Fall 2013

A fast algorithm for general matrix factorization

Xuan Zhou
University of Iowa

A FAST ALGORITHM FOR GENERAL MATRIX FACTORIZATION

by

Xuan Zhou

A thesis submitted in partial fulfillment of the
requirements for the Master of Science degree
in Electrical and Computer Engineering
in the Graduate College of
The University of Iowa

December 2013

Thesis Supervisor: Assistant Professor Mathews Jacob

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

MASTER'S THESIS

This is to certify that the Master's thesis of

Xuan Zhou

has been approved by the Examining Committee for the thesis requirement for the Master of Science degree in Electrical and Computer Engineering at the December 2013 graduation.

Thesis Committee: _____
Mathews Jacob, Thesis Supervisor

_____
Er-wei Bai

_____
Weiyu Xu

ACKNOWLEDGEMENTS

First of all I would like to express my sincere gratitude to my supervisor Dr. Mathews Jacob for his contribution and support throughout my Master's study. He has always given me constructive and effective guidance in my research and inspired me with brilliant ideas. He is always there to listen to my ideas and to share his advice. I greatly appreciate the efforts he put on me and respect his profound knowledge and great passion on research and his patience and encouragement on every students.

I also appreciate Dr. Er-wei Bai and Dr. Weiyu Xu being my Master defense committee with their advice on the thesis. Special thanks goes to all my labmates in the Computational Biomedical Imaging Group (CBIG): Yue Hu, Sajan Goud, Chen Cui, Merry Mani, Greg Ongie, Sampada Bhave, Ipshita Bhattacharya, Arvind Balachandrasekaran, Sunrita Poddar, Yasir Baqqal. Thanks for all the great ideas and advice from them. I cherish the memory of academic discussion and wonderful lab life with them.

Last but not least, I would like to thank my husband Zhongyi Yuan, my parents and parents in law. Without their love, support and encouragement, I cannot get through this period.

Again, millions of thanks to all the people who helped me in the past years.

ABSTRACT

Matrix factorization algorithms are emerging as popular tools in many applications, especially dictionary learning method for recovering biomedical image data from noisy and ill-conditioned measurements. We introduce a novel dictionary learning algorithm based on augmented Lagrangian (AL) approach to learn dictionaries from exemplar data and it can be extended to general matrix factorization problems due to different constraints. Specifically, we use the alternating minimization strategy to decouple the dictionary learning scheme into three main subproblems, which can be solved efficiently. The proposed algorithm can accommodate arbitrary priors on the dictionary, which enables us to inject prior information into the learning process. We validate the algorithm using simulated data and demonstrate its utility in the context of denoising. Comparisons with existing methods show a considerable speedup over other methods. More importantly, we observe that the proposed algorithm is able to recover the dictionaries correctly, even at high sparsity levels and is relatively insensitive to initialization.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF ALGORITHMS

Algorithm

CHAPTER 1
INTRODUCTION

The factorization of a given matrix $\mathbf{Y}$ into two matrices $\mathbf{D}$ and $\mathbf{X}$ is a classical problem with wide-ranging applications. Since this factorization is not unique, several constraints or penalties were introduced on the factors to make the problem well-posed and have desirable solutions. For example, the widely used principal component analysis (PCA) [7, 10] factorization can be formulated as a matrix factorization scheme with low-rank constraints. Similarly, non-negativity constraints can be used to obtain non-negative matrix factorization [16]. In the recent years, several researchers have considered sparse matrix factorizations in the context of dictionary learning. Most of the classical dictionary learning algorithms focus on sparsity promoting ℓ_1 norm on the coefficient matrix $\mathbf{X}$, while the columns of the dictionary $\mathbf{D}$ is constrained to have unit norm. Even though these problems can be generalized to a uniform form, they are solved using different algorithms and there is no general frame currently. Therefore, we aim to propose a fast algorithm for general matrix factorization. In this thesis, we focus on the problem of dictionary learning, but the proposed algorithm is able to extend to general matrix factorization.

The recovery of image data from noisy and ill-conditioned measurements is a common problem in many biomedical inverse problems. Algorithms that rely on the sparsity of the signal in pre-determined dictionaries have shown to be highly effective in practical applications. While classical schemes rely on analytical dictionaries (eg. discrete cosine transform, wavelets), Several dictionary learning algorithms (e.g. K-

SVD, MOD) [1, 5, 22] in which the dictionaries and their coefficients are learned from the under-sampled data directly, have been introduced in the recent years and this strategy is far more effective in ill-posed inverse problems than using pre-determined atoms.

However, one of the main challenges associated with these algorithms is their high computational complexity, especially when applied to large scale imaging problems. Another challenge is the non-convexity of the criterion, which makes the algorithms vulnerable to local minima. Current methods rely on initializing the algorithm with good initial guesses (e.g. discrete cosine transform) to obtain good solutions. Therefore, it is important to find an effective algorithms that are less sensitive to initialization to obtain good dictionaries. In addition, most of those algorithms such as K-SVD and MOD have the scale ambiguity problem. To address this problem, we usually assume atoms with unit column norms. However, some researchers have argued for the use of other convex constraints (e.g. bounded Frobenius norm, sparsity of the basis functions) to inject prior information into the dictionary learning algorithms. The majorize-minimize framework was introduced in [22] to introduce more flexible priors. This scheme alternates between the minimization of two different majorizations of the criterion and a projection to the convex dictionary constraint set. We observe this method to be computationally expensive, mainly because of the steepest descend algorithms used to minimize the majorizations and the discrepancies between the different steps.

To address the challenges above in dictionary learning, we aim to develop a

novel matrix factorization algorithm in this thesis and extend it to solve general matrix factorization problems. We formulate the dictionary learning problem as a constrained optimization problem, where the linear combination of the data-consistency and a sparsity penalty on the coefficients is minimized, subject to a dictionary constraint. Based on the recent success of non-convex sparsity penalties [20, 2, 9], we consider $\ell_p : p < 1$ norm on matrix entries to get the sparse representation. We introduce an auxiliary variable that is constrained to be equal to the coefficient matrix; this approach enables us to transfer the sparsity penalty onto the auxiliary variable from the coefficient matrix. And then, the augmented Lagrangian (AL) framework is used to impose the equality constraint [8]. We also introduce another auxiliary variable that is constrained to be equal to the dictionary and use the AL framework to enforce the equality constraint. Next, we use a projection step as in [22] to enforce constraints on the dictionary. We expect this approach to yield smoother and faster convergence. The proposed iterative algorithm proceeds by alternating between three simple steps:

1. update the auxiliary variable on coefficient matrix, which involves an analytical shrinkage,
2. update the coefficient matrix as a quadratic optimization problem, and
3. update the dictionary by solving a constrained subproblem.

We solve for the first quadratic sub-problem of updating the coefficients analytically. The constrained subproblem of updating the dictionary is solved using the AL scheme to enforce another auxiliary variable equality problem and using the projec-

tion method to enforce dictionary constraints. Since all of the above steps are simple, we obtain a computationally efficient algorithm.

We determine the utility of the proposed algorithm using simulated data as well as in the context of MR image denoising. The results in the simulations show that the proposed algorithm is considerably faster than the classical methods, while yielding dictionaries that are closer to the original at almost all sparsity levels. We also observe that the improved dictionaries translate to improve denoising performance. While we only demonstrate the utility of the algorithm in denoising, we expect this scheme to be useful in other biomedical applications including recovery from under sampled data and tomography.

The rest of thesis is organized as follows: Chapter 2 presents the background of some popular matrix factorization problems and reviews the principle of the augmented Lagrangian method which we have used in our proposed algorithm. The proposed algorithm is introduced in Chapter 3. Chapter 4 is devoted to experimental results which demonstrating that our algorithm has lots of good properties.

