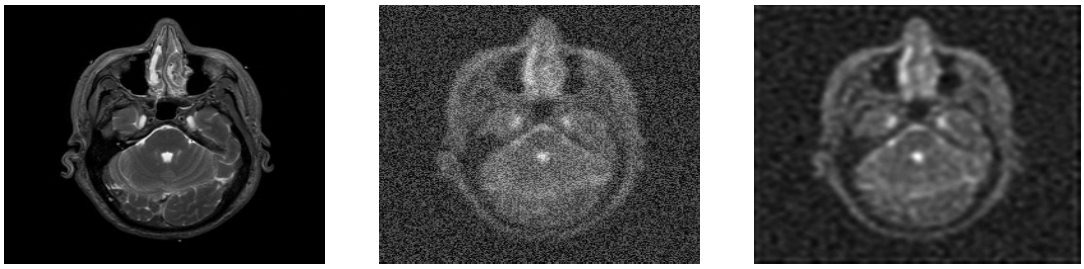
Figure 5.2: Lena and Blind Deconvolution

initializations for the ground truth estimate and PSF for both our blind NMF deconvolution algorithm and for Matlab's blind deconvolution method. Note, we initialize p to be ones and the PSF to be a random 9-by-9 matrix with local Gaussian distribution, with standard deviation unassumed.

The purpose of the blind deconvolution problem is to find a good fit p and point spread function, so we check the following in Table 5.2: the closeness of the ground truth image found to the actual ground truth and the closeness of the PSF found to the original PSF. As well, we compare these using the measurement of the divergence gap.

The Table 5.2 shows that the blind deconvolution NMF not only finds an estimate p with a smaller divergence gap to the truth (when compared to deconvblind), but it also finds an estimated PSF with a smaller divergence gap. In addition, regularization by $(\alpha, \gamma) = (0.5, 10^{-6})$ reduces even further the divergence gap between the estimated p and the truth. Observe, though, that the divergence gap between the measured data f and the representation increases. This is to be expected since we are moving further away from the noisy data and closer to the truth. In Figures 5.2a-5.2c, we can visually verify that blind deconvolution NMF finds a better result for p, and the regularized image has even less noise.

To further confirm the blind deconvolution NMF is a successful method, we examine a second example. Consider Figure 5.3a. This is a T2 image of a human male head found in the



(a) Truth (b) Convoluted and Noise (c) Blind NMF $(\alpha, \gamma) = (5, 10^{-5})$

Figure 5.3: Brain and Blind Deconvolution NMF

database The Visible Human Project [60] obtained by an Magnetic Resonance Imaging (MRI) scan [27,45,54]. This is one of many types of medical imaging applications; descriptions of other types, such as PET and ODT, can be found in [3, 7, 69]. The process of scanning, collecting data, and reconstructing data into images results in much noise and corruption.

For this experiment, we corrupt this image by a 9-by-9 point-spread function with local Gaussian distribution and standard deviation 2 (15 atoms) and the addition of 53% noise. Note, the image in Figure 5.3a is 256 pixels-by-256 pixels.

Our goal is to remove the convolution and smooth the noise so all that remains is the true image of the brain. We initialize p to be a vector of ones, and we initialize the PSF to be a random 9-by-9 matrix with local Gaussian distribution.

Table 5.3 shows that once again blind deconvolution minimizes the gap with the truth. In addition, regularization blind deconvolution NMF further minimizes this gap with $(\alpha, \gamma) = (5, 10^{-5})$. This is validated by Figure 5.3c. We can see that noise is significantly dulled.

Table 5.3: Brain Scan: Deconvblind v. NMF Blind

Algorithm	$D(psf0\|\|psf)$	$D(p_0\|\|p)$	$D(f\|\|Ap)$
deconvblind	0.7151	3.5103e6	1.4688e7
blind NMF	0.0305	3.0324e6	5.8582e5
reg. blind NMF	0.0320	1.0825e6	6.0925e6

Table 5.4: Numerics for Complete Blind Comparison 1

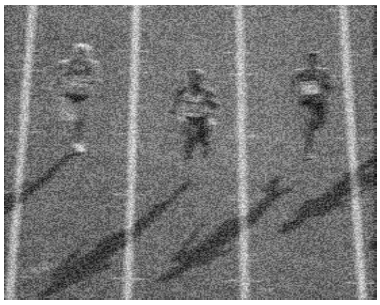
Algorithm	$D(f\|\|Ap)$	$D(p0\|\|Ap)$	$D(p0\|\|p)$
div. NMF	9.4873e4	8.9039e5	n/a
blind NMF	1.9940e5	8.4175e5	8.7641e5

5.3.1 Blind Comparison

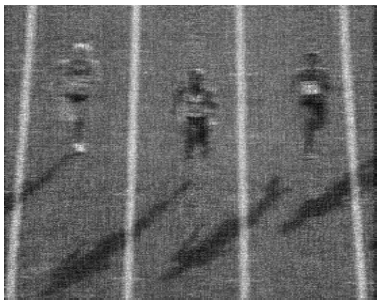
In this subsection, we investigate the outcome of the blind NMF method in a completely blind situation. That is, we do not have prior information about the atoms of the kernel function, e.g. the number of atoms or their distribution. Recall in chapter 2, we recover a representation using the divergence NMF. We consider the divergence NMF as a completely blind method since it finds a representation based on the data without prior information on the factors. We will compare the divergence NMF to blind NMF.

For the purpose of this section, we will convolute the runners image by a motion blur with the parameters $(len, \theta) = (6, 6)$, the length and the angle of the line that the image is smeared along, and add 21% noise. See Figure 5.4a. For a fair comparison between the divergence NMF and the blind NMF, we want to be as blind as possible. Hence, we will assume atoms far from the truth: they have a local Gaussian distribution and compile into a symmetric PSF. In addition, we will assume there are 15 atoms.

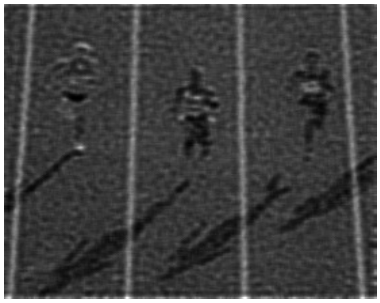
Consider Table 5.4 where we have results for Figures 5.4b-5.4d. We let middle rank be 250 for the divergence NMF (and iterate 100). The nature of the convolution allows us to examine not only representation but also the ground truth estimate p. Therefore, we give the divergence gap for both the representation and p for the blind NMF. By the Table, we see that divergence NMF finds a better representation compared to the measured data. The blind NMF, however, finds a better ground truth estimate in terms of the representation found. Given the blind situation, the representation by blind NMF finds a better estimate than the resulting p. Visually, we can verify in Figure 5.4d that the blind NMF representation gives a clearer image than the estimating p and an even clearer image than the divergence NMF.



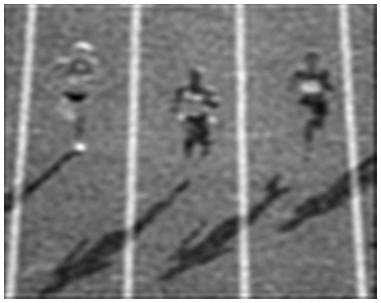
(a) Motion and Noise



(b) Standard NMF

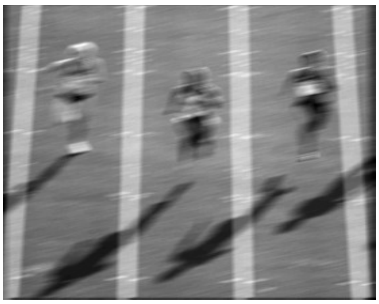


(c) blind NMF p, 15

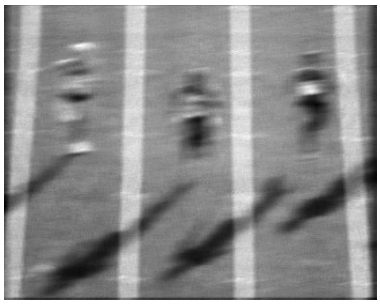


(d) blind NMF Ap, 15

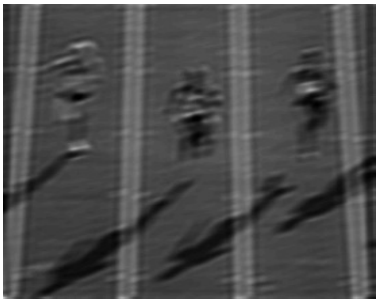
Figure 5.4: Complete Blind Comparison 1



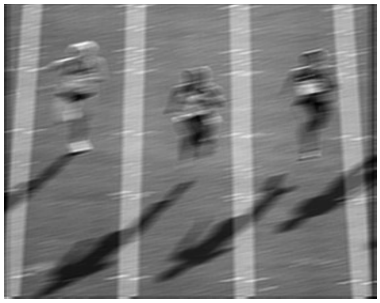
(a) More Motion



(b) Standard NMF



(c) blind NMF p, 15



(d) blind NMF p, 3

Figure 5.5: Complete Blind Comparison 2

Now, consider Figures 5.5a-5.5d, where Figure 5.5a is the runners image convoluted with a motion filter by parameters $(len, \theta) = (15, 15)$. We do not include noise because we consider the image as corrupted enough. We provide results by divergence NMF, blind NMF with the same assumptions as previously, and a second case of blind NMF where we assume there are 3 atoms that form a 3×3 point-spread function with local Gaussian distribution. We have numerics in Table 5.5.

For this case, the divergence NMF loses its advantage even for the gap between representation and measured data. For the blind NMF with the assumed 15 atoms, we have similar results as previously: the representation provides a better estimate for the ground truth than the actual ground truth estimate. We provide the estimated p in Figure 5.5c. In the third case,

Table 5.5: Numerics for Complete Blind Comparison 2

Algorithm	$D(f\|\|Ap)$	$D(p0\|\|Ap)$	$D(p0\|\|p)$
div. NMF	6.5339e3	2.8818e5	n/a
blind NMF, 15	3.7108e3	2.8560e5	3.0576e5
blind NMF, 3	355.8214	2.8055e5	2.7449e5

we see in the Table that assuming 3 atoms is advantageous in all areas. We give the resulting p in Figure 5.5d.

In summary, blind NMF does best when there is prior information, as we showed earlier. However, we show in this section that blind NMF is comparable to divergence NMF as a completely blind method. In addition, the results depend on the assumptions made on the atoms of the data. We see in the second example that this change in assumption improved results.

5.4 Coordinate-Descent

In this section, we develop an alternative algorithm for the deconvolution problem, a coordinate-descent method. That is, we will assume A is known.

We want to optimize

$$\min_{p\geq 0} \ \frac{1}{2}\,|Ap - f|_2^2, \tag{5.8}$$

where $f \in \mathbb{R}^N$, $A \in \mathbb{R}^{N\times N}$, and $p \in \mathbb{R}^N$, $N = nm$ where the measured image is $n \times m$. Similar to earlier, we will assume $f \geq 0$ and the atoms a of A are nonnegative.

Coordinate-descent methods update one variable p_i at a time by solving a sequence of minimization subproblems [16,28,35,48,72,77]. These subproblems are scalar updates. This type of method can be more beneficial since scalar operations are simpler to implement. However, this method is only efficient when the subproblems can be solved relatively quick [48]. Commonly, the method will visit each coordinate of p in order. That is, it will update p_1, then update p_2, and we do this for all p_i until we finally update p_N.

In this section, we derive a coordinate-descent method for the NMF problem with a cyclic sweep. Consider the i-th coordinate subproblem is

$$\begin{aligned} \min_\eta F(\eta) = &\quad \min_\eta \ \tfrac{1}{2}\,|A(p+\eta e_i) - f|_2^2, \\ \text{subject to} &\quad p + \eta e_i \geq 0 \end{aligned}$$

where e_i is the unit vector with 1 in the i-th position and 0 elsewhere, and η is the update step-size on the i-th coordinate of p. We want to find the direction η which minimizes the cost function. Notice,

$$\begin{aligned} |A(p+\eta e_i) - f|_2^2 &= \langle A(p+\eta e_i) - f, A(p+\eta e_i) - f\rangle_{\mathbb{R}^N} \\ &= \langle Ap - f, Ap - f\rangle_{\mathbb{R}^N} + 2\eta\langle Ap - f, Ae_i\rangle_{\mathbb{R}^N} + \eta^2\langle Ae_i, Ae_i\rangle_{\mathbb{R}^N} \end{aligned}$$

Thus, if we let $r = Ap - f$ and $a_i = Ae_i =$ the i-th column of A, then the cost function $F(\eta)$ becomes

$$\frac{1}{2}(|r|^2 + 2\eta\langle r, a_i\rangle_{\mathbb{R}^N} + \eta^2|a_i|^2).$$

For an interior solution $p_i + \eta > 0$, η needs to satisfy

$$\langle r, a_i\rangle_{\mathbb{R}^N} + \eta|a_i|^2 = 0,$$

i.e.

$$\eta^+ = \frac{-\langle r, a_i\rangle_{\mathbb{R}^N}}{|a_i|^2}.$$

Therefore, we have the update

$$p_i^+ = \max(0, p_i + \eta),$$

the projection of $p_i + \eta$ onto nonnegativity. It is important to observe that as each coordinate of p changes, so does the corresponding residual r. That is,

$$r^+ = Ap^+ - f = A(p+\eta e_i) - f = Ap - f + \eta a_i = r + \eta a_i.$$

In summary, we obtain the coordinate-descent in one sweep, Algorithm 5.2.