CHAPTER 2
BACKGROUND

2.1 Introduction of Popular Matrix Factorization Problems

Matrix factorization arises in a wide range of application domains and many problems can be posed as matrix factorization problems. In this thesis, we discuss some famous matrix factorization problems based on different constraints. The widely used Principal Component Analysis (PCA) [7] can be formulated as a matrix factorization scheme with low-rankedness constraints. Low-rankedness is useful for learning a lower dimensionality representation. Besides, sparse PCA [10] can be used to find local features that constitute the dataset, for example parts of faces, for a dataset of facial images. Independent Component Analysis (ICA) [11] has statistic independent constraints on subcomponents and is feasible for processing multidimensional data. Similarly, nonnegativity, which is a natural constraint when modeling data with physical constraints, such as chemical concentrations in solutions, pixel intensities in images and radiation dosages for cancer treatment, can be used to obtain non-negative matrix factorization (NMF) [16]. NMF offers a good model for additive data such as text or images. Since sparsity is useful for modeling the conciseness of the representation or the latent features, several researchers have considered sparse matrix factorization in the context of dictionary learning which is very good for image reconstruction [1].

2.1.1 Principal Component Analysis

Principal Component Analysis (PCA) is an essential tool for data analysis and unsupervised dimensionality reduction. Using PCA, a multivariate dataset can be represented by a sequence of orthogonal components that convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables. By capturing directions of maximum variance in the data, the principal components offer a way to compress the data with minimum information loss. PCA has been widely studied and used in pattern recognition and signal processing, such as data compression, feature extraction, noise filtering, signal restoration and classification [13].

Let the data $\mathbf{Y}$ be a $M \times N$ matrix, where M and N are the number of observations and the number of variables, respectively. Without loss of generality, assume the variables contained in the columns of $\mathbf{Y}$ are centered. Let the SVD of $\mathbf{Y}$ be

$$\mathbf{Y} = \mathbf{T}\boldsymbol{\Sigma}\mathbf{V}^T$$

then $\mathbf{U} = \mathbf{T}\boldsymbol{\Sigma}$ are the principal components (PCs), and the columns of $\mathbf{V}$ are the corresponding loadings of the principal components. Usually a small number of PCs are chosen to represent the data, thus a great dimensionality reduction is achieved. The synthesis view of PCA can be written as

$$\min_{\mathbf{U},\mathbf{V}} \ \frac{1}{2}\left\|\mathbf{Y} - \mathbf{U}\mathbf{V}^T\right\|_F^2 \quad \text{subject to } \mathbf{V}^T\mathbf{V} = \mathbf{I} \tag{2.1}$$

where columns of $\mathbf{U}$ and $\mathbf{V}$ represent PCs and PC loading vectors respectively. However, each principal component in 2.1 is a linear combination of all the original vari-

ables, thus it is often difficult to interpret the PCs, especially when M is large as frequently encountered in practical applications. To overcome this problem, a new approach for estimating PCs with sparse loadings, which is called sparse principal component analysis (SPCA), has been proposed.

Note that there are many different formulations for the SPCA problem. The one implemented here is based on the SPCA method in [7] for identifying sparse components as a regularized low-rank matrix approximation

$$\min_{\mathbf{U},\mathbf{V}} \ \frac{1}{2}\left\|\mathbf{Y}-\mathbf{U}\mathbf{V}^T\right\|_F^2 + \mathbf{P}_\lambda(\mathbf{V}) \qquad \text{subject to } \ \|\mathbf{u}_i\|_{\ell_2} = 1 \tag{2.2}$$

In [7], the regularization function $\mathbf{P}_\lambda(\mathbf{V})$ has been considered as the soft thresholding (or ℓ_1 or lasso), the hard thresholding, and the smoothly clipped absolute deviation (SCAD). Traditionally, an iterative algorithm is considered to minimize (2.2) with respect to $\mathbf{U}$ and $\mathbf{V}$. Firstly, consider the problem of optimizing over $\mathbf{U}$ for a fixed $\mathbf{V}$. And then solve the problem over $\mathbf{V}$ for a fixed $\mathbf{U}$. This problem 2.2 is convex with respect to $\mathbf{U}$ for fixed $\mathbf{V}$ and vice versa. It is however not jointly convex in the pair ($\mathbf{U}$, $\mathbf{V}$). Therefore, the convergence rate of this iterative alternating algorithm is slow.

2.1.2 Independent Component Analysis

Since PCA relies completely on second-order statistics of the data, it is important to develop a new computational tool for analyzing multidimensional data. So Independent component analysis (ICA) is proposed. It can separate a multivariate signal into additive subcomponents that are maximally independent. ICA has

been widely used on revealing interesting information on brain activity from electrical recordings of an electroencephalogram (EEG) and feature extraction. In signal processing, ICA is used to find suitable representations for image, audio or other kind of data for tasks like compression and denoising.

One of the ICA models where noise is taken into consideration takes the form [12]

$$\mathbf{Y} = \mathbf{D}\mathbf{X} + \mathbf{E}$$

where $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_N]$ is the given observation signals, $\mathbf{D} \in R^{M \times K}$ denotes the regressors, $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_N]$ denotes independent components, and $\mathbf{E} = [\varepsilon_1, ..., \varepsilon_N]$ represents noise. The problem of ICA is to estimate unknown constant $\mathbf{D}$ and independent $\mathbf{X}$ from the noisy data $\mathbf{Y}$.

One way to approach this noisy ICA problem is maximizing joint likelihood [12, 18]. The densities $P_i(\cdot)$ of the $\mathbf{x}_i$ are usually known or already approximated. Assume that the noise is Gaussian distributed with known covariance matrix Σ and is independent of each other for different variables. Then, the joint log-likelihood function can be represented as

$$L(\mathbf{D}, \mathbf{X}) = -\sum_{j=1}^{N} \left[\frac{1}{2} \left\| \mathbf{y}_j - \mathbf{D}\mathbf{x}_j \right\|_{\mathbf{\Sigma}^{-1}}^2 + \sum_{i=1}^{K} \left(-\log P(x_{i,j}) \right) \right] + C$$

where $\|\mathbf{e}\|_{\mathbf{\Sigma}^{-1}}^2$ is defined as $\mathbf{e}^T \mathbf{\Sigma}^{-1} \mathbf{e}$, and C is an irrelevant constant. Since here we assume Σ is known, the problem of maximizing $L(\mathbf{D}, \mathbf{X})$ can be rewritten as

$$\min_{\mathbf{D},\mathbf{X}} \ \frac{1}{2} \left\| \widetilde{\mathbf{Y}} - \mathbf{D}\widetilde{\mathbf{X}} \right\|_F^2 + \sum_{j=1}^{N} \sum_{i=1}^{K} \left(-\log P(x_{i,j}) \right) \tag{2.3}$$

where $\widetilde{\mathbf{Y}} = \mathbf{Y}\boldsymbol{\Sigma}^{-1/2}$ and $\widetilde{\mathbf{X}} = \mathbf{X}\boldsymbol{\Sigma}^{-1/2}$. The expectation-maximization (EM) algorithm provides a general iterative approach for computing maximum likelihood estimates. A problem with the EM algorithm is, however, that the computational complexity grows exponentially with the dimension of the data.

2.1.3 Non-negative Matrix Factorization

Non-negative matrix factorization (NMF) [16] is useful for finding representations of non-negative data. The non-negativity constraints make the representation purely additive (allowing no subtractions), in contrast to many other linear representations such as PCA and ICA. NMF has been applied to many areas such as images analysis [16, 21], document clustering [15], data analysis [14], etc.

Given a matrix $\mathbf{Y}$ of size $M \times N$, the model of NMF can be described as

$$\min_{\mathbf{D},\mathbf{X}} \frac{1}{2} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \quad \text{such that } \mathbf{D} \geq 0, \mathbf{X} \geq 0 \tag{2.4}$$

To address this problem, [16] devised a multiplicative algorithm that is simple to implement and also shows good performance. However, it lacks optimization properties [6]. The gradient algorithms [19] have been proposed for this optimization. Gradient descent method is simple to implement, but convergence can be slow.