Algorithm 5.2 coordinate-descent method

$r = Ap - f$; for $i = 1 : M$

$$\begin{aligned} a &= A(:, i) \\ \eta &= (-r'a)/(a'a) \\ p_i &= \max(0, p_i + \eta) \\ r &= r + \eta a \end{aligned}$$

In general, we have noisy data f and we use the regularization formulation as before

$$\min_{p \geq 0} \ \frac{1}{2} |Ap - f|_2^2 + \alpha \sum_j p_j + \frac{\gamma}{2} \langle p, Hp \rangle_{\mathbb{R}^N}, \tag{5.9}$$

where $\alpha \geq 0$ is the regularization on sparsity in p and $\gamma \geq 0$ is the regularization parameter on the smoothness in p, H is the discrete negative Laplacian operator.

The new subproblem for each coordinate is

$$\begin{aligned} &\min_\eta && \tfrac{1}{2} |A(p + \eta e_i) - f|_2^2 + \alpha \textstyle\sum_j (p + \eta e_i)_j + \tfrac{\gamma}{2} \langle\, (p + \eta e_i), H(p + \eta e_i) \,\rangle_{\mathbb{R}^N}, \\ &\text{subject to} && p + \eta e_i \geq 0 \end{aligned}$$

We derive the method for this problem in the same manner as above. Thus,

$$\eta^+ = \frac{-\langle r, a_i \rangle_{\mathbb{R}^N} + \alpha + \gamma (Hp)_i}{|a_i|^2 + \gamma H_{i,i}}.$$

Remark 6. Currently, this CD-NMF method is effective for small-scale problems because of cost. Therefore, we omit numerics since big data is our interest. We include this method only as an alternative for the deconvolution problem. A greedy-sweep is often employed for coordinate-descent methods in order to improve speed. That is, future work involves developing an efficient CD-NMF for large-scale problems. □

5.5 Primal-Dual Active Set Method and Deconvolution

In this section, we develop a primal-dual active set method and investigate this algorithm as a deconvolution method.

Consider the optimization problem

$$\min_{p \geq 0} \ \frac{1}{2} |Ap - f|_2^2, \tag{5.10}$$

where $p \in \mathbb{R}^N$, $A \in \mathbb{R}^{N \times N}$, $f \in \mathbb{R}^N$. In addition, $f \geq 0$ and the atoms of A are nonnegative. We first state the necessary optimality. Let

$$L(p, \lambda) = \frac{1}{2}|Ap - f|_2^2 - \langle \lambda, p \rangle_{\mathbb{R}^N}$$

be the Lagrangian function and $\lambda \geq 0$ the Lagrange multiplier associated with the inequality constraint $p \geq 0$. Note, $\lambda \in \mathbb{R}^N$. Necessary optimality is given by

$$A'(Ap - f) - \lambda = 0, \quad \lambda \geq 0 \text{ and } \langle \lambda, p \rangle_{\mathbb{R}^N} = 0.$$

Recall that the complementarity condition determines the primal-dual active set method. That is,

$$\lambda = \max(0, \lambda - cp).$$

Once again, we let $c = 1$. This determines the active and inactive sets

$$\mathcal{A} = \{i \,|\, (\lambda - p)_i > 0\}, \quad \text{where } p_{\mathcal{A}} = 0$$

and

$$\mathcal{I} = \{i \,|\, (\lambda - p)_i \leq 0\}, \quad \text{where } \lambda_{\mathcal{I}} = 0.$$

Thus, the primal-dual active set method solves the following system for λ^+ given the current p:

$$A'Ap - A'y - \lambda^+ = 0 \tag{5.11}$$

where $p_{\mathcal{A}} = 0$. That is, $\lambda^+_{\mathcal{A}} = (A'Ap - A'y)_{\mathcal{A}}$. And for p^+, we solve

$$(A'Ap^+ - A'y)_{\mathcal{I}} = 0 \tag{5.12}$$

where $\lambda^+_{\mathcal{I}} = 0$. The second system (5.12) gives

$$(A'Ap^+)_{\mathcal{I}} = (A'y)_{\mathcal{I}}.$$

We solve this system by the iterative method conjugate gradient [67]. Note, we use this as an incomplete solver. In Figure 5.6d, we have plotted the CG residuals versus the number of iterates for results in Figure 5.6c. More specifically, we record the residuals for 5 PDAS iterations, or 50 CG iterations. We can see for each PDAS iteration, the CG method significantly reduces the residual within 10 iterations. Thus, it effectively finds an estimate p_{CG} for a given p at the i-th iteration of PDAS.

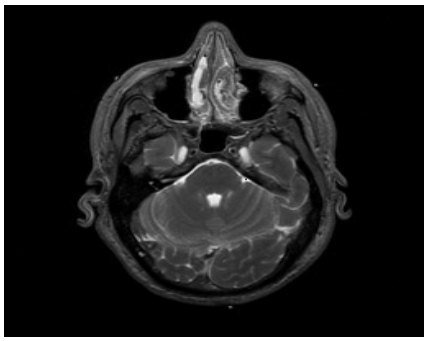
Remark 7. The atoms of A are known. Therefore, the construction of A isn't necessary since this may be costly for N large. The following numerics result from the PDAS and the atoms of A. □

As we stated earlier, generally we have noisy data. Therefore, we use the regularization formulation (5.9) once again. Thus, we modify the systems:

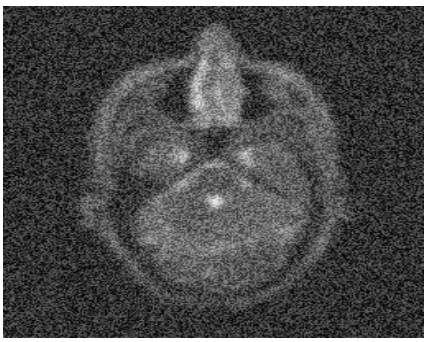
$$\begin{aligned} &\lambda^+ = (A'Ap - A'y + \alpha e + \gamma Hp), \quad p_{\mathcal{A}} = 0 \\ &(A'Ap^+ - A'y + \alpha e - \gamma Hp)_{\mathcal{I}} = 0 \end{aligned}$$

where e is a vector of ones.

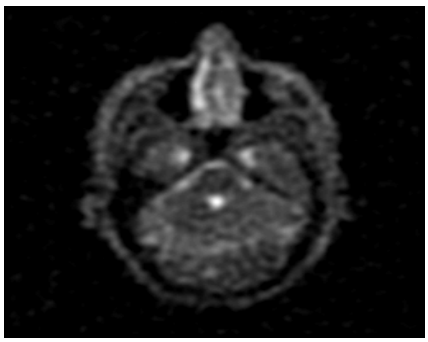
Reconsider the brain scan Figure 5.6a. We convolve the image with the same point-spread



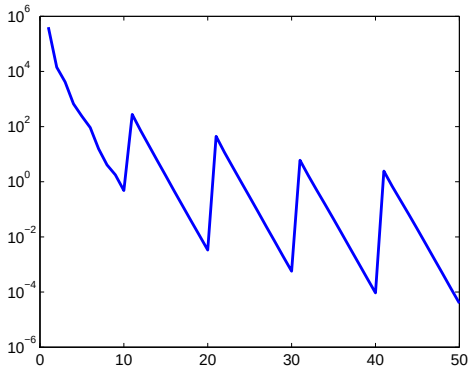
(a) Truth



(b) Convoluted and Noise



(c) PDAS $(\alpha, \gamma) = (70, 10^{-5})$



(d) CG Residuals

Figure 5.6: Brain and PDAS

function as before, i.e. a 9-by-9 PSF with local Gaussian distribution and standard deviation 2 (15 atoms) using 'imfilter(·)'. In addition, we add the same degree of relative noise (53%). We do not construct the operator A. We take advantage of the atoms of A and convolve instead. In Figure 5.6c, we display regularized results for $(\alpha, \gamma) = (70, 10^{-5})$. The resulting p has fidelity 0.0981 to Figure 5.6a.

In summary, we have developed a primal-dual active set method that finds a representing p for a large $N = 65536$. In addition, we've shown that the PDAS method is successful as a deconvolution method.

5.6 Conclusion

In this chapter we have shown that the divergence NMF can be used as not only a deconvolution technique, but it can also be used in the blind deconvolution problem. We have shown with the concept of the atoms that we can obtain a better estimate for both the ground truth and point-spread function. Future work involves developing a more general algorithm for the atomic deconvolution. Currently, we can only address point-spread functions of local Gaussian distribution for sizes $n \times n$ of odd $n = 3 : 9$.

Moreover, we have introduced the coordinate-descent NMF method and the primal-dual active set method as deconvolution methods, i.e. we know the convolutional operator A. Future work involves developing a greedy coordinate-descent method, increasing speed and expanding implementability to large systems. In addition to advancing the CD-NMF, we will adjust the deconvolution PDAS-NMF for the general NMF problem in order to eliminate the costly system solve.

Chapter 6

Triple Nonnegative Matrix Factorization

In this chapter, we examine the triple Nonnegative Matrix Factorization ASP and its current application. We introduce a new conceptual approach to the triple NMF by enforcing sparsity on S. The nonzeros highlight essential components in A and P. In addition, we discuss the special case of the triple NMF, a symmetric Nonnegative Matrix Factorization.

6.1 Introduction

Consider the representation for data $Y \in \mathbb{R}^{n \times m}$:

$$Y \approx ASP, \tag{6.1}$$

where $A \in \mathbb{R}^{n \times k}$, $P \in \mathbb{R}^{p \times m}$, $S \in \mathbb{R}^{k \times p} \geq 0$ element-wise, and $k, p < \min(n, m)$. Note, for the purpose of this chapter, we will let $k = p$.

This representation is the triple Nonnegative Matrix Factorization (tNMF), and it is used in few applications. For example, in text mining, near-orthogonality is asked of both A and P. The purpose is to obtain clusters in the rows of P and columns of A to induce more meaning

from the data Y. S introduces degrees of freedom in order to attain this restricted model while maintaining goodness of fit [20, 21, 47].

The optimization problem for the example is

Problem 6.1. *Given a data matrix* $Y \in \mathbb{R}^{n\times m}$ *and an integer* $p < \min(n, m)$*, find the matrices* $A \in \mathbb{R}^{n\times p}$, $P \in \mathbb{R}^{p\times m}$, $S \in \mathbb{R}^{p\times p} \geq 0$ *such that*

$$\min_{A,\,S,\,P} \ \frac{1}{2}\,||ASP - Y||_F^2 + \beta_1 \sum_{k,\hat{k}} (A'A - I)_{k,\hat{k}} + \beta_2 \sum_{k,\hat{k}} (PP' - I)_{k,\hat{k}} \tag{6.2}$$

$$\textit{subject to } A,\ P \geq 0,\ S \geq 0,\ A'A = I,\ PP' = I$$

where the first term is the fidelity of the triple representation to Y. The second term is the regularization of the near-orthgonality of the columns of A by β_1. As β_1 increases, the sum of the off-diagonals of $A'A$ should decrease. Similarly, the last term is the regularization of the near-orthogonality of the rows of P by β_2. As β_2 increases, the sum of the off-diagonals of PP' should decrease. Observe, without the near-orthogonality regularization of A, the triple NMF reduces to the bilinear NMF if we let $A = AS$. Therefore, the constraint is necessary to study triple NMF.

In this chapter, we will modify Problem 6.1 based on a sparse $S \in \mathbb{R}^{p\times p}$ whose nonzero entries highlight the essential tensorial components of A and P. For instance,

$$A \begin{bmatrix} 0 & s_{1,2} & 0 & 0 & 0 \\ 0 & 0 & 0 & s_{2,4} & 0 \\ s_{3,1} & s_{3,2} & 0 & 0 & s_{3,5} \\ s_{4,1} & 0 & 0 & 0 & 0 \\ 0 & 0 & s_{5,3} & s_{5,4} & 0 \end{bmatrix} P.$$

Alternatively, given this new concept we can then write (6.1) component-wise as

$$Y_{i,j} \approx s_{i,j}(A_{*i} \otimes P_{j*}). \tag{6.3}$$

In the remainder of this chapter, we discuss the tNMF approach in text mining and our new approach. In addition, we discuss a special case of the tNMF called the symmetric NMF.

6.2 Triple Nonnegative Matrix Factorization

In this section, we will describe the derivation for the unregularized triple NMF and the regularized triple NMF according to Problem 6.1.

Consider the case when $\beta_1 = 0$ and $\beta_2 = 0$. We then have the problem

$$\min_{A,\,S,\,P} \ \frac{1}{2}\,||ASP - Y||_F^2 \tag{6.4}$$

where A, P, S are subject to nonnegativity. The NMF algorithm for this problem is similar to the standard NMF in the following manner. For the A-update, replace P by SP. For the P-update, replace A by AS. We see in a moment this gives us the A- and P-updates for the triple NMF. However, the tNMF differs by the third factor.

More specifically, we can derive the unregularized tNMF algorithm by the scaled element-wise gradient descent updates

$$\begin{aligned} A &\longleftarrow A - \eta_A \tfrac{\partial J}{\partial A}, \\ P &\longleftarrow P - \eta_P \tfrac{\partial J}{\partial P}, \\ S &\longleftarrow S - \eta_S \tfrac{\partial J}{\partial S}. \end{aligned}$$

The partial derivatives of the cost function (6.4) are

$$\begin{aligned} \tfrac{\partial J}{\partial A} &= (ASP - Y)(SP)', \\ \tfrac{\partial J}{\partial P} &= (AS)'(ASP - Y), \\ \tfrac{\partial J}{\partial S} &= A'(ASP - Y)P'. \end{aligned}$$

We choose the scaling step-sizes

$$\begin{aligned}
\eta_A &= A./(\, A(SP)(SP)^{'}\,) \\
\eta_P &= P./(\,(AS)^{'}(AS)P\,) \\
\eta_S &= S./(\, A^{'}(ASP)P^{'}\,).
\end{aligned}$$

We can substitute into the gradient method to obtain Algorithm 6.1. We have descent by extension of the Seung-Lee proof.

Algorithm 6.1 triple NMF (tNMF)

$$A \leftarrow A. * (\, Y(SP)'\,)./(\, A(SP)(SP)'\,)$$

$$P \leftarrow P. * (\,(AS)'Y\,)./(\,(AS)'(AS)P\,)$$

$$S \leftarrow S. * (\, A'YP'\,)./(\, A'(ASP)P'\,)$$

As we stated earlier, if only the nonnegativity constraint exists on A, then the triple NMF reduces to the original bilinear form by letting $A = AS$ [21]. Therefore, the near-orthogonality constraint is necessary for the triple NMF. Recall Problem 6.1. To obtain Algorithm 6.2, we use the scaled element-wise gradient update defined earlier. We modify the partial derivatives based on cost function 6.2:

$$\begin{aligned}
\tfrac{\partial J}{\partial A} &= (ASP - Y)(SP)^{'} + \beta_1(AE - A), \\
\tfrac{\partial J}{\partial P} &= (AS)^{'}(ASP - Y) + \beta_2(EP - P), \\
\tfrac{\partial J}{\partial S} &= A^{'}(ASP - Y)P'.
\end{aligned}$$

We also modify our choice in scaling step-sizes

$$\begin{aligned}
\eta_A &= A./(\ A(SP)(SP)^{'} + \beta_1(AE - A)\ ) \\
\eta_P &= P./(\ (AS)^{'}(AS)P + \beta_2(EP - P)\ ) \\
\eta_S &= S./(\ A^{'}(ASP)P^{'}\ )
\end{aligned}$$

Substitution into the gradient method gives Algorithm 6.2. We also have the L^2 projection, as

Algorithm 6.2 least squares tNMF with near-orthgonality

$$\begin{aligned}
A &\leftarrow A. * (\ Y(SP)'\ )./(\ (\ A(SP)(SP)'\ ) + \beta_1(AE - A)\ ) \\
A &\leftarrow \sqrt{diag(\ diag(\ A'A\ )\ )}^{-1} \\
P &\leftarrow P. * (\ (AS)'Y\ )./(\ (\ (AS)'(AS)P\ ) + \beta_2(EP - P)\ ) \\
P &\leftarrow \sqrt{diag(\ diag(\ PP'\ )\ )}^{-1} \\
S &\leftarrow S. * (\ A'YP'\ )./(\ A'(ASP)P'\ )
\end{aligned}$$

in chapter 3, for A and P.

Recall the divergence criterion. We derive a triple NMF for

$$D(Y||ASP) = \sum_{i,j}(Y_{i,j}\log(\frac{Y_{i,j}}{(ASP)_{i,j}}) - Y_{i,j} + (ASP)_{i,j}).$$

If we replace the Frobenius norm in Problem 6.1 with this criterion, we obtain a new variational formulation. Algorithm 6.3 is derived similarly to the triple Frobenius NMF in Algorithm 6.2. We include the regularization, but one can obtain the unregularized divergence tNMF by setting β_1, $\beta_2 = 0$ and removing the projection steps for A and P.

Algorithm 6.3 divergence tNMF with near-orthgonality

$$
\begin{aligned}
A &\leftarrow A. * \left(\frac{(SP)'(Y./\,(ASP)\,)}{\sum_j (SP)_{k,j} + \beta_1(AE - A)} \right) \\
A &\leftarrow \sqrt{diag(\,diag(\,A'A\,)\,)}^{-1} \\
P &\leftarrow P. * \left(\frac{(AS)'(Y./\,(ASP)\,)}{\sum_i (AS)_{i,k} + \beta_2(EP - P)} \right) \\
P &\leftarrow \sqrt{diag(\,diag(\,PP'\,)\,)}^{-1} \\
S &\leftarrow S. * \left(\frac{A'(Y./(ASP))P'}{\sum_i A_{i,k} \sum_j P_{k,j}} \right)
\end{aligned}
$$

6.3 Conceptual Advancement

We will now review the discussion from chapter 4 about the NMF-singular values. First, recall the SVD for rank $r = \min(n, m)$ data matrix Y:

$$Y = U\Sigma V',$$

where Σ is the matrix whose diagonal entries are the singular values ordered as $\sigma_1 > \sigma_2 > \cdots > \sigma_r \geq 0$, and U, V are the real, orthonormal matrices with corresponding singular vectors.

Previously, we considered the rows of V' as a basis for the SVD of Y, where $U\Sigma$ is an assignment of that basis. For the NMF, we consider the rows of P to be the basis and A to be

its assignment. This allows us to relate the SVD to the NMF in the following manner:

rows of V' basis for SVD	–	rows of P basis for NMF
$U\Sigma$ assignment of SVD	–	A assignment of NMF

Based on this connection, we defined the NMF-singular values

$$s_k = ||A(:,k)||_1 \quad \text{for } k = 1, \ldots, \bar{p},$$

where $\bar{p} = \min(n, m)$. These values determine which rows of P contribute the most in the representation AP.

We want to modify the connection for the triple NMF. Consider Problem 6.1. As we stated, the near-orthogonality constraint on both A and P creates a partition of clusters in columns and rows, respectively. We now introduce a new regularized problem

Problem 6.2. *Given a data matrix $Y \in \mathbb{R}^{n\times m}$ and an integer $p < \min(n, m)$, find the matrices $A \in \mathbb{R}^{n\times p}$, $P \in \mathbb{R}^{p\times m}$, $S \in \mathbb{R}^{p\times p} \geq 0$ such that*

$$\min_{A,\,S,\,P} \quad \tfrac{1}{2}\,||ASP - Y||_F^2 + \alpha||S||_1 + \beta_1 \textstyle\sum_{k,\hat{k}}(A'A - I)_{k,\hat{k}} + \beta_2 \sum_{k,\hat{k}}(PP' - I)_{k,\hat{k}} + \frac{\rho}{2}|S|^2 \qquad (6.5)$$

$$\textit{subject to } A \geq 0,\ P \geq 0,\ S \geq 0,\ A'A = I, PP' = I$$

When α, β_1, β_2, $\rho = 0$, we have cost function (6.4). For (6.5), the first, third and fourth terms are previously explained for cost function (6.2). The second term is the regularization of the sparsity of S by α, where sparsity is measured by the ℓ_1-norm of S. As α increases, the sum will decrease. The significance of sparsity on S is to highlight the essential components (6.3) of Y in the triple NMF representation. In addition to the sparsity regularization, we have the additional penalty term $\frac{\rho}{2}|S|^2$ where ρ is the penalty parameter. Because A and P are regularized to be normalized in columns and rows respectively, entries in S will grow in magnitude. In order to prevent the entries from being too large, the penalty term will control the magnitude of nonzeros in S.

Given this new concept for the triple NMF, we can now make a stronger connection between the NMF and the SVD.

rows of V' clusters for SVD	–	rows of P clusters for NMF
colns of U clusters for SVD	–	columns of A clusters for NMF
σ_k contribution scale	–	s_k magnitude of contribution

Observe, we allow the position of the nonzeros to be determined by α rather than enforcing the nonzeros to be on the diagonal of S.

For Problem 6.2, we derive Algorithm 6.4 for the least squares and Algorithm 6.5 for divergence. We derive these Algorithms in the same manner as Algorithms 6.2 and 6.3, therefore we omit the discussion.

Algorithm 6.4 least squares tNMF with α

$$
\begin{aligned}
A &\leftarrow A. * (Y(SP)')./ (\, (A(SP)(SP)') + \beta_1(AE - A) \,) \\
A &\leftarrow \sqrt{diag(\, diag(\, A'A \,) \,)}^{-1} \\
P &\leftarrow P. * (\, (AS)'Y \,)./ (\, (\, (AS)'(AS)P \,) + \beta_2(EP - P) \,) \\
P &\leftarrow \sqrt{diag(\, diag(\, PP' \,) \,)}^{-1} \\
S &\leftarrow S. * (\, A'YP' \,)./ (\, (\, A'(ASP)P' \,) + \alpha E + \rho S \,)
\end{aligned}
$$

6.4 Application: Text Clustering

In this section, we will use the generalized Kullback-Leibler (K-L) divergence

$$D(Y||ASP) = \sum_{i,j} (Y_{i,j} \log(\frac{Y_{i,j}}{(ASP)_{i,j}}) - Y_{i,j} + (ASP)_{i,j}) \tag{6.6}$$

Algorithm 6.5 divergence tNMF with α

$$
\begin{aligned}
A &\leftarrow A. * \left(\frac{(SP)'(Y./(ASP))}{\sum_j (SP)_{k,j} + \beta_1(AE - A)} \right) \\
A &\leftarrow \sqrt{diag(diag(A'A))}^{-1} \\
P &\leftarrow P. * \left(\frac{(AS)'(Y./(ASP))}{\sum_i (AS)_{i,k} + \beta_2(EP - P)} \right) \\
P &\leftarrow \sqrt{diag(diag(PP'))}^{-1} \\
S &\leftarrow S. * \left(\frac{A'(Y./(ASP))P'}{\sum_i A_{i,k} \sum_j P_{k,j} + \alpha E + \rho S} \right)
\end{aligned}
$$

as the fidelity criterion for the triple NMF and analyze the medlars dataset, which is a dataset of medical abstracts[1]. The dataset contains 1,033 documents and 5,831 terms. It can be found in a file with format Harwell-Boeing for sparse data. We choose middle rank $p = 15$ by a method described in chapter 4. We want to show we are given the essential components of Y in the representation through the analysis of a sparse S.

To analyze the clusters, there are two types of evaluations. The first type is external evaluation. This is useful when you have some prior knowledge about the clustering results, such as class labels. Hence, the clustering success is determined by the amount of information correctly corresponding to the expected information in a particular cluster. Criterion used for external evaluations include purity and entropy [21, 32].

The second type of validation is internal evaluation. When prior knowledge is unknown, these measures will determine success of the clustering by evaluating the within-cluster matrix

[1] http://web.eecs.utk.edu/research/lsi/

and the between-cluster matrix [24, 52, 53]. One such measure is called the Calinski-Harabasz (CH) index. We first define the within-cluster matrix

$$W = \sum_{k=1}^{p} (C_k - \mu_k)(C_k - \mu_k)',$$

where C_k is cluster k and μ_k is the mean of cluster k. The between-cluster matrix is defined

$$B = \sum_{k=1}^{p} n_k(\mu_k - \mu)(\mu_k - \mu)',$$

where μ is the mean of the entire set of clusters and n_k is the number of sample in cluster C_k. The CH index finds the following ratio

$$CH = \frac{trace(B)}{(p-1)} \frac{(n-p)}{trace(W)}.$$

The smaller this value, the better the clustering. That is, we want the similarity between clusters to be smaller and the similarity among points within a cluster to be larger. We will use the CH index for the medlars database since we do not have specified classes.

For the medlars dataset, A will represent clusters on the documents, and P will represent clusters on the terms. We do have labels on the terms, so we have additional validation by hand in Table 6.1, which shows the success of each cluster by inferring topics. We determine the most significant clusters in P by the nonzeros in Z, where Z is the sparse S. For this experiment, we choose the p-largest entries in S to be the nonzeros in Z. We then look at the largest entries within a cluster and the corresponding terms (we look at the entries whose values are larger than half of the max entry). We use these terms to extract a topic. For the table, we look at the 5 most significant clusters in P and then give at most ten terms.