2.1.4 Dictionary Learning

Dictionary learning is the process of finding a dictionary $\mathbf{D}$ in which a given set of training samples $\mathbf{Y}$ has sparse approximation $\mathbf{X}$. It has been widely used in image denoising, compression, regularization in inverse problems, feature extraction [1, 4]. The task of computing a representation for samples can be formally described

by

$$\min_{\mathbf{D},\mathbf{X}} \quad \frac{1}{2}\|\mathbf{Y}-\mathbf{D}\mathbf{X}\|_F^2 \quad \text{such that} \quad \|\mathbf{x}_i\|_0 \leq L \tag{2.5}$$

where $\mathbf{x}_i$ represents the ith column of $\mathbf{X}$ and L is the sparsity.

Note that the data consistency term in (2.5) is dependent on the product $\mathbf{DX}$, while the penalty term is only dependent on $\mathbf{X}$. If no constraint is applied on the dictionary $\mathbf{D}$, the optimization (2.5) will result in arbitrarily small coefficients. To avoid this scale ambiguity problem, it is a common practice to constrain the dictionary to unit column norm. Besides, the ℓ_0-norm minimization problem is NP hard. It is replaced by convex surrogate, the ℓ_1 norm. The ℓ_1-norm minimization problem is more popular because its envelope is convex and therefore easier to theoretically analyze [17]. Therefore, the problem (2.5) becomes

$$\arg\min_{\mathbf{D},\mathbf{X}} \quad \frac{1}{2}\|\mathbf{Y}-\mathbf{D}\mathbf{X}\|_F^2 + \lambda\|\mathbf{X}\|_{\ell_1} \quad \text{such that} \quad \|\mathbf{d}_i\|_{\ell_2} = 1 \tag{2.6}$$

where $\mathbf{d}_i$ represents the ith column of $\mathbf{D}$.

To address this problem, most of the current algorithms alternate between the estimation of the sparse coefficients $\mathbf{X}$, assuming the dictionary to be known, and the dictionary atoms $\mathbf{D}$, assuming the coefficients to be known.

2.2 Review of the AL Approach

In this section, we will review the basic idea of the augmented Lagrangian (AL) algorithm. We first consider the general constrained problem

$$\min_{\mathbf{x}} \; f(\mathbf{x}) = f(x_1, x_2, ..., x_n) \tag{2.7}$$

$$\text{subject to } c_i(\mathbf{x}) = 0, \qquad i = 1, ..., m.$$

Then, we can have the Lagrange function as

$$\begin{aligned} L(\mathbf{x}, \lambda) &= f(\mathbf{x}) + \lambda^T \mathbf{c}(\mathbf{x}) && (2.8) \\ &= f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i c_i(\mathbf{x}), && (2.9) \end{aligned}$$

where the vector $\lambda = (\lambda_1, ..., \lambda_m)^T$ is the Lagrange multiplier estimate.

If $\mathbf{x}^*$ is the solution to the problem, then $\mathbf{x}^*$ is a stationary point of $L(\mathbf{x}, \lambda^*)$ and we have

$$\nabla_{\mathbf{x}} L(\mathbf{x}^*, \lambda^*) = \nabla f(\mathbf{x}^*) + \nabla^T \mathbf{c}(\mathbf{x}^*) \lambda^* = 0; \tag{2.10}$$

$$\nabla_{\lambda} L(\mathbf{x}^*, \lambda^*) = \mathbf{c}(\mathbf{x}^*) = 0.$$

To satisfy these conditions, an alternate iterating method is applied with respect to $\mathbf{x}$ and λ. The updating rule for λ is

$$\lambda_{k+1} = \lambda_k - h\mathbf{c}(\mathbf{x}_k)$$

where h is the step size. This steepest descent scheme is easy to implement, but it converge slow and it is hard to choose appropriate h to guarantee convergence. To address those problems, the augmented Lagrangian method [8] is proposed and its function can be written as

$$\begin{aligned} L(\mathbf{x}, \lambda, \mu) &= f(\mathbf{x}) + \mathbf{c}^T(\mathbf{x}) \lambda + \frac{1}{2} \mu \|\mathbf{c}(\mathbf{x})\|^2 && (2.11) \\ &= f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i c_i(\mathbf{x}) + \frac{1}{2} \mu \sum_{i=1}^{m} c_i(\mathbf{x})^2, && (2.12) \end{aligned}$$

where μ is the penalty parameter. According the KKT conditions,

$$\begin{aligned} \nabla_{\mathbf{x}} L\left(\mathbf{x}_k, \lambda_k, \mu_k\right) &= \nabla f\left(\mathbf{x}_k\right) + \nabla^T \mathbf{c}\left(\mathbf{x}_k\right) \lambda_k + \mu_k \nabla^T \mathbf{c}\left(\mathbf{x}_k\right) \mathbf{c}\left(\mathbf{x}_k\right) \quad (2.13) \\ &= \nabla f\left(\mathbf{x}_k\right) + \nabla^T \mathbf{c}\left(\mathbf{x}_k\right) \left(\lambda_k + \mu_k \mathbf{c}\left(\mathbf{x}_k\right)\right) \quad (2.14) \end{aligned}$$

The sequence $\mathbf{x}$ may converge to the minimum point $\mathbf{x}^*$ only if $\lambda_k \to \lambda^*$. Compare (2.14) and (2.10), we can deduce

$$\lambda^* \approx \lambda_k + \mu_k \mathbf{c}\left(\mathbf{x}_k\right) \quad (2.15)$$

By rearranging this expression, we get

$$\mathbf{c}\left(\mathbf{x}_k\right) \approx -\frac{1}{\mu_k}(\lambda^* - \lambda_k)$$

so we conclude that if λ_k is close to λ^*, we can obtain a good estimate of $\mathbf{x}^*$ even when μ_k is not particularly large. Also, (2.15) suggests a formula similar to (2.2) in classic Lagrange method for updating the estimate λ_k by

$$\lambda_{k+1} = \lambda_k + \mu_k \mathbf{c}\left(\mathbf{x}_k\right) \quad (2.16)$$

and we initialize the penalty parameter $\mu_k > 0$ as a small value and gradually increase it.

CHAPTER 3
METHODOLOGY

In this chapter, we will propose a fast algorithm for general matrix factorization problems. First, we generalize an uniform formula for matrix factorization problems according to different constraints. To minimize this objective function, a novel scheme based augmented Lagrangian approach is developed. Then, we use the alternating minimization strategy to decouple the scheme into three main subproblems, which can be solved efficiently. As an example, we focus on applying the proposed algorithm on dictionary learning problem with different dictionary constraints which can be extended to many other matrix factorization problems.

3.1 The Objective Function

The approximation of matrix factorization problems can be generalized as

$$\arg\min_{\mathbf{D},\mathbf{X}} \quad \frac{1}{2}\|\mathbf{Y}-\mathbf{D}\mathbf{X}\|_F^2 + \lambda\varphi(\mathbf{X}) \quad \text{such that} \quad \mathbf{D}\in\Delta \tag{3.1}$$

where $\varphi(\mathbf{X})$ and Δ are constraints on $\mathbf{X}$ and $\mathbf{D}$. Depending on the constraints utilized, the resulting factors show very different representational properties and the problem can be used in different applications.

- PCA

From (2.1), we can get the objective function of PCA problem

$$\min_{\mathbf{D},\mathbf{X}} \ \frac{1}{2}\|\mathbf{Y}-\mathbf{D}\mathbf{X}\|_F^2 \ \text{such that} \ \mathbf{X}^T\mathbf{X}=\mathbf{I} \tag{3.2}$$

Here, PCA enforce a constraints on $\mathbf{X}$. For SPCA case, if we consider representative ℓ_1-norm penalty ($\|\mathbf{V}\|_{\ell_1}$) as $\mathbf{P}_\lambda(\mathbf{V})$, then formula (2.2) can be rewritten as

$$\min_{\mathbf{D},\mathbf{X}} \ \frac{1}{2}\left\|\widetilde{\mathbf{Y}}-\mathbf{D}\mathbf{X}\right\|_F^2 + \lambda\|\mathbf{D}\|_{\ell_1} \quad \text{such that } \left\|\mathbf{x}^i\right\|_{\ell_2} = 1 \tag{3.3}$$

where $\mathbf{x}^i$ represents the ith row of $\mathbf{X}$; $\widetilde{\mathbf{Y}} = \mathbf{Y}^T$; $\mathbf{D} = \mathbf{V}$; $\mathbf{X} = \mathbf{U}^T$. Problem (3.3) is similar to dictionary learning problem which we will discuss later. SPCA problem enforces sparsity penalty on dictionary $\mathbf{D}$ and row-norm constraints (similar to column-norm constraints in dictionary learning) on coefficient matrix $\mathbf{X}$.

- ICA

In ICA problem, we assume that the density of the independent components is double exponential (Laplace), which is a classical example of a supergaussian distribution. Many real applications, such as feature extraction and speech processing, show this distribution. Therefore, we have

$$-\log P(x_{i,j}) = \sqrt{2}\,|x_{i,j}| + C'$$

where C' is an irrelevant constant. Thus, our problem becomes

$$\min_{\mathbf{D},\widetilde{\mathbf{X}}} \ \frac{1}{2}\left\|\widetilde{\mathbf{Y}} - \mathbf{D}\widetilde{\mathbf{X}}\right\|_F^2 + \sqrt{2}\sigma\left\|\widetilde{\mathbf{X}}\right\|_{\ell_1} \tag{3.4}$$

Note that this problem of finding independent $\mathbf{X}$ is formally equivalent to that of dictionary learning. In our case, however, the ℓ_1 norm arises from the Laplacian prior rather than the sparsity prior.

- NMF

The objective function of NMF problem has been presented in (2.4)

$$\min_{\mathbf{D},\mathbf{X}} \frac{1}{2}\|\mathbf{Y}-\mathbf{D}\mathbf{X}\|_F^2 \qquad \text{such that } \mathbf{D}\geq 0, \mathbf{X}\geq 0 \tag{3.5}$$

It can be seen as non-negativity constriants on $\mathbf{D}$ and $\mathbf{X}$.

- DL

Extended from (2.6), the dictionary learning problem can be mathematically formulated as

$$\arg\min_{\mathbf{D},\mathbf{X}} \frac{1}{2}\|\mathbf{Y}-\mathbf{D}\mathbf{X}\|_F^2 + \lambda\varphi(\mathbf{X}) \text{ such that } \mathbf{D}\in\Delta \tag{3.6}$$

Here, $\varphi(\mathbf{X})$ is the sparsity prior, $\mathbf{Y}=[\mathbf{y}_1,\mathbf{y}_2,..,\mathbf{y}_N]$ and $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,..,\mathbf{x}_N]$ are the data and coefficient matrices, respectively. To avoid this scale ambiguity problem, it is a common practice to constrain the dictionary to a class specified by Δ.

3.2 A Fast Algorithm Design

We first use the variable splitting approach to rewrite (3.1) as

$$\min_{\mathbf{D},\mathbf{X},\mathbf{Z}} \frac{1}{2}\|\mathbf{Y}-\mathbf{D}\mathbf{X}\|_F^2 + \lambda\varphi(\mathbf{Z}) \quad \text{such that} \quad \mathbf{X}=\mathbf{Z};\ \mathbf{D}\in\Delta \tag{3.7}$$

Here, $\mathbf{Z}$ is an auxiliary variable used to simplify the optimization process. Using the AL framework to enforce the constraint ($\mathbf{X}=\mathbf{Z}$) in (3.7), we get the augmented Lagrangian function as

$$\mathcal{L}_\beta(\mathbf{X},\mathbf{Z},\mathbf{D}) = \frac{1}{2}\|\mathbf{Y}-\mathbf{D}\mathbf{X}\|_F^2 + \lambda\varphi(\mathbf{Z}) + \lambda\langle\mathbf{\Lambda},\mathbf{X}-\mathbf{Z}\rangle + \frac{\lambda\beta}{2}\|\mathbf{X}-\mathbf{Z}\|_F^2 \quad \text{such that} \quad \mathbf{D}\in\Delta \tag{3.8}$$

where $\mathbf{\Lambda}$ is the matrix of Lagrange multipliers and β is the penalty parameter. The inner product of two matrices is specified by $\langle\mathbf{A},\mathbf{B}\rangle = \text{trace}(\mathbf{A}^T\mathbf{B})$. The proposed

algorithm then alternates between the updates of the variables $\mathbf{X}$, $\mathbf{Z}$, $\mathbf{D}$ as well as the associated Lagrange parameters. The pseudocode for this algorithm is presented in Algorithm 3.1 and the update steps are shown in details as below.

Algorithm 3.1 Scheme of the proposed algorithm

Input: Given signals $\mathbf{Y}$

1: Initialization: $\mathbf{D}^0$, $\mathbf{X}^0$, $\mathbf{Z}^0$, $\mathbf{\Lambda}^0$, $\beta^0 > 0$

2: **while** stop-criterion not satisfied **do**

3: $\mathbf{Z}^{n+1} = \arg\min_{\mathbf{Z}} \|\mathbf{X}^n - \mathbf{Z}\|_F^2 + \frac{2}{\beta^n}\varphi(\mathbf{Z}) + \frac{2}{\beta^n}\langle \mathbf{\Lambda}^n, \mathbf{X}^n - \mathbf{Z}\rangle$;

4: $\mathbf{X}^{n+1} = \arg\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{Y} - \mathbf{D}^n\mathbf{X}\|_F^2 + \frac{\lambda\beta^n}{2}\|\mathbf{X} - \mathbf{Z}^{n+1}\|_F^2 + \lambda\langle \mathbf{\Lambda}^n, \mathbf{X} - \mathbf{Z}^{n+1}\rangle$;

5: $\mathbf{D}^{n+1} = \arg\min_{\mathbf{D}} \frac{1}{2}\|\mathbf{Y} - \mathbf{D}\mathbf{X}^{n+1}\|_F^2$ such that $\mathbf{D} \in \Delta$;

6: $\mathbf{\Lambda}^{n+1} = \mathbf{\Lambda}^n + \beta^n(\mathbf{X}^{n+1} - \mathbf{Z}^{n+1})$;

7: $\beta^{n+1} = \rho\beta^n$;

8: **end while**

Output: $\mathbf{D}$ and $\mathbf{X}$

3.2.1 Update of $\mathbf{Z}$

In this step, We assume $\mathbf{D}$ and $\mathbf{X}$ to be fixed. From (3.8), we get

$$\mathcal{L}_\beta(\mathbf{Z}, \mathbf{X}^n, \mathbf{D}^n) = \|\mathbf{X}^n - \mathbf{Z}\|_F^2 + \frac{2}{\beta}\langle \mathbf{\Lambda}, \mathbf{X}^n - \mathbf{Z}\rangle + \frac{2}{\beta}\varphi(\mathbf{Z}) \tag{3.9}$$

Here, different priors for coefficients $\mathbf{X}$ will result in different regularizers $\varphi(\mathbf{X})$. The problem becomes

$$\mathbf{Z}^{n+1} = \arg\min_{\mathbf{Z}} \mathcal{L}_\beta \left(\mathbf{Z}, \mathbf{X}^n, \mathbf{D}^n\right),$$

Later, we will show how to address this optimization problem according to different applications.

3.2.2 Update of $\mathbf{X}$

To optimize (3.8) with respect to $\mathbf{X}$, we keep $\mathbf{Z}$ and $\mathbf{D}$ fixed and update $\mathbf{X}$ as $\mathbf{X}^{n+1} = \arg\min_{\mathbf{X}} \mathcal{L}_\beta(\mathbf{X}, \mathbf{Z}^{n+1}, \mathbf{D}^n)$:

$$\mathbf{X}^{n+1} = \arg\min_{\mathbf{X}} \ \frac{1}{2}\left\|\mathbf{Y} - \mathbf{D^n X}\right\|_F^2 + \frac{\lambda\beta}{2}\left\|\mathbf{X} - \mathbf{Z}^{n+1}\right\|_F^2 + \lambda\left\langle \mathbf{\Lambda}^n, \mathbf{X} - \mathbf{Z}^{n+1}\right\rangle$$

This quadratic subproblem can be solved using conjugate gradients algorithm. In most cases, a few CG steps are sufficient. For small scale problems, the coefficients can be updated analytically as

$$\mathbf{X}^{n+1} = \left((\mathbf{D}^n)^T\mathbf{D}^n + \lambda\beta\,\mathbf{I}\right)^{-1}\left((\mathbf{D}^n)^T\mathbf{Y} + \lambda\beta\mathbf{Z}^{n+1} - \lambda\mathbf{\Lambda}^n\right) \tag{3.10}$$

Note that once the $K \times K$ matrix $\mathbf{Q} = \left((\mathbf{D}^n)^T\mathbf{D}^n + \lambda\beta\,\mathbf{I}\right)^{-1}$ is pre-computed, the coefficients for each of the dataset $\mathbf{x}_i^{n+1}$ are computed as $\mathbf{Q}\left(\mathbf{y}_i + \lambda\beta\,\mathbf{z}_i^{n+1} + \lambda\,\mathbf{\Lambda}_i^n\right)$.