For A, unlike P, we do not have have labels for the documents. Hence, we cannot analyze success by comparison. We do, however, show 2 results. First, we use the regularization term measure $\sum_{k,\bar{k}} (A'A - I)_{k,\bar{k}}$ to determine how close to the identity is $A'A$. This will provide us

Table 6.1: Topics Determined from Clustered Terms

Cluster	# significant terms	terms	topic determined
1	7	cells, alveolar, marrow, electron, lung, epithelial	bone marrow and lungs
2	5	homormone, growth, serum, dna, glucose	growth horomone
3	13	breast, insipidus, diabetes, renal, excretion, urinary, therapy, treatment, steroid, cancer, sodium	breast cancer and diabetes insipidus
4	22	antigens, lens, protein, crystallin, fractions, gel, cell, soluble, molecular, human	lens proteins (methods for separation)
5	17	fatty, acids, rat, fetal, kidney, rats, glucose, acid, content, increased, hour, normal	glucose and fatty acides in kidneys

with the degree of distinctness of clusters. Second, we use the internal cluster validation measure Calinski H. The smaller this value the better the clusters.

In Table 6.1, we display the first 5 clusters for regularized results. In addition, we include Table 6.2. This table includes a base case for unregularized A, P results compared to the regularized A, P results. Recall our earlier statement, however. Regularization on A and P are necessary for triple NMF to matter.

For the regularization on A and P, we choose $\beta_1 = 5$ and $\beta_2 = 20$ in order to increase the partitioning in both A and P as compared to the unregularized partitions. Note, we let $\rho = 0$ because values in Y are small enough. We see in Table 6.1 that term clusters in P suc-

Table 6.2: Clustering Numerics

α	ρ	β_1	$\frac{\sum_{k,\bar{k}}(A'A-I)_{k,\bar{k}}}{p^2}$	β_2	$\frac{\sum_{k,\bar{k}}(PP'-I)_{k,\bar{k}}}{p^2}$	calinski(A)	calinski(P)	fidelity
0	0	0	0.0287	0	0.2313	0.0110	0.2120	8.7176e4
15	0	5	0.0279	20	0.1994	0.0181	0.1260	8.9005e4

cessfully extract topics. We get the quantities $\frac{\sum(PP'-I)_{k,\bar{k}}}{p^2} = 0.1994$ and $\frac{\sum(A'A-I)_{k,\bar{k}}}{p^2} = 0.0279$. These values are certainly smaller than the unregularized quantities. The Calinski measures are calinski$(A) = 0.0181$ and calinski$(P) = 0.1260$.

In Table 6.2, we state the fidelity values for both unregularized and regularized for this example. However, we note that given that the we look for a representation with structural properties of partitions and sparsity, the divergence gap between Y and ASP should be large.

The triple NMF has been used in text mining [21, 47] as a clustering algorithm. However, we take advantage of the added sparse-ness in S to further highlight clusters. This not only provides direction in analysis but it also allows us to store less information, i.e. store only the essential tenor products $s_{i,j}(A_{*,i} \otimes P_{j,*})$.

6.5 Application: Image Compression

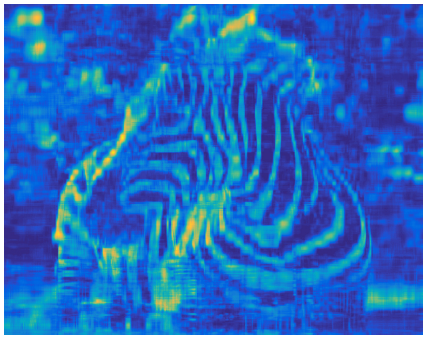
In this section, we discuss the sparse triple NMF for image compression. The nonzero elements in S highlight essential clusters in A and P. The information retained in the clusters of A and P is most significant.

In Figure 6.1a, we have an image 801 pixels by 901 pixels with the distinct pattern of the zebra stripes [4]. We note the following: first, we will begin with $p = 90$, which is chosen as the optimal choice for middle rank according to the method developed in chapter 4. Secondly, we use the Frobenius tNMF. Lastly, results shown in this section use the same number of iterations and the same initializations for A, S, and P.

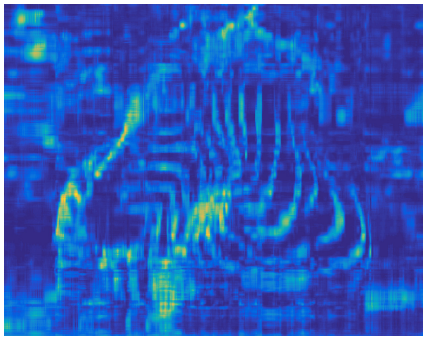
In Figures 6.1b-6.1d, we keep constant $\beta_1 = 1$ and $\beta_2 = 1$ since we want to demonstrate the influence of sparsity on representation. Additionally, we find for the tNMF in this application, $\rho \neq 0$ is necessary for a good representation, which we discuss in more detail. We have results in Table 6.3 for Figure 6.1.

In the Table, the cases where $\rho = 0$ result in a highly sparse S. This is desirable for our purpose of maintaining only the most essential tensorial components of A and P. However,

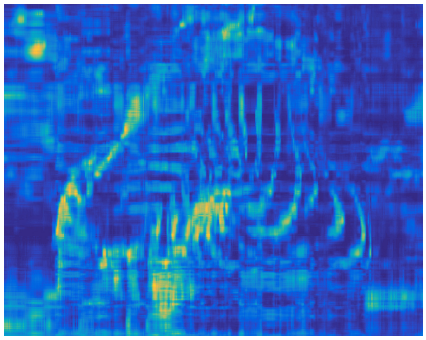
(a) Truth



(b) $(p, \beta_1, \beta_2, \alpha, \rho) = (90, 1, 1, 0, 0)$



(c) $(p, \beta_1, \beta_2, \alpha, \rho) = (90, 1, 1, 250, 1)$



(d) $(p, \beta_1, \beta_2, \alpha, \rho) = (90, 1, 1, 500, 1)$

Figure 6.1: Zebra

Table 6.3: Numerics for Zebra

α	ρ	$\frac{\sum_{k,\bar{k}}(A'A-I)_{k,\hat{k}}}{p^2}$	$\frac{\sum_{k,\bar{k}}(PP'-I)_{k,\hat{k}}}{p^2}$	% sparsity(S)	$\frac{\|\|ASP-Y\|\|_F}{\|\|Y\|\|_F}$
0	0	0.1113	0.1708	98.22	0.2441
0	1	0.0492	0.0675	95.65	0.2875
500	0	0.1071	0.1649	98.40	0.2472
500	1	0.0537	0.0796	96.17	0.3223
250	1	0.0523	0.0753	96.05	0.3046

we also want more partitioned clusters in A and P since β_1, $\beta_2 \neq 0$. We can clearly see that given $\rho = 1$, we achieve a better representation in terms of the near-orthogonality in A and in P. While we lost approximately 2 percent sparsity, we have not lost a significant amount. Therefore, a proper balance is achieved when the penalty is enforced.

We now examine Figures 6.1c and 6.1d, where both representations are results of $\rho = 1$ and where we use $\alpha = 250, 500$ respectively. Although α has increased from 0 to 500, causing S to be extremely sparse, we can see in these Figures, that we still maintain the most significant features in the image. That is, the zebra stripes are the distinct characteristic of Figure 6.1a, and there is still a trace in Figure 6.1d.

In this section, we show through an example that tNMF still captures the most important features of an image and the sparsity enhancement on S allows us to use less storage.

6.6 Symmetric Nonnegative Matrix Factorization

A special case of the Triple Nonnegative Matrix Factorization (tNMF) is the Symmetric Nonnegative Matrix Factorization (sNMF). There are many data matrices that arise in practical applications that are symmetric, e.g. covariance matrices, the network Laplacian operator [41]. A purpose of the sNMF is to find a symmetric representation that highlights the significant features. We propose a sNMF as the symmetric representation for this data compression. In this section, we will begin with a discussion of a bilinear sNMF and progress to the triple sNMF.

The problem of the symmetric Nonnegative Matrix Factorization is

Problem 6.3. *Given a symmetric matrix $Y \in \mathbb{R}^{n\times n}$ and the positive integer $p < n$, find an $M \in \mathbb{R}^{n\times p}$ that optimizes*

$$\begin{array}{ll} \min_M J(M) = & \min_M \ \frac{1}{2}\,||MM' - Y||_F^2 \\ \textit{subject to} & M \geq 0 \end{array}$$

Algorithm 6.6 (to find a representation for Problem 6.3) is derived in the same manner as the standard NMF. That is, consider the scaled element-wise gradient method

$$M \leftarrow M - \eta_M \frac{\partial J}{\partial M}$$

with scaled step-size and derivative

$$\eta_M = M./(MM'M), \quad \frac{\partial J}{\partial M} = (M'M - Y)M.$$

Substitution gives us the update presented in Algorithm 6.6, with the added relaxation step discussed in chapters 3 and 5.

Algorithm 6.6 sNMF

$$\begin{array}{rcl} m & \leftarrow & M. * (YM)./(MM'M) \\ M & \leftarrow & (1-t)M + tm \end{array}$$

However, as was the case for the bilinear NMF with orthogonality constraints, we need to introduce degrees of freedom on this symmetric bilinear case in order to maintain accuracy. In addition, consider the Cholesky Decomposition. This is an exact bilinear factorization $Y = LL'$ which requires Y to be square and symmetric positive-definite. The LDL'-factorization introduces a diagonal matrix in order to avoid extracting square roots in computations. Moreover, matrices for which the Cholesky factorization does not exist can still have LDL'-factorization where D absorbs the negative signs [26, 40, 55].

We now introduce a diagonal matrix in order to provide degrees of freedom to the representation. We introduce D, a diagonal matrix, without the constraint of nonnegativity which allows D to absorb signs, securing a more accurate representation.

Problem 6.4. *Given a symmetric matrix $Y \in \mathbb{R}^{n\times n}$ and the positive integer $p < n$, find an $M \in \mathbb{R}^{n\times p}$ and $D \in \mathbb{R}^{p\times p}$ that optimizes*

$$\begin{array}{ll} \min_{M,D} J(M,D) = & \min_{M,D} \ \frac{1}{2}\,||MDM' - Y||_F^2 \\ \textit{subject to} & M \geq 0 \end{array}$$

where D is a diagonal matrix.

Algorithm 6.7 MDM'-NMF

$$\begin{array}{rcl} M & \leftarrow & M.*\,(YMD)./(\,MDM'\,MD) \\ \ell & \leftarrow & \ell.*\,(\,diag(M'YM)./diag(M'\,(MDM')\,M)\,) \end{array}$$

As we mentioned, the degrees of freedom introduced by D are integral to the accuracy of a symmetric representation. Let ℓ be the vector containing the diagonal entries. Recall, in Matlab, diag$(\cdot)$ is a function that takes a vector and makes a diagonal matrix, e.g. $D = \text{diag}(\ell)$, or takes a matrix and extracts its diagonal entries into a vector. For Problem 6.4, we develop Algorithm 6.7.

Let $J(M,D) = \frac{1}{2}||MDM' - Y||_F^2$, where D is a diagonal matrix. The element-wise gradient descent methods with scaling are

$$\begin{aligned} M &\leftarrow M - \eta_M \tfrac{\partial J}{\partial M} \\ \ell &\leftarrow \ell - \eta_\ell \tfrac{\partial J}{\partial \ell}. \end{aligned}$$

For M,

$$\eta_M = M./(MDM'\,(MD)), \quad \frac{\partial J}{\partial M} = (MDM' - Y)MD.$$

Thus,

$$M \leftarrow M. * (YMD)./(MDM'(MD))$$

in order to obtain the update in Algorithm 6.7.