According to the AL principle, the Lagrange multipliers $\mathbf{\Lambda}$ are updated at each step using the rule:

$$\mathbf{\Lambda}^{n+1} = \mathbf{\Lambda}^n + \beta(\mathbf{X}^{n+1} - \mathbf{Z}^{n+1}) \tag{3.11}$$

3.2.3 Update of $\mathbf{D}$ based on AL approach

In this step, we also use the variable splitting approach and get the update rule from 3.8

$$\min_{\mathbf{D},\mathbf{Q}} \frac{1}{2} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \quad \text{such that} \quad \mathbf{D} = \mathbf{Q}; \mathbf{Q} \in \Delta. \tag{3.12}$$

Here, $\mathbf{Q}$ is an auxiliary variable used to simplify the optimization process. Using the AL framework to enforce the constraint ($\mathbf{D} = \mathbf{Q}$), we get the augmented Lagrangian as

$$\mathcal{L}_\alpha(\mathbf{D}, \mathbf{Q}) = \frac{1}{2} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 + \langle \mathbf{\Gamma}, \mathbf{D} - \mathbf{Q} \rangle + \frac{\alpha}{2} \|\mathbf{D} - \mathbf{Q}\|_F^2 \quad \text{such that} \quad \mathbf{Q} \in \Delta. \tag{3.13}$$

where $\mathbf{\Gamma}$ is the matrix of Lagrange multipliers and α is the penalty parameter.

Solving for $\mathbf{Q}$, assuming $\mathbf{D}$ to be fixed, we obtain

$$\begin{aligned} \mathbf{Q}^{m+1} &= \arg\min_{\mathbf{Q}} \ \langle \mathbf{\Gamma}^m, \mathbf{D}^m - \mathbf{Q} \rangle + \frac{\alpha}{2} \|\mathbf{D}^m - \mathbf{Q}\|_F^2 \ \text{ such that } \ \mathbf{Q} \in \Delta \qquad (3.14) \\ &= \arg\min_{\mathbf{Q}} \ \left\| \underbrace{\mathbf{D}^m + \frac{1}{\alpha}\mathbf{\Gamma}^m}_{\mathbf{B}^m} - \mathbf{Q} \right\|_F^2 \ \text{ such that } \ \mathbf{Q} \in \Delta. \end{aligned}$$

Later we will use the projection scheme to solve this subproblem according to diferent constraints Δ.

We update $\mathbf{D}$ as

$$\mathbf{D}^{m+1} = \arg\min_{\mathbf{D}} \ \frac{1}{2} \left\|\mathbf{Y} - \mathbf{D}\mathbf{X}^{n+1}\right\|_F^2 + \left\langle \mathbf{\Gamma}^m, \mathbf{D} - \mathbf{Q}^{m+1} \right\rangle + \frac{\alpha}{2} \left\|\mathbf{D} - \mathbf{Q}^{m+1}\right\|_F^2 \tag{3.15}$$

This quadratic subproblem can be solved using conjugate gradients algorithm. In most cases, a few CG steps are sufficient. For small scale problems, the dictionary

can be updated analytically as

$$\mathbf{D}^{m+1} = \left(\mathbf{Y}\left(\mathbf{X}^{n+1}\right)^T + \alpha\mathbf{Q}^{m+1} - \mathbf{\Gamma}^m\right)\left(\mathbf{X}^{\mathbf{n+1}}\left(\mathbf{X}^{n+1}\right)^T + \alpha\mathbf{I}\right)^{-1} \tag{3.16}$$

Also, the Lagrange multipliers $\mathbf{\Gamma}$ are updated at each step using the rule:

$$\mathbf{\Gamma}^{m+1} = \mathbf{\Gamma}^m + \alpha(\mathbf{D}^{m+1} - \mathbf{Q}^{m+1}) \tag{3.17}$$

The detailed description of this framework is listed in Algorithm 3.2. And this framework is flexible enough to account for arbitrary constraints on $\mathbf{D}$. Later, we will show how to use projection operators to enforce different constraints $\mathbf{Q} \in \Delta$.

Algorithm 3.2 The detailed scheme for updating dictionary $\mathbf{D}$

Input: Given signals $\mathbf{Y}$ and $\mathbf{X}$

1: Initialization: $\mathbf{D}^0$, $\mathbf{Q}^0$, $\mathbf{\Gamma}^0$, $\alpha^0 > 0$

2: **while** stop-criterion not satisfied **do**

3: $\mathbf{Q}^{m+1} = \arg\min_{\mathbf{Q}} \langle \mathbf{\Gamma}^m, \mathbf{D}^m - \mathbf{Q}\rangle + \frac{\alpha}{2}\left\|\mathbf{D}^m - \mathbf{Q}\right\|_F^2$ such that $\mathbf{Q} \in \Delta$;

4: $\mathbf{D}^{m+1} = \arg\min_{\mathbf{D}} \frac{1}{2}\left\|\mathbf{Y} - \mathbf{D}\mathbf{X}^{n+1}\right\|_F^2 + \langle \mathbf{\Gamma}^m, \mathbf{D} - \mathbf{Q}^{m+1}\rangle + \frac{\alpha}{2}\left\|\mathbf{D} - \mathbf{Q}^{m+1}\right\|_F^2$;

5: $\mathbf{\Gamma}^{m+1} = \mathbf{\Gamma}^m + \alpha^m(\mathbf{D}^{m+1} - \mathbf{Q}^{m+1})$;

6: $\alpha^{m+1} = \rho\alpha^m$;

7: **end while**

8: $\mathbf{D}_0 = \mathbf{D}_m$

Output: $\mathbf{D}$

In summary, the proposed algorithm consists of two-level nested loop. The

inner loop updates the dictionary and enforce the constraints while the outer loop minimizes the objective function to update the coefficient matrix and enable the accuracy of the method. To accelerate this scheme, we choose to update dictionary $\mathbf{D}$ and coefficient matrix $\mathbf{X}$ at the same time in the inner loop as K-SVD algorithm does. The detailed description of this improved scheme is shown in Algorithm 3.3.

3.3 Application to Dictionary Learning

In this section, we focus on apply our proposed algorithm to address dictionary learning problem according to different constraints on dictionary $\mathbf{D}$. We will apply Algorithm 3.3 to address the problem (3.6) according to different Δ. Since we have already got the updating rules for $\mathbf{X}$ and $\mathbf{D}$, the next step is to introduce the rules for updating $\mathbf{Z}$ and $\mathbf{Q}$.

3.3.1 Update of $\mathbf{Z}$

Based on the recent success of non-convex sparsity penalty as in [20], we consider Schatten p-norms on entries of coeffcient matrix $\mathbf{X}$ as the sparsity prior $\varphi(\mathbf{X})$. The ℓ_p norm better approximates the original NP hard problem than ℓ_1. It can be specified by

$$\|\mathbf{X}\|_{\ell_p} = \left(\sum_{i,j} |x_{ij}|^p \right)^{1/p}, \ 0 \leq p \leq 1$$

Solving for $\mathbf{Z}$, we obtain

$$\begin{aligned}\mathbf{Z}^{n+1} &= \arg\min_{\mathbf{Z}} \ \|\mathbf{X}^n - \mathbf{Z}\|_F^2 + \frac{2}{\beta}\|\mathbf{Z}\|_{\ell_p} + \frac{2}{\beta}\langle \mathbf{\Lambda}^n, \mathbf{X}^n - \mathbf{Z}\rangle \\ &= \arg\min_{\mathbf{Z}} \left\| \underbrace{\mathbf{X}^n + \frac{1}{\beta}\mathbf{\Lambda}^n}_{\mathbf{T}^n} - \mathbf{Z} \right\|_F^2 + \frac{2}{\beta}\|\mathbf{Z}\|_{\ell_p}\end{aligned}$$

This problem is analytically solved by shrinkage of the entries of $\mathbf{T}^n$ [9]

$$\mathbf{Z}^{n+1} = \frac{\mathbf{T}^n}{|\mathbf{T}^n|}\left(|\mathbf{T}^n| - \frac{1}{\beta}|\mathbf{T}^n|^{p-1}\right)_+ . \tag{3.18}$$

3.3.2 Update of $\mathbf{Q}$

In this step, we will derive the optimum of (3.14) under different dictionary constraints. Commonly used constraints are Frobenius-norm constraints $\|\mathbf{D}\|_F^2 = 1$ and the column-norm constraints $\|\mathbf{d}_i\|_{\ell_2} = 1;\ 0 < i \leq K$, which are non-convex. Our algorithm can be extented to many other constraints as listed in the following.

- Dictionaries with Convex Column-Norm Constraints

Instead of considering non-convex column-norm constraints, the convex column-norm constraints can be defined as

$$\Delta_C = \left\{\mathbf{Q}_{M\times K} : \|\mathbf{q}_j\|_2^2 \leq c\right\}, j = 1, 2, ..., K$$

From (3.6), if $\|\mathbf{b}_j\|_2^2 \leq c$, we update $\mathbf{q}_j = \mathbf{b}_j$. Otherwise we scale the column to have the largest acceptable norm($c^{1/2}$).

$$\mathbf{Q}^* = \left\{\mathbf{q}_j^*\right\}_{j=1,2,...,K}$$

$$\mathbf{q}_j^* = \begin{cases} \mathbf{b}_j & \|\mathbf{b}_j\|_2^2 \leq c \\ \frac{c^{1/2}}{\|\mathbf{b}_j\|_2}\mathbf{b}_j & \text{else} \end{cases}$$

where $\mathbf{q}_j$ and $\mathbf{b}_j$ are the j^{th} columns of $\mathbf{Q}$ and $\mathbf{B}$ respectively.

- Dictionaries with Convex Frobenius-Norm Constraints

Compared with the dictionaries with column-norm constraints, the dictionaries under Frobenius-norm constraints can have columns with different norms, which means atoms with large norms have more chance to present in the approximation. This property can improve its performance when right weights are chosen. The convex Frobenius-norm constraints can be defined as

$$\Delta_F = \left\{ \mathbf{Q}_{M\times K} : \|\mathbf{Q}\|_F^2 \leq c \right\},$$

If $\mathbf{B} \in \Delta_F$, we can update $\mathbf{Q}$ as $\mathbf{Q} = \mathbf{B}$. Otherwise we scale $\mathbf{B}$ to have Frobenius-norm $c^{1/2}$.

$$\mathbf{Q}^* = \begin{cases} \mathbf{B} & \|\mathbf{B}\|_F^2 \leq c \\ \frac{c^{1/2}}{\|\mathbf{B}\|_F}\mathbf{B} & \text{else} \end{cases}$$

- Dictionaries with Joint Sparsity Constraints

To satisfy the requirement for small size dictionary in some application, the joint sparsity constraint is introduced to encourage dictionary reduction. It can be defined as

$$\Delta_J = \{ \mathbf{Q}_{M\times K} : \mathbf{J}_p(\mathbf{Q}) \leq c,\ p = 1, 2 \text{ or } \infty \},$$
$$\mathbf{J}_p(\mathbf{Q}) = \sum_j \left(\sum_i |q_{ij}|^p \right)^{1/p}$$

When $p = 1$, $\mathbf{J}_p(\mathbf{Q}) = \|\mathbf{Q}\|_{l_1}$ is the ℓ_1-norm of $\mathbf{Q}$. And when $p = 2$, $\mathbf{J}_p(\mathbf{Q}) = \left\| \|\mathbf{q}_j\|_{\ell_2} \right\|_{\ell_1}$ is defines as the ℓ_1-$\ell_2$norm of $\mathbf{Q}(\|\mathbf{Q}\|_{\ell_1\text{-}\ell_2})$.

With the help of a Lagrangian multiplier θ, (3.14) becomes to

$$\mathbf{Q}^{m+1} = \arg\min_{\mathbf{Q}} \ \|\mathbf{B}^m - \mathbf{Q}\|_F^2 + \theta(\mathbf{J}_p(\mathbf{Q}) - c). \tag{3.19}$$

A. $p = 1$ **case**

The problem turns to

$$\mathbf{Q}^{m+1} = \arg\min_{\mathbf{Q}} \ \|\mathbf{B}^m - \mathbf{Q}\|_F^2 + \theta(\|\mathbf{Q}\|_{l_1} - c). \tag{3.20}$$

This problem is analytically solved by shrinkage of the entries of $\mathbf{B}^m$:

$$\mathbf{Q}^{m+1} = \frac{\mathbf{B}^m}{|\mathbf{B}^m|}\left(|\mathbf{B}^m| - \frac{\theta}{2}\right)_+.$$

To satisfy the constraint Δ_J, we use Bisection method to find the optimal θ.

B. $p = 2$ **case**

The problem turns to

$$\mathbf{Q}^{m+1} = \arg\min_{\mathbf{Q}} \ \|\mathbf{B}^m - \mathbf{Q}\|_F^2 + \theta(\|\mathbf{Q}\|_{l_1\text{-}l_2} - c). \tag{3.21}$$

To simplify it, we get

$$\mathbf{q}_j^{m+1} = \arg\min_{\mathbf{q}_j} \sum_j \left\|\mathbf{b}_j^m - \mathbf{q}_j\right\|_F^2 + \theta \left\|\mathbf{q}_j\right\|_{l_2} \ j = 1, 2, ..., K. \tag{3.22}$$

where $\mathbf{q}_j$ is the j^{th} column of $\mathbf{Q}$. According to the thresholding operator, we get the optimal solution for (3.22)

$$\mathbf{q}_j^* = \begin{cases} \mathbf{0} & \|\mathbf{b}_j\|_2 \le \frac{\theta}{2} \\ \frac{\|\mathbf{b}_j\|_2 - \theta/2}{\|\mathbf{b}_j\|_2}\mathbf{b}_j & \text{else} \end{cases}$$

- Dictionaries with a Tight Frame

It is currently popular to use a fixed orthogonal bases such as wavelet bases and adaptively chosen orthogonal bases for noise removal [3]. Therefore, we consider a uniform normalized tight frame, that is, an orthogonal basis as dictionary constraint.