We make several observations in order to develop the update for ℓ. Recall now from chapter 3 the inner product $\langle X, V \rangle_F = \sum_{i,j} X_{i,j} \cdot V_{i,j}$. Also observe

$$\begin{aligned} [M'(MDM')M]_{k,\hat{k}} &= \textstyle\sum_{i,j} M_{i,k}(\sum_t M_{i,t}D_{t,t}M_{j,t})M_{j,\hat{k}} \\ [\,M'YM\,]_{k,\hat{k}} &= \textstyle\sum_{i,j} M_{i,k}Y_{i,j}M_{j,\hat{k}}. \end{aligned}$$

In addition, $D = (\ell_1 E_{1,1} + \ell_2 E_{2,2} + \cdots + \ell_p E_{p,p}) = \sum_k \ell_k E_{k,k}$, where $E_{i,j} \in \mathbb{R}^{p \times p}$ is the unit matrix with 1 in the (i,j)-entry and 0 elsewhere. Therefore, $MDM' = M(\sum_k \ell_k E_{k,k})M' = \sum_k \ell_k (ME_{k,k}M')$. Thus, the $\frac{\partial J}{\partial \ell_{\hat{k}}}$ we obtain is

$$\begin{aligned} \tfrac{\partial J}{\partial \ell_{\hat{k}}} &= \langle\, (MDM' - Y)\,,\, (ME_{\hat{k},\hat{k}}M')\,\rangle_{\mathbb{R}^{n\times n}} \\ &= \langle\, (MDM')\,,\, (ME_{\hat{k},\hat{k}}M')\,\rangle_{\mathbb{R}^{n\times n}} - \langle\, Y\,,\, (ME_{\hat{k},\hat{k}}M')\,\rangle_{\mathbb{R}^{n\times n}} \\ &= \textstyle\sum_{i,j}[MDM']_{i,j} \cdot [ME_{\hat{k},\hat{k}}M']_{i,j} - \sum_{i,j}[Y]_{i,j} \cdot [ME_{\hat{k},\hat{k}}M']_{i,j} \\ &= \textstyle\sum_{i,j}(\sum_t M_{i,t}D_{t,t}M_{j,t}) \cdot (M_{i,\hat{k}}M_{j,\hat{k}}) - \sum_{i,j} Y_{i,j} \cdot (M_{i,\hat{k}}M_{j,\hat{k}}) \\ &= \textstyle\sum_{i,j} M_{i,\hat{k}}(\sum_t M_{i,t}D_{t,t}M_{j,t})M_{j,\hat{k}} - \sum_{i,j} M_{i,\hat{k}}Y_{i,j}M_{j,\hat{k}}. \end{aligned}$$

Notice, this is exactly equal to $[M'(MDM')M]_{k,\hat{k}} - [M'YM]_{k,\hat{k}}$ when $k = \hat{k}$. That is, when we are at the $\hat{k}$-th diagonal of $[M'(MDM')M]_{k,\hat{k}} - [M'YM]_{k,\hat{k}}$. Hence, we have

$$\frac{\partial J}{\partial \ell_{\hat{k}}} = \mathrm{diag}(M'(MDM')M) - \mathrm{diag}(M'YM).$$

With $\frac{\partial J}{\partial \ell_{\hat{k}}}$, we can now choose our diagonal-scale step-size to be

$$\eta_{\ell_k} = \frac{\ell_k}{[\mathrm{diag}(M'(MDM')M)]_k}.$$

Table 6.4: MM^T vs. MDM^T

algorithm	fidelity
sNMF	0.0760
MDM'-NMF	0.0398

We then attain the update

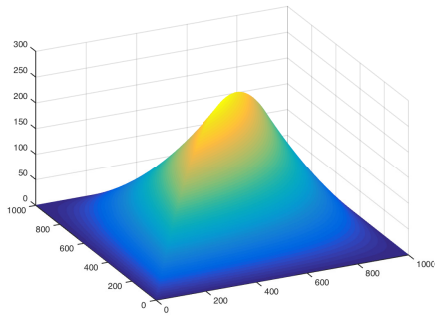
$$\ell \leftarrow \ell. * (\,\mathrm{diag}(M'YM))./(\mathrm{diag}(M'\,(MDM')\,M)\,).$$

Consider the 1000×1000 matrix representation of the inverse negative Laplacian operator in Figure 6.2a. This is a symmetric nonnegative matrix. We apply both the sNMF and the MDM′-NMF, for $p = 10$, and compare in Table 6.4. As we stated earlier, the purpose of these algorithms is to find a symmetric representation that still highlights significant features.

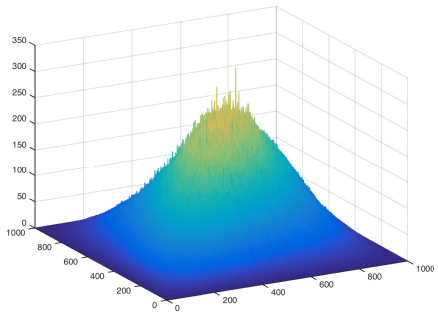
For the sNMF, we found $t = 0.5$ provides best results for this case. In addition, MDM'-NMF method does find a better representation in terms of fidelity, proving there is benefit with the degrees of freedom introduced by the D matrix. This is confirmed in Figures 6.2b and 6.2c, where the latter reduces the spikes seen in the former. Furthermore, to demonstrate the success of the clusters captured by MDM'-NMF, we provide the five clusters based on the dominant ℓ_k in Figure 6.2d. We note here we can regularize in order to obtain smooth structure.

In this next example, consider the social network presented in Figure 6.3a. Each node i represents a person i, and each link represents a friendship between person i and person j. There are two lines between each person since a friendship needs both people to participate. A matrix representation of this system will be symmetric. Specifically, we have each row indexed by each person, and each column indexed by each person. A link is represented by 1 and 0 will represent no link.

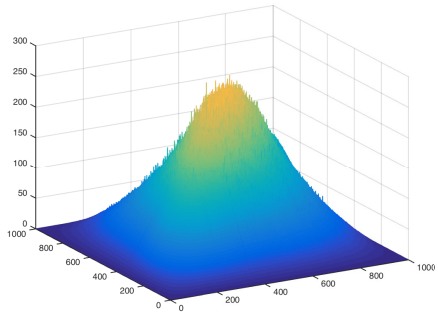
For this particular example, we know we need two distinct clusters, therefore we formulate



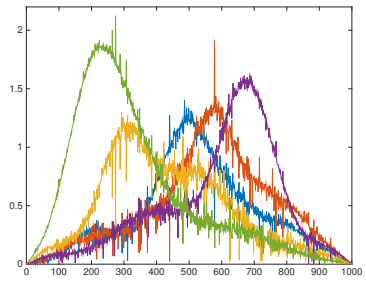
(a) Negative Laplacian



(b) MM'

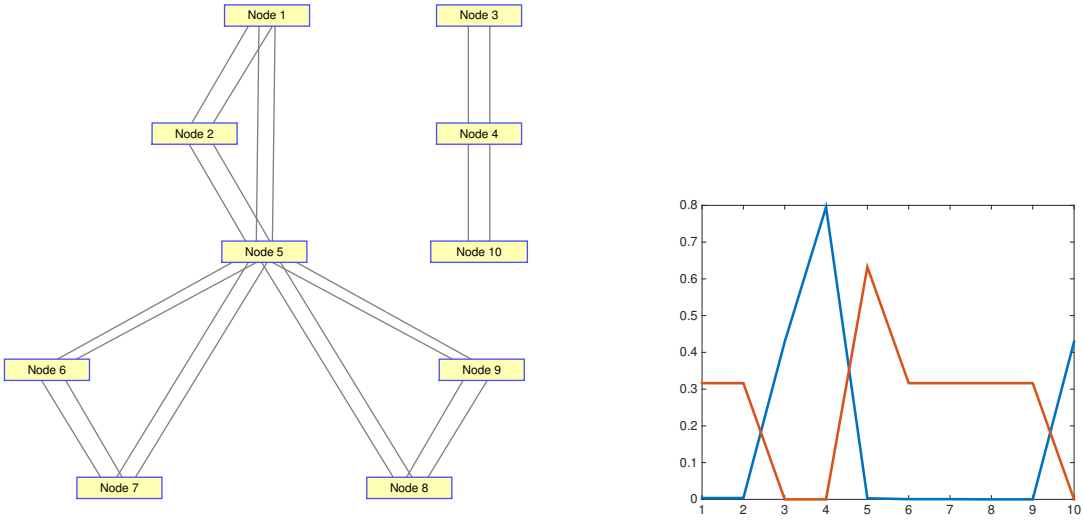


(c) MDM'



(d) MDM' Clusters

Figure 6.2: Symmetric Analysis



(a) Social Network (b) Clusters

Figure 6.3: Social Network and Clusters by MDM'

the regularized problem

$$\min_{M \geq 0} \frac{1}{2} ||MDM' - Y||_F^2 + \beta \sum_{k,\hat{k}} (M'M - I)_{k,\hat{k}}$$

subject to $M'M = I$ and D =diagonal. We will use Algorithm 6.7 with regularization in order to demonstrate further that we can extract meaningful clusters. That is,

$$\begin{aligned} \ell &\leftarrow \ell .* \left(diag(M'YM)./diag(M' \left(MDM'\right) M) \right) \\ M &\leftarrow M .* \left(YMD\right)./\left(MDM' MD + \beta(ME - M) \right) \\ M &\leftarrow \sqrt{diag\left(diag(M'M) \right)}^{-1} \end{aligned}$$

We have $p = 2$ and we let $\beta = 1.e - 4$. We receive the clusters in Figure 6.3b.

By the matrix D, we have that the orange cluster represents the most significant cluster. Studying this line, we see the orange cluster is associated with the larger friendship circle, where person 5 is the most popular and the points with 0-values are not in this friendship circle at all.

Then, examining the blue cluster, we see it corresponds to the smaller friendship circle, where person 4 is the most popular and again the points with 0-values are not in the friendship circle. We have successfully extracted meaningful information of this symmetric system by MDM'.

In summary, the symmetric NMFs presented in this section find a symmetric representation. The prevalence of symmetric data suggests a symmetric representation will have significance in the data analysis of these matrices.

6.7 Conclusion

In this chapter, we have introduced a new conceptual approach for the tNMF. That is, we constrain S to be a sparse matrix. Furthermore, we can interpret these weights as NMF-singular values for the triple representation. The nonzeros then highlight the most important components in A and P. In addition to this advancement, we have also introduced the concept of a symmetric triple NMF for the symmetric data that frequently appear in real life applications.

Future work involves applying the tNMF to the applications: wavelet compression, hyperspectral imaging [79], and neural networking, where the matrix representations are symmetric and symmetric NMFs can be used to cluster activity and identify patterns among neurons.

Chapter 7

Low-rank and Sparse Component Extraction via Nonnegative Matrix Factorization

In this chapter, we examine the decomposition of a data array into its low-rank and sparse components. Currently, the Principal Component Pursuit (PCP) is a convex program that finds this decomposition. We develop a method to find the alternative decomposition of low-rank, sparse, and noise components. This method takes advantage of the flexibility of the NMF. We implement the PCP-NMF in applications of shadow/light separation and foreground-background separation.

7.1 Introduction

Suppose we have a large data matrix $Y \in \mathbb{R}^{n \times m}$ and we know that it can be decomposed into

$$Y = L + S, \tag{7.1}$$

where $L \in \mathbb{R}^{n \times m}$ is a low-rank matrix and $S \in \mathbb{R}^{n \times m}$ is a sparse matrix [9, 14, 50, 80]. Both the augend and addend are unknown. That is, we do not know what is the rank of L, nor do we know the degree of sparsity of S and where the nonzeros are located.

The problem of finding the decomposition is formulated into the optimization problem

$$\min_{L,\,S} \; rank(L) + \lambda ||S||_0 \tag{7.2}$$

$$\text{subject to } Y - L - S = 0,$$

where $||S||_0$ counts the number of nonzeros in S and $\lambda > 0$ is a regularization parameter for the cardinality of S [9,50]. However, rank minimization problems and ℓ_0 optimization problems are computationally difficult to solve.

The Principal Component Pursuit (PCP) is a convex program that relaxes the optimization problem (7.2) to find the decomposition $Y = L + S$ via the nuclear norm and the ℓ_1-norm. Formally, that is,

$$\min_{L,\,S} \; ||L||_* + \lambda ||S||_1 \tag{7.3}$$

$$\text{subject to } Y - L - S = 0,$$

where $\lambda > 0$ is now the regularization parameter on the sparsity of S measured by $||S||_1 = \sum_{i,j} S_{i,j}$, and $||L||_* = \sum_k \sigma_k(L)$ is the nuclear norm defined by the sum of the singular values of L. The nuclear norm is often chosen in optimization as a convex relaxation for minimizing rank [15]. In addition, the ℓ_1-norm on S is often chosen in place of the ℓ_0-norm [66]. In [14], it is shown one can find the exact decomposition by PCP under three conditions. First, L is imposed to be not sparse. Second, S is imposed to be full-rank. Third, $\lambda = \frac{1}{\sqrt{\max(n,m)}}$. We refer you to [14] for more details.

There are limitations to PCP and optimization system (7.3), however. First, algorithms are computationally expensive. Second, in practical applications, there is noise corruption in every entry of data. We may also deal with missing data. Therefore, an exact decomposition of the

truth may not be available. [9]

A further relaxation of optimization problem (7.2) exists in [80]. Rather than aiming to find an exact $Y = L + S$, the method aims to find a fast approximation $Y = L + S + \Sigma$, where Σ is model error. The optimization problem is formulated as

$$\min_{L,S} \ ||(L+S) - Y||_F^2 \tag{7.4}$$

$$\text{subject to } \ rank(L) \leq r, \, card(S) \leq c$$

where both rank of L and cardinality of S are specified a priori by r and c, respectively. This reduces the amount of complexity in solving the problem. This method uses QR-factorizations for SVD approximations, reducing the computational complexity of the PCP algorithms thus increasing speed. However, the algorithm requires computing an $n \times n$ matrix, LL' at each iteration. As you will see in the following applications, n is significantly large, and this computation can be costly in terms of time and memory.