The constraint can be defined as

$$\Delta_T = \left\{ \mathbf{Q}_{M\times K} : \mathbf{Q}^T\mathbf{Q} = \mathbf{I} \right\}$$

where $\mathbf{I}$ is the identical operator. In (3.14), the problem is to

$$\min_{\mathbf{Q}} \|\mathbf{B} - \mathbf{Q}\|_F^2 = \min_{\mathbf{Q}} Tr(\mathbf{B}^T\mathbf{B}) + Tr(\mathbf{Q}^T\mathbf{Q}) - 2Tr(\mathbf{Q}^T\mathbf{B})$$

Since the first two terms are constants, the problem becomes

$$\max_{\mathbf{Q}} \ Tr(\mathbf{Q}^T\mathbf{B}) \text{ such that } \ \mathbf{Q} \in \Delta_T$$

Using Lagrangian multipliers method, we define the Lagrangian function as

$$L(\mathbf{Q}, \mathbf{\Theta}) = \langle \mathbf{Q}, \mathbf{B} \rangle - \frac{1}{2} \left\langle \mathbf{\Theta}, \mathbf{Q}^T\mathbf{Q} - \mathbf{I} \right\rangle$$

where $\mathbf{\Theta}$ is a Lagrangian multiplier matrix. Using the constraint $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$, (3.3.2) can be rewritten as

$$L(\mathbf{Q}, \mathbf{\Theta}) = -\sum_{j=1}^{K} \mathbf{q}_j^T\mathbf{b}_j + \sum_{j=1}^{K} \left(\frac{\theta_{j,j}}{2}(\mathbf{q}_j^T\mathbf{q}_j - 1) + \sum_{l>j}^{K} \theta_{j,l}(\mathbf{q}_j^T\mathbf{q}_l) \right)$$

where $\mathbf{q}_j$ represents the jth column of $\mathbf{Q}$. By setting $\frac{\partial L}{\partial \mathbf{q}_j} = 0$, we can get

$$\mathbf{b}_j = \theta_{j,j}\mathbf{q}_j + \theta_{j,l}\mathbf{q}_l$$

which can be written as a matrix form

$$\mathbf{B} = \mathbf{Q}\mathbf{\Omega}$$

where $\mathbf{\Omega}_{i,j} = \theta_{j,i}$. If $\mathbf{\Omega}$ is invertible, then

$$\mathbf{Q} = \mathbf{B}\mathbf{\Omega}^{-1}$$

Suppose that the SVD of $\mathbf{B}$ is $\mathbf{B} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ with $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, and $\mathbf{V}^T\mathbf{V} = \mathbf{V}\mathbf{V}^T = \mathbf{I}$, then we have

$$\begin{aligned} Tr(\mathbf{Q}^T\mathbf{B}) &= Tr\left(\left(\mathbf{\Omega}^{-1}\right)^T\mathbf{B}^T\mathbf{B}\right) \\ &= Tr\left(\left(\mathbf{\Omega}^{-1}\right)^T\mathbf{V}\mathbf{\Sigma}\mathbf{U}^T\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\right) \\ &= Tr\left(\left(\mathbf{\Omega}^{-1}\right)^T\mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T\right) \end{aligned}$$

$$\begin{aligned} \mathbf{I} &= \mathbf{Q}^T\mathbf{Q} \\ &= \left(\mathbf{\Omega}^{-1}\right)^T\mathbf{B}^T\mathbf{B}\mathbf{\Omega}^{-1} \\ &= \left(\mathbf{\Omega}^{-1}\right)^T\mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T\mathbf{\Omega}^{-1} \end{aligned}$$

Let $\mathbf{A} = \mathbf{V}^T\mathbf{\Omega}^{-1}\mathbf{V}$, we get

$$\begin{aligned} Tr(\mathbf{Q}^T\mathbf{B}) &= Tr\left(\mathbf{V}^T\left(\mathbf{\Omega}^{-1}\right)^T\mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T\mathbf{V}\right) \\ &= Tr\left(\mathbf{A}^T\mathbf{\Sigma}^2\right) \end{aligned}$$

$$\begin{aligned} \mathbf{I} &= \mathbf{V}^T\left(\mathbf{\Omega}^{-1}\right)^T\mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T\mathbf{\Omega}^{-1}\mathbf{V} \\ &= \mathbf{A}^T\mathbf{\Sigma}^2\mathbf{A} \end{aligned} \tag{3.23}$$

To satisfy (3.23), $\mathbf{A}$ is a diagonal matrix with $\mathbf{A}_{ii} = \mathbf{\Sigma}_{ii}^{-1}$. Therefore, $\mathbf{\Omega}^{-1} = \mathbf{V}\mathbf{A}\mathbf{V}^T = \mathbf{V}\mathbf{\Sigma}^{-\mathbf{1}}\mathbf{V}^T$ and then

$$\begin{aligned} \mathbf{Q}^* &= \mathbf{B}\mathbf{\Omega}^{-1} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{V}\mathbf{\Sigma}^{-\mathbf{1}}\mathbf{V}^T \\ &= \mathbf{U}\mathbf{V}^T \end{aligned}$$

Algorithm 3.3 The improved scheme of Algorithm 3.1

Input: Given signals $\mathbf{Y}$

Initialization: $\mathbf{D}^0$, $\mathbf{X}^0$, $\mathbf{Z}^0$,$\mathbf{Q}^0$, $\mathbf{\Lambda}^0$, $\mathbf{\Gamma}^0$, $\beta^0 > 0$

while stop-criterion not satisfied (loop in n) **do**

  Initialization: $\alpha^0 > 0$

  while stop-criterion not satisfied (loop in m) **do**

    $\mathbf{Z}^{n,m+1} = \arg\min_{\mathbf{Z}} \|\mathbf{X}^{n,m} - \mathbf{Z}\|_F^2 + \frac{2}{\beta^n}\varphi(\mathbf{Z}) + \frac{2}{\beta^n}\left\langle \mathbf{\Lambda}^{n,m}, \mathbf{X}^{n+1} - \mathbf{Z}\right\rangle$;

    $\mathbf{X}^{n,m+1} = \arg\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{Y} - \mathbf{D}^{n,m}\mathbf{X}\|_F^2 + \frac{\lambda\beta^n}{2}\left\|\mathbf{X} - \mathbf{Z}^{n,m+1}\right\|_F^2 + \lambda\left\langle \mathbf{\Lambda}^{n,m}, \mathbf{X} - \mathbf{Z}_{n,m+1}\right\rangle$;

    $\mathbf{Q}^{n,m+1} = \arg\min_{\mathbf{Q}} \left\langle \mathbf{\Gamma}^{n,m}, \mathbf{D}^{n,m} - \mathbf{Q}\right\rangle + \frac{\alpha^{n,m}}{2}\|\mathbf{D}^{n,m} - \mathbf{Q}\|_F^2$ such that $\mathbf{Q} \in \Delta$;

    $\mathbf{D}^{n,m+1} = \arg\min_{\mathbf{D}} \frac{1}{2}\left\|\mathbf{Y} - \mathbf{D}\mathbf{X}^{n,m+1}\right\|_F^2 + \left\langle \mathbf{\Gamma}^n, \mathbf{D} - \mathbf{Q}^{n,m}\right\rangle + \frac{\alpha^{n,m}}{2}\|\mathbf{D} - \mathbf{Q}^{n,m}\|_F^2$;

    $\mathbf{\Gamma}^{n,m+1} = \mathbf{\Gamma}^{n,m} + \alpha^{n,m}(\mathbf{D}^{n,m+1} - \mathbf{Q}^{n,m+1})$;

    $\mathbf{\Lambda}^{n,m+1} = \mathbf{\Lambda}^{n,m} + \beta^n(\mathbf{X}^{n,m+1} - \mathbf{Z}^{n,m+1})$;

    $\alpha^{n,m+1} = \rho\alpha^{n,m}$;

  end while

  $\beta^{n+1} = \rho\beta^n$;

  $\mathbf{X}^{n,0} = \mathbf{X}^{n,m}$; $\mathbf{Z}^{n,0} = \mathbf{Z}^{n,m}$; $\mathbf{D}^{n,0} = \mathbf{D}^{n,m}$; $\mathbf{Q}^{n,0} = \mathbf{Q}^{n,m}$; $\mathbf{\Lambda}^{n,0} = \mathbf{D}^{n,m}$; $\mathbf{\Gamma}^{n,0} = \mathbf{Q}^{n,m}$;

end while

Output: $\mathbf{D}$ and $\mathbf{X}$

CHAPTER 4
EXPERIMENTAL RESULTS

We determine the ability of the different algorithms in recovering the synthetic dictionaries from training data generated using them. We also demonstrate the utility of the algorithm in the the context of patch based 2D MR image denoising [4] and 3D dynamic image denoising.