In this chapter, we examine a Nonnegative Matrix Factorization (NMF) algorithm as a method to find the PCP decomposition (7.1). The NMF has been successfully applied to many applications, such as clustering and classification, image processing, and gene pattern analysis [46, 47, 57], as a dimension reduction and feature enhancing data analysis tool. We are interested in extending this already-vast functionality to the PCP decomposition. We formulate an alternative relaxation for PCP (7.3)

$$\min_{A,\,P,\,S} \ \frac{1}{2} \, ||(AP + S) - Y||_F^2 + \alpha ||S||_1 \tag{7.5}$$

$$\text{subject to } A \geq 0, \, P \geq 0, \, Y \geq S \geq 0$$

where $\alpha \geq 0$ is the regularization parameter for sparsity on S and A, P are low-rank results from NMF. In the NMF algorithm, p is specified prior to implementation. However, we do not specify cardinality of S. Rather, we allow α to regularize the cardinality. Furthermore, system (7.5)

constrains A, P, S to have nonnegative elements. Therefore, producing results that capture the essential structure and localized features. In the following sections, we will discuss the method development and implementation of PCP-NMF.

7.2 Algorithm Development

The Nonnegative Matrix Factorization (NMF) is a representation of the data matrix $Y \in \mathbb{R}^{n\times m}$

$$Y \approx AP,$$

where $A \in \mathbb{R}^{n\times p}$, $P \in \mathbb{R}^{p\times m}$, $A,\ P \geq 0$ element-wise, and $p < \min(n, m)$ [5, 65].

To find such a representation, scientists Lee and Seung optimize the fidelity function

$$\min_{A,\,P}\ J(A,P) = \min_{A,\,P}\ \frac{1}{2}\,||AP - Y||_F^2$$

subject to the nonnegativity constraint on $A,\ P \geq 0$. We are interested in finding the representation

$$Y \approx AP + S$$

where $S \geq 0$ is a sparse matrix. Thus, we formulate

Problem 7.1. *Given a data matrix $Y \in \mathbb{R}^{n\times m}$ and the integer $p < \min(n, m)$, find an $A \in \mathbb{R}^{n\times p}$, $P \in \mathbb{R}^{p\times m}$, and $S \in \mathbb{R}^{n\times m}$ that optimizes*

$$\min_{A,\,P,\,S}\ J(A,P,S) = \min_{A,\,P,\,S}\ \tfrac{1}{2}\,||(AP+S) - Y||_F^2 + \alpha||S||_1$$

$$\textit{subject to } A,\ P \geq 0, \quad Y \geq S \geq 0$$

where $\alpha \geq 0$ is the regularizaton parameter for the sparsity in S.

There are significant differences between this problem and the relaxation (7.4). As stated previously, we do not constrain the cardinality of S. Rather, we regularize the sparsity of S

and bound the magnitude of elements to be less than or equal to Y and greater than or equal to 0. Secondly, in our method the middle rank p is chosen prior to implementation while in (7.4), the rank of L is only bounded above by an r chosen a priori. Thirdly, we do not have a low-rank representation L. We have low-rank factors A and P. If we interpret the rows of P as the basis for the p-dimensional space and A as the assignment, we can determine features in AP by the rows of P and their significance in Y by A. Note, although we can let $L = AP$, it is not necessarily true there exists an L such that we can find an exact factorization AP. Therefore, L and AP are not necessarily the same.

For Problem 7.1, we must find the minimizing AP and we need to find the minimizing S. To simplify this problem, we develop two subproblems to be optimized. First,

$$\min_{A,\,P} J_1(A,P) = \min_{A,\,P} \ \frac{1}{2}\,||(AP+S)-Y||_F^2,$$

$$\text{subject to} \ \ A,\, P \geq 0$$

where S is a constant, and

$$\min_{S} \ J_2(S) = \min_{S} \ \frac{1}{2}\,||(AP+S)-Y||_F^2 + \alpha||S||_1$$

$$\text{subject to} \ \ Y \geq S \geq 0$$

where A and P are constant and $\alpha \geq 0$. We can solve subproblem 1 using the Lee-Seung NMF

Algorithm 7.1 PCP-NMF

$$\begin{array}{lll} (A,P) & \longleftarrow & NMF(Y-S) \\ S & \longleftarrow & \max(0, \min(Y,\, Y-AP-\alpha E)) \end{array}$$

algorithm [65]. That is, given the element-wise gradient descent methods

$$A \leftarrow A - \eta_A(\tfrac{\partial J_1}{\partial A}),$$
$$P \leftarrow P - \eta_P(\tfrac{\partial J_1}{\partial P}),$$

the scaling step-sizes

$$\eta_A = A./(APP'),$$
$$\eta_P = P./(A'AP),$$

and the gradient

$$\nabla J_1 = \left(\frac{\partial J_1}{\partial A}, \frac{\partial J_1}{\partial P}\right) = \left(APP' + (S-Y)P',\ A'AP + A'(S-Y)\right),$$

by substitution, we have

$$A \leftarrow A. * (\hat{Y}P')./(APP')$$

$$P \leftarrow P. * (A'\hat{Y})./(A'AP)$$

where $\hat{Y} = Y - S$. Note, the upper bound on S preserves nonnegativity in $NMF(\hat{Y})$. This is simply the Frobenius NMF updates with $\hat{Y}$ instead of Y. We have that both A and P are descent updates, see chapter 2 and [65].

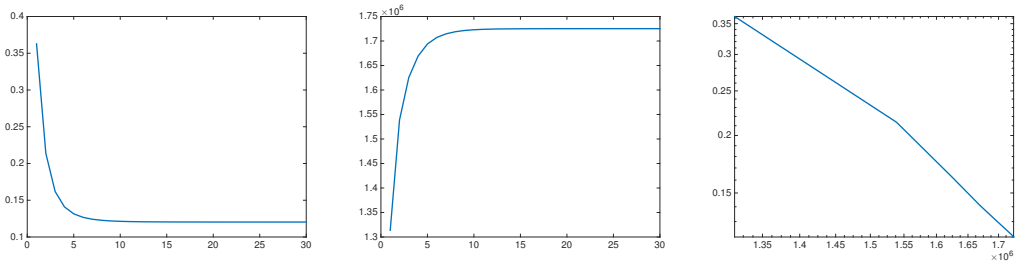
Once $(AP)^+$ is obtained, we can solve the second subproblem. Necessary optimality is given by

$$\frac{\partial J_2}{\partial S} = ((AP)^+ + S - Y) + \alpha E = 0,$$

where $E = \text{ones}(n, m)$. Thus,

$$S^+ = Y - (AP)^+ - \alpha E.$$

We are also constrained to the fact that $Y \geq S \geq 0$ in order to maintain nonnegativity when



(a) fidelity (b) cardinality (c) fidelity v. cardinality

Figure 7.1: Stopping Analysis

alternating to subproblem 1. Hence,

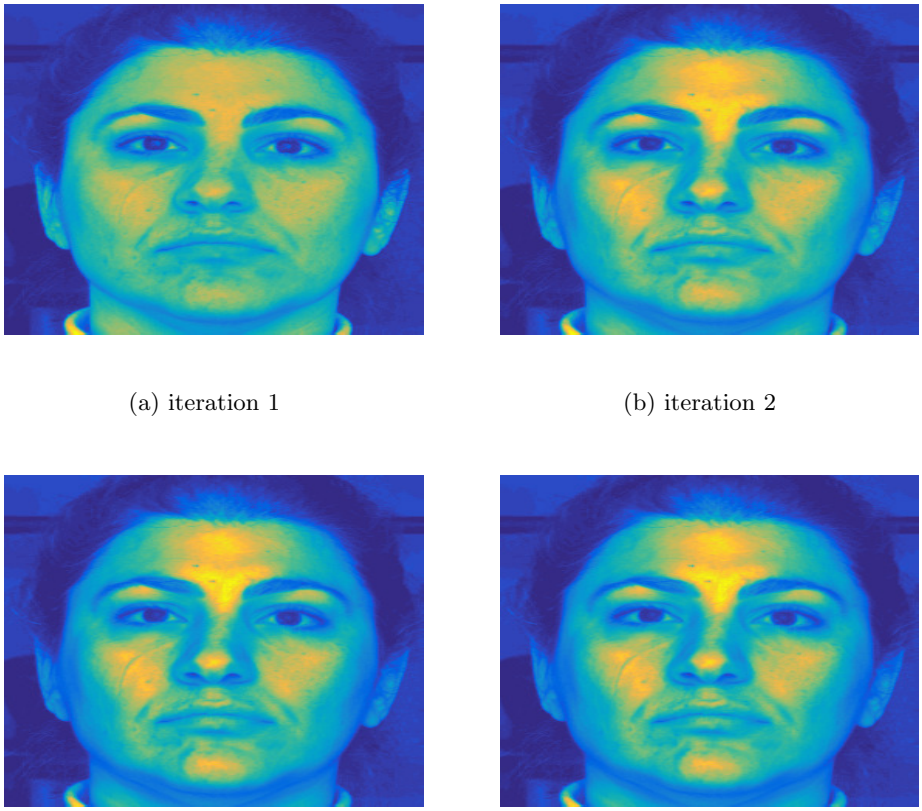
$$S^+ = \max(0, \min(Y, Y - AP - \alpha E)).$$

Remark 1. As we stated earlier, we do not have the truth for proper middle rank of AP nor do we have the truth for S, i.e. how sparse and where nonzeros are located. What we will achieve with the PCP-NMF is a good representation with a low-rank AP and a sparse enough S. □

Remark 2. The alternation in Algorithm 7.1 requires a discussion of stopping criterion. We examine fidelity and sparsity in S as iterations increase. We do so for Figure 7.3a. Consider Figures 7.1a and 7.1b.

Clearly, as iterations increase, fidelity, defined as $\frac{||(AP+S)-Y||_F}{||Y||_F}$, in Figure 7.1a decreases and the number of nonzeros in S (Figure 7.1b) increases. In Figure 7.1c, we have plotted fidelity versus cardinality. At the turning point, we are approximately iteration 2. This is where the decrease in fidelity risks the loss of sparsity. We show in Figures 7.2b-7.2d the resulting low-dimensional representation from PCP-NMF iterations 1 through 4. Observe, light extremities removed in Figure 7.2a (result after 1 iteration) are revived. Therefore, for this example, the number of alternating iterations should be one. We exercise this determination for all examples in this chapter. □

Remark 3. The goodness of fit for the NMF, number of iterations necessary, and computational



(a) iteration 1

(b) iteration 2

(c) iteration 3

(d) iteration 4

Figure 7.2: PCP-NMF Representation Face 1

time are all dependent on the complexity p chosen. While we explored this topic in chapter 4, the choice of p for this chapter requires a different approach. The representation we care about is $AP + S$, not AP. For the applications in this chapter, the intent of the PCP decomposition is to capture the varying components of the data in S and the stabilized components in AP. To this end, we choose $p = 1$. Each column of AP will then contain the same information, and each column of S will be forced to contain the variation in order to better represent Y. One can certainly choose another p, as long as these qualities are not lost. □

7.3 Numerical Experiments

There are many applications in which the PCP decomposition is beneficial, e.g. video surveillance, face recognition, latent semantic indexing, and ranking [14,50,80]. We will look specifically at video processing and facial recognition.

7.3.1 Shadow/Light Corruption

Consider the Figures of faces 7.3a-7.3j[1]. The Database provides 65 images of each face in the same position with different lighting, and each image is originally 480 pixels by 640 pixels in the grayscale [25]. We take the first original image and resize it to tighten the frame and concentrate only on the face. The sizes of the images in the Figures 7.3a-7.3j are 301×271, 300×291, 301×271 and 271×251, respectively. We do this for all 65 images per face. We then stretch each image and lay them column by column to create a Face Matrix with 65 columns. These are the data we use for the example below. We will apply PCP-NMF in order to remove shadow and/or light interference.

We restate here that the accuracy of $L+S$ to Y is not important. We care that S is as sparse as possible while only maintaining the shadow/light corruption and L contains the underlying image. However, we provide the information for the reader in Table 7.1. In addition, we give

[1]These images from Yale Face Database B at http://vision.ucsd.edu/ leekc/ExtYaleDatabase/ExtYaleB.html

Table 7.1: PCP-NMF for Shadow/Light Removal

Image	α	% sparse	$\frac{\|Y-(L+S)\|_F}{\|Y\|_F}$
Face 1	10	75%	0.3628
Face 2	5	62%	0.3076
Face 3	15	76%	0.3371
Face 4	15	74%	0.3297

the percent of sparse-ness of the resulting S matrix given the value of α. We also the Figures 7.3b-7.3l for our analysis.

It it clear by the table, PCP-NMF finds a representation $AP + S$ with a largely sparse S. Furthermore, we can visually verify that shadow corruptions as well as light extremities are removed, while facial features are still captured in the low-rank component.

In conclusion, we have used the 65 corrupted images to extract the shadow/light into S, leaving the underlying facial features in L.

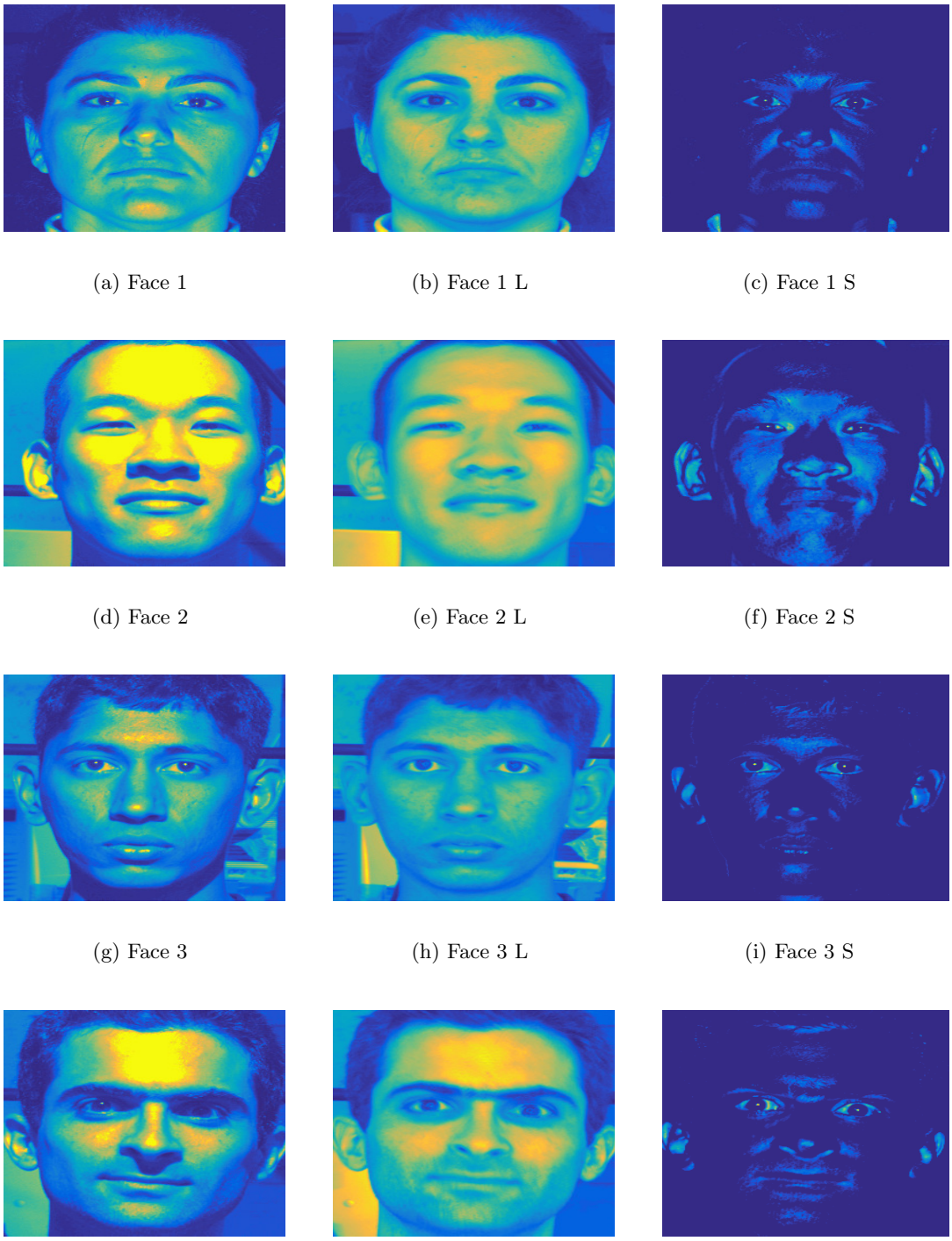
7.3.2 Video Analysis

In this section, we perform experiments in the application of video processing. The goal for our next example is to extract the moving aspects of a video into the sparse matrix and capture the stable characteristics into the low-rank representation. Again, we are not concerned about the exactness of $L + S$. We will concentrate on the visual results.

Consider the Figures 7.4a-7.4e. These images come from a video[2] of a fountain. The video primarily contains the fountain and the running water. However, in the frames displayed, the video captures the movement of people from one end of the frame to the other.

The video has 523 frames with size 128 pixels by 160 pixels. We create a matrix of this video by taking each frame, stretching them into column vectors, and laying them side by side. We then have the data matrix Y size 20480×523. We show the results in Figures 7.4b-7.4f. We found for this example that $\alpha = 3$ allows S to remain sparse while capturing the running water

[2]video from http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html



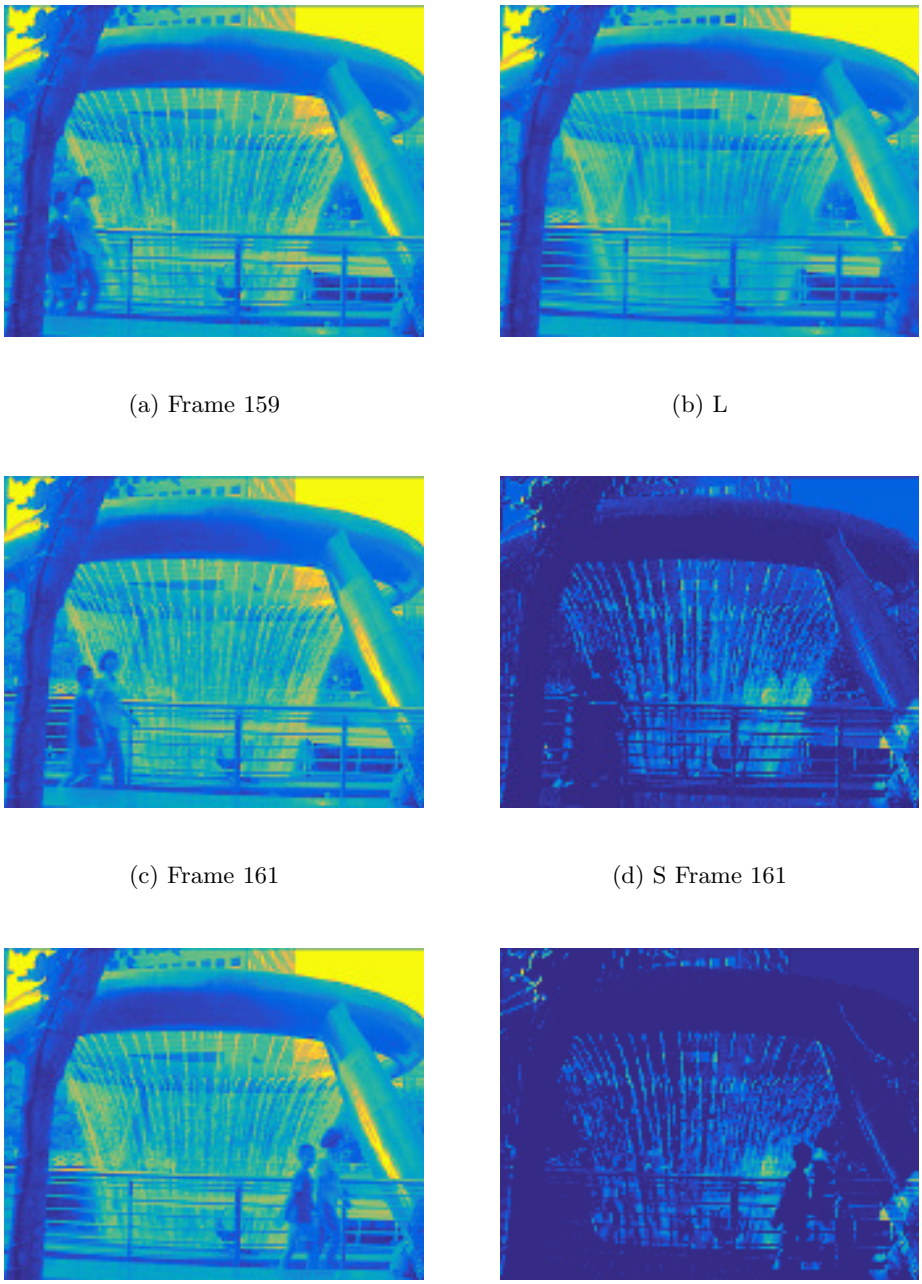
(a) Face 1 (b) Face 1 L (c) Face 1 S

(d) Face 2 (e) Face 2 L (f) Face 2 S

(g) Face 3 (h) Face 3 L (i) Face 3 S

(j) Face 4 (k) Face 4 L (l) Face 4 S

Figure 7.3: Shadow/Light Removal and PCP-NMF



(a) Frame 159

(b) L

(c) Frame 161

(d) S Frame 161

(e) Frame 200

(f) S Frame 200

Figure 7.4: Fountain Video Analysis

Table 7.2: Numerical Results for Videos

α	% sparse	relative norm
3	61%	0.0156
15	86%	0.0427

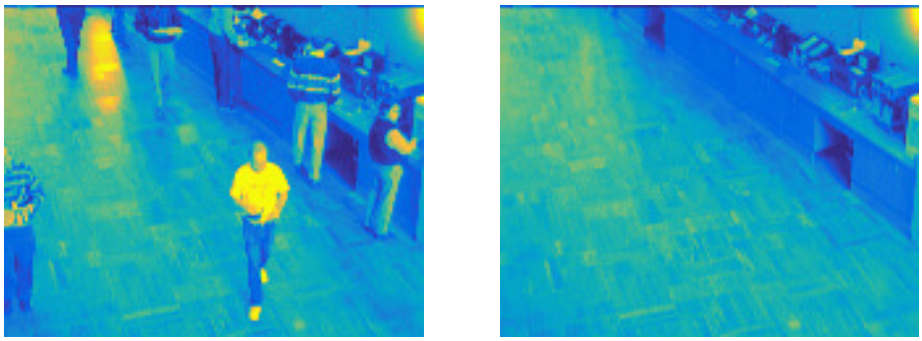
and the moving silhouettes of the people.

To confirm this algorithm indeed is applicable to video processing, we will implement PCP-NMF on another example. Consider a security video from a restuarunt with 3055 frames, each 120 pixels by 160 pixels. We show a few frames where there is clear indication of movement in Figures 7.5a-7.5e. We then show the results from PCP-NMF in Figures 7.5b-7.5f. It is clear, visually, that PCP-NMF successfully separates the background and the movement into L and S, respectively.

7.4 Conclusion

In summary, we have developed a PCP-NMF method, and we have shown it successfully separates low-rank and sparse components in both shadow/light corruption and video processing applications. As we have stated earlier, the nonnegativity of this representation gives a more physical meaning in AP and S for inherently nonnegative data (such as in the images and videos in our examples). For example, data with missing entries is often an issue in image processing or computational biology. The PCP decomposition is shown to be useful in the matrix completion problem [80]. With the PCP-NMF, a solution is guaranteed to have nonnegative elements as an image naturally does.

The PCP-NMF is an "offline" method/batch method [9]. For the examples presented, this is not an issue. However, videos may collect data continuously in time, changing results. To fix this issue, we need to include frames as time continues. Future work involves developing an "online" PCP-NMF.



(a) Frame 1

(b) L

(c) Frame 3

(d) S Frame 3

(e) Frame 8

(f) S Frame 8

Figure 7.5: Restaurant Video Analysis

Chapter 8

Conclusions and Future Work

8.1 Conclusions

In this thesis, we investigated the Nonnegative Matrix Factorization (NMF) as a significant data analysis method. We first provided a basic understanding of the NMF as a representation, and we made insightful observations of the standard NMF algorithms. We then built upon this foundation with conceptual and algorithmic advancements.

We enhanced the NMF as a representation through the development of PDAS-NMF, a primal-dual active set method that secures a better nonnegative matrix factorization. In addition, we modify PDAS-NMF to address the regularized NMF problem. That is, we develop a PDAS-NMF that attains structural properties based on prior information. An important characteristic of the NMF problem is the choice of middle rank, and we have developed a method for a proper choice. This method is dependent on the newly defined NMF-singular values. We compare this method to the statistical AIC and show that we do find a proper p.

We have also examined concepts beyond the NMF as a nonnegative representation. Specifically, we have advanced NMF analysis. We describe the role of atoms in image deconvolution, i.e. the classification of the convolution kernel. We developed a divergence NMF that takes advantage of this information, and we have shown this method is effective for blind deconvolution.

In addition to the NMF and deconvolution, we enhanced the triple NMF by including sparsity on the additional factor and highlighting significant components in the representation. Finally, we developed a PCP-NMF that finds a PCP decomposition. That is, the Principal Component Pursuit decomposition is sparse data components extracted from low-rank data components.

8.2 Future Work

At the end of each chapter, we describe ways to advance concepts and methods. In this section, we present a couple more ways to further advance the details of this thesis.

Parallel computing: Parallel computing introduces more processing power, and thus has become a significant option for data mining applications [73, 74]. Because the NMF is a multifaceted data analysis tool, parallel computing methods for the NMF have been developed [2, 23, 49, 62]. The purpose of these methods is to accelerate the NMF algorithm and to make the NMF more suitable for bigger data. We are interested in developing parallelized adaptations for the methods developed in this thesis in an attempt to increase their viability when dealing with bigger data. For example, the coordinate-descent method presented in chapter 5 is effective for small-scale data. Developing a parallel version will increase speed and capability with large-scale data.

Deep learning: Deep learning is the extraction of significant simple and complex features. Hierarchical models are used in order develop this extraction [8]. This is a multi-layer approach, where each feature is represented by a layer and the bottom most layer will be the simplest features and the top most layer will be the most complex features. Each of these layers are trained separately, pointing out which features are developed at each layer and how features from lower layers are combined to form higher layer features [68]. A hierarchical NMF is introduced in [1,68]. The standard NMF has one hidden layer [31]. Therefore, it will not uncover very many complex features. However, if we stack the NMF, we will have a hierarchical structure with the added benefit of nonnegativity, i.e. the nonnegativity allows for only additive combinations of features. The first layer of features is determined by a simple execution of the NMF. The second

layer of patterns is determined by another execution of the NMF on the matrix of assignments. This is done for all layers. As with parallel computing, in the interest of advancing the concepts and methods of this thesis, we want to develop a hierarchical NMF extension for the methods. For example, a hierarchical structure for the PCP-NMF may provide us with information about more complex features in videos.

Bibliography

[1] J. Ahn, S. Kim, J. Oh, and S. Choi. Multiple nonnegative–matrix factorization of dynamic PET images. In *Proceedings of Asian Conference on Computer Vision*, pages 1009–1013, January 2004.

[2] A. Anisimov, O. Marchenko, E. Nasirov, and S. Palamarchuk. A method for parallel non-negative sparse large matrix factorization. In *Text, Speech and Dialogue*, pages 344–352. Springer, 2014.

[3] D. L. Bailey. *Positron Emission Tomography: Basic Sciences*. Springer Science & Business Media, 2005.

[4] Santa Banta. < http://www.santabanta.com/photos/zebras/15507006.htm >. Accessed: 03-11-2014.

[5] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics & Data Analysis*, 52(1):155–173, 2007.

[6] D. S. Biggs and M. Andrews. Acceleration of iterative image restoration algorithms. *Applied optics*, 36(8):1766–1775, 1997.

[7] D. A. Boas, D. H. Brooks, E. L. Miller, C. A. DiMarzio, M. Kilmer, R. J. Gaudette, and Q. Zhang. Imaging the body with diffuse optical tomography. *Signal Processing Magazine*, 18(6):57–75, 2001.

[8] Y. Boureau, Y. L. Cun, and et. al. Sparse feature learning for deep belief networks. In *Advances in neural information processing systems*, pages 1185–1192, 2008.

[9] T. Bouwmans and E. H. Zahzah. Robust PCA via principal component pursuit: A review for a comparative evaluation in video surveillance. *Computer Vision and Image Understanding*, 122:22–34, 2014.

[10] H. Bozdogan. Model selection and akaike's information criterion (aic): The general theory and its analytical extensions. *Psychometrika*, 52(3):345–370, 1987.

[11] I. Britton. < http://www.freefoto.com/preview/15-01-33/Tree-Black-and-White >, 2004. Accessed: 01-10-2013.

[12] K. P. Burnham and D. R. Anderson. Multimodel inference understanding AIC and BIC in model selection. *Sociological methods & research*, 33(2):261–304, 2004.

[13] S. Butenko and Panos M. Pardalos. *Numerical Methods and Optimization: An Introduction*. CRC Press, 2014.

[14] E. J. Candés, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3), 2011.

[15] E. J. Candès and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *Information Theory, IEEE Transactions on*, 56(5):2053–2080, 2010.

[16] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM journal on scientific computing*, 20(1):33–61, 1998.

[17] C. P. de Campos, P. M. Rancoita, I. Kwee, E. Zucca, M. Zaffalon, and F. Bertoni. Discovering subgroups of patients from DNA copy number data using NMF on compacted matrices. *PloS one*, 8(11), 2013.

[18] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, pages 1–38, 1977.

[19] K. Devarajan. Nonnegative matrix factorization: an analytical and interpretive tool in computational biology. *PLoS computational biology*, 4(7), 2008.

[20] C. Ding, T. Li, and W. Peng. On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Computational Statistics & Data Analysis*, 52(8):3913–3927, 2008.

[21] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix tri-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, (ACM), pages 126–135, 2006.

[22] C. H. Ding, X. He, and H. D. Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. *SDM*, 5:606–610, April 2005.

[23] C. Dong, H. Zhao, and W. Wang. Parallel nonnegative matrix factorization algorithm on the distributed memory platform. *International journal of parallel programming*, 38(2):117–137, 2010.

[24] V. Estivill-Castro. Why so many clustering algorithms: a position paper. *ACM SIGKDD Explorations Newsletter*, 4(1):65–75, 2002.

[25] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. From few to many: illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001.

[26] P. E. Gill, M. A. Saunders, and J. R. Shinnerl. On the stability of cholesky factorization for symmetric quasidefinite systems. *SIAM Journal on Matrix Analysis and Applications*, 17(1):35–46, 1996.

[27] A. M. González and et. al. Segmentation of brain mri structures with deep machine learning. 2012.

[28] E. T. Hale, W. Yin, and Y. Zhang. A fixed-point continuation method for ℓ_1-regularized minimization with applications to compressed sensing. *CAAM TR07-07, Rice University*, 2007.

[29] C. Hansen. The L-curve and its use in the numerical treatment of inverse problems. Technical report, Uuniversity of Dunmark, 1999.

[30] Michael Hintermüller, Kazufumi Ito, and Karl Kunisch. The primal-dual active set strategy as a semismooth newton method. *SIAM Journal on Optimization*, 13(3):865–888, 2002.

[31] G. E. Hinton. Learning multiple layers of representation. *Trends in cognitive sciences*, 11(10):428–434, 2007.

[32] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, (ACM), pages 50–57, 1999.

[33] D. Hooker. < http://en.wikipedia.org/wiki/Lenna#/media/File:Lenna.png >, 1973. Accessed: 01-10-2015.

[34] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *The Journal of Machine Learning Research*, 5:1457–1469, 2004.

[35] C. Hsieh and I. S. Dhillon. Fast coordinate descent methods with variable selection for non-negative matrix factorization. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1064–1072. ACM, 2011.

[36] K. Ito and B. Jin. *Inverse problems: Tikhonov theory and algorithms*, volume 22. World Scientific Publishing Company, 2014.

[37] K. Ito and K. Kunisch. The primal-dual active set method for nonlinear optimal control problems with bilateral constraints. *SIAM Journal on Control and Optimization*, 43(1):357–376, 2004.

[38] K. Ito and K. Kunisch. *Lagrange multiplier approach to variational problems and applications*, volume 15. SIAM, 2008.

[39] J. P. Kaipio and E. Somersalo. Classical regularization methods. *Statistical and Computational Inverse Problems*, pages 7–48, 2005.

[40] A. Krishnamoorthy and D. Menon. Matrix inversion using cholesky decomposition. *arXiv preprint arXiv:1111.4144*, 2011.

[41] D. Kuang, H. Park, and C. H. Ding. Symmetric nonnegative matrix factorization for graph clustering. *SDM*, 12:106–117, 2012.

[42] J. Kuha. AIC and BIC comparisons of assumptions and performance. *Sociological Methods & Research*, 33(2):188–229, 2004.

[43] M. Laasmaa. The Analysis of richardson-lucy deconvolution algorithm with application to microscope images. Master's thesis, Tallinn University of Technology, 2009.

[44] E. Y. Lam and J. W. Goodman. Iterative statistical approach to blind image deconvolution. *JOSA A*, 17(7):1177–1184, 2000.

[45] G. Landi, E. L. Piccolomini, and F. Zama. A total variation-based reconstruction method for dynamic MRI. *Computational and Mathematical Methods in Medicine*, 9(1):69–80, 2008.

[46] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[47] T. Li and C. Ding. The relationships among various nonnegative matrix factorization methods for clustering. In *Sixth International Conference on Data Mining, 2006*, ICDM, pages 362–371. IEEE, December 2006.

[48] Y. Li and S. Osher. Coordinate descent optimization for ℓ_1 minimization with application to compressed sensing; a greedy algorithm. *Inverse Problems and Imaging*, 3(3):487–503, 2009.

[49] C. Liu, H. Yang, J. Fan, L. He, and Y. Wang. Distributed nonnegative matrix factorization for web-scale dyadic data analysis on mapreduce. In *Proceedings of the 19th international conference on World wide web*, pages 681–690. ACM, 2010.

[50] X. Luan, B. Fang, L. Liu, W. Yang, and J. Qian. Extracting sparse error of robust PCA for face recognition in the presence of varying illumination and occlusion. *Pattern Recognition*, 47(2):495–508, 2014.

[51] L. B. Lucy. An iterative technique for the rectification of observed distributions. 79(6):745–754, 1974.

[52] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[53] U. Maulik and S. Bandyopadhyay. Performance evaluation of some clustering algorithms and validity indices. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(12):1650–1654, 2002.

[54] D. W. McRobbie, E. A. Moore, M. J. Graves, and M. R. Prince. *MRI from picture to proton*. Cambridge University Preds, 2003.

[55] C. D. Meyer. *Matrix analysis and applied linear algebra*. Siam, 2000.

[56] M. Mudrova and A. Procházka. Principal component analysis in image processing.

[57] M. F. Ochs and E. J. Fertig. Matrix factorization for transcriptional regulatory network inference. *Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 2012 IEEE Symposium on*, pages 387–396, May 2012.

[58] M. F. Ochs, R. S. Stoyanova, F. Arias-Mendoza, and T. R. Brown. A new method for spectral decomposition using a bilinear Bayesian approach. *Journal of Magnetic Resonance*, 137(1):161–176, 1999.

[59] M.F. Ochs, L. Rink, S. Tarn, C.and Mburu, T. Taguchi, B. Eisenberg, and A.K. Godwin. Detection of treatment-induced changes in signaling pathways in gastrointestinal stromal tumors using transcriptomic data. *Cancer research*, 69(23):9125–9132, 2009.

[60] U.S. National Library of Medicine. < http://www.nlm.nih.gov/research/visible/mri.html >. Accessed: 01-02-2015.

[61] P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.

[62] B. Recht, C. Re, J. Tropp, and V. Bittorf. Factoring nonnegative matrices with linear programs. In *Advances in Neural Information Processing Systems*, pages 1214–1222, 2012.

[63] W. H. Richardson. Bayesian-based iterative method of image restoration. *JOSA*, 62(1):55–59, 1972.

[64] M. Rossi. < http://www.theguardian.com/music/musicblog/2012/jul/26/readers-recommend-songs-running >, 2012. Accessed: 08-03-2014.

[65] D. Seung and L. Lee. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13:556–562, 2001.

[66] Y. Sharon, J. Wright, and Y. Ma. Computation and relaxation of conditions for equivalence between ℓ_1 and ℓ_0 minimization.

[67] J. R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain, 1994.

[68] H. A. Song and S. Y. Lee. Hierarchical representation using NMF. *Neural Information Processing*, pages 466–473, January 2013.

[69] X. Wang, J. Tian, X. Li, J. Dai, and L. Ai. Detecting brain activations by constrained non-negative matrix factorization from task-related BOLD fMRI. *Medical Imaging 2004*, pages 675–682, April 2004.

[70] J. Wilson, N. Patwari, and F. G. Vasquez. Regularization methods for radio tomographic imaging. Citeseer.

[71] S. Wold, K. Esbensen, and P. Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1):37–52, 1987.

[72] T. Wu and K. Lange. Coordinate descent algorithms for lasso penalized regression. *The Annals of Applied Statistics*, pages 224–244, 2008.

[73] X. Wu, X. Zhu, G. Wu, and W. Ding. Data mining with big data. *Knowledge and Data Engineering, IEEE Transactions on*, 26(1):97–107, 2014.

[74] H. Xiao. Towards parallel and distributed computing in large-scale data mining: A survey. 2010.

[75] Y. Yang. Can the strengths of AIC and BIC be shared? A conflict between model identification and regression estimation. *Biometrik*, 92(4):937–950, 2005.

[76] Z. Yang, H. Zhang, Z. Yuan, and E. Oja. Kullback-Leibler divergence for nonnegative matrix factorization. *Artificial Neural Networks and Machine Learning?ICANN 2011*, pages 250–257, 2011.

[77] W. Yin, S. Osher, D. Goldfarb, and J. Darbon. Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing. *SIAM Journal on Imaging Sciences*, 1(1):143–168, 2008.

[78] B. Zhang and S. Horvath. A general framework for weighted gene co–expression network analysis. *Statistical Applications in Genetics and Molecular Biology*, 4(1):1–43, 2005.

[79] X. Zhao, F. Wang, T. Huang, M. K. Ng, and R. J. Plemmons. Deblurring and sparse unmixing for hyperspectral images. *IEEE T. Geoscience and Remote Sensing*, 51(7-1):4045–4058, 2013.

[80] T. Zhou and D. Tao. Godec: Randomized low-rank & sparse matrix decomposition in noisy case. In *Proceedings of the 28th International Conference on Machine Learning*, (ICML-11), pages 33–40, 2011.