4.1 Representation Results for Synthetic Data

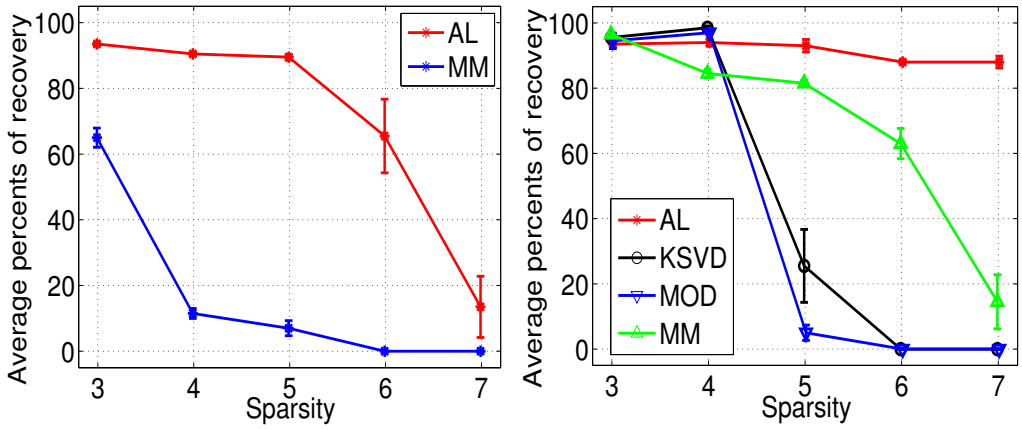
We consider a 20×40 random dictionary $\mathbf{D}$ with normalized columns. 1280 training samples are generated by taking weighted linear combination of its atoms. The weights are assumed to be k-sparse, where k is a pre-determined constant. Both the sparsity patterns and the magnitude of the coefficients are assumed to be randomly distributed. Dictionaries are learned using different algorithms and the learned dictionaries are compared with the ground truth. The training test is repeated for five different training datasets to ensure fair comparisons. If the squared error between a learned and true atom is below 1%, it is classified as correctly identified.

We compare the proposed method (DL-AL) against MOD [5], K-SVD [1], and dictionary learning with majorize minimize algorithm (DL-MM) [22]. The implementations of K-SVD, MOD, and DL-MM were downloaded from the webpages of the authors. The ℓ_0 norm of the sparse vectors in MOD and KSVD were set to k from 3 to 7, while we set $\lambda = 0.1; p = 0.5$; $c = 1$ for column-norm constraints and $c = K$ for Frobenius-norm constraints in DL-AL and DL-MM; these parameters yielded a

sparsity approximately equal to k.

The average percentages and standard deviations of correctly recovered atoms using dictionary learning methods with MM and AL under Frobenius-norm constraints are shown in Fig. 1(a) and using the four dictionary learning methods above under column-norm constraints shown in Fig. 1(b). We observe that the DL-AL scheme provides better recovery for almost all sparsity levels in the column-norm dictionary constraints problem, compared to other methods. Besides, DL-AL scheme with column-norm dictionary constraints performs better than that with Frobenius-norm constaints.

Fig. 2 shows the average percentages and standard deviations of correctly recovered atoms under both dictionary constraints for ℓ_1 norm and ℓ_p norm problems. The results of ℓ_p norm problem present higher recovery for both constraints. The comparison of execution times in Fig. 3 show that the complexity of the proposed scheme grows much slower than the competing methods, making it desirable in large scale imaging problems. We study the sensitivity of the algorithms to local minima by initializing the dictionaries as random matrices with Gaussian (randn) or uniform distributed entries (rand) in Fig. 4. The variance of the results are shown by the error bars. We observe that the AL-DL method is relatively insensitive to the initialization, resulting in smaller error bars and results that are insensitive to the distribution of the initialization.



(a) Frobenius-norm constraints (b) Column-norm constraints

Figure 4.1: Comparison of the dictionary recovery success rates using (a)MM and AL dictionary learning methods with Frobenius-norm constraints; (b)K-SVD, MOD, MM, AL dictionary learning methods with column-norm constraints. We observe that the proposed scheme is able to provide good recovery at high sparsity levels, compared to classical methods.

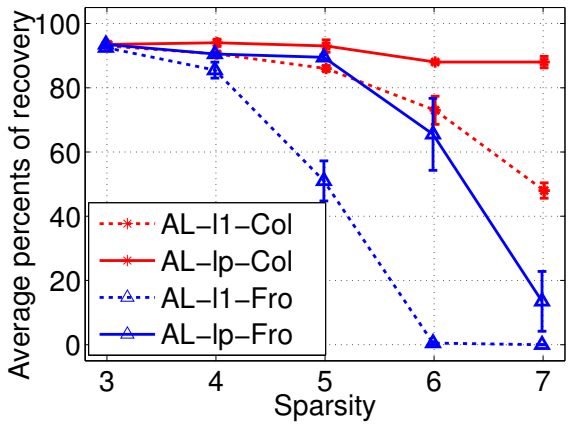


Figure 4.2: A comparison of the dictionary recovery success rates using AL dictionary learning methods with ℓ_p norm and ℓ_1 norm sparse penalty term.

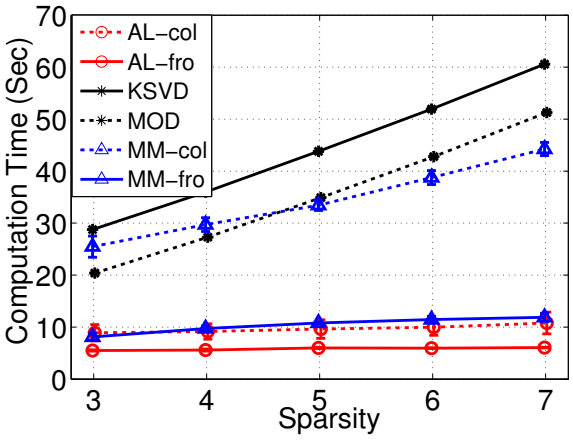


Figure 4.3: A comparison of computation time of the algorithms as a function of the sparsity. We observe that the computation time of the proposed algorithm increases at a much lower pace than the competing algorithms.

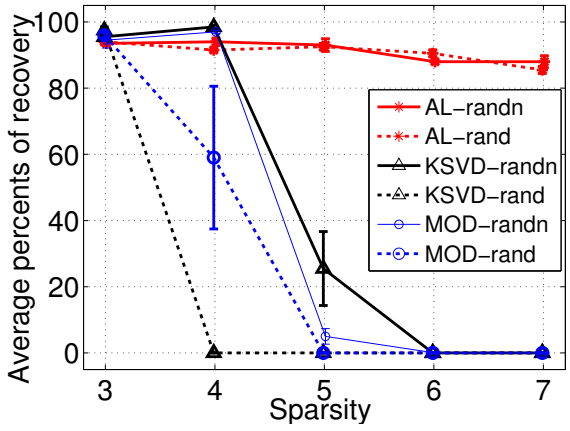


Figure 4.4: A comparison of the dictionary recovery success rates using K-SVD, MOD, AL dictionary learning methods with different dictionary initializations.

4.2 Denoising Results for Real Data

4.2.1 2D Image Denoising

We demonstrate the utility of the dictionary learning scheme in denoising brain MR image of size 256×256 in Figures 4.5 and 4.6; this approach exploits the redundancy of the patches in the image. The original image in Fig. 4.5.(a) is contaminated by a random zero-mean Gaussian noise with $\sigma = 8$ as shown in Fig. 4.5.(b). We extract all possible patches of size 6×6 from the contaminated image. The dictionary is trained using one tenth of these patches. We use a random initialization of the dictionary with 36 atoms. We expect the dictionary elements to be noise free, thanks to the averaging of similar patches. Once the dictionary is estimated, all the the noisy patches are denoised by considering their sparse approximation with the learned dictionary. The final image is then recovered by the weighted averaging of the image patches. We compare the quality of the denoised images in Fig. 4.5. The quality of the reconstructed images are determined using the signal to noise ratio (SNR) between the reconstruction and original images. The SNR of the images is determined as

$$SNR = -10 \log_{10} \frac{\|\mathbf{I}_{\text{recon}} - \mathbf{I}_{\text{orig}}\|_F^2}{\|\mathbf{I}_{\text{orig}}\|_F^2} \tag{4.1}$$

where $\mathbf{I}_{\text{recon}}$ is the reconstructed image; $\mathbf{I}_{\text{orig}}$ is the original image; $\|\cdot\|$ is the Frobenius norm.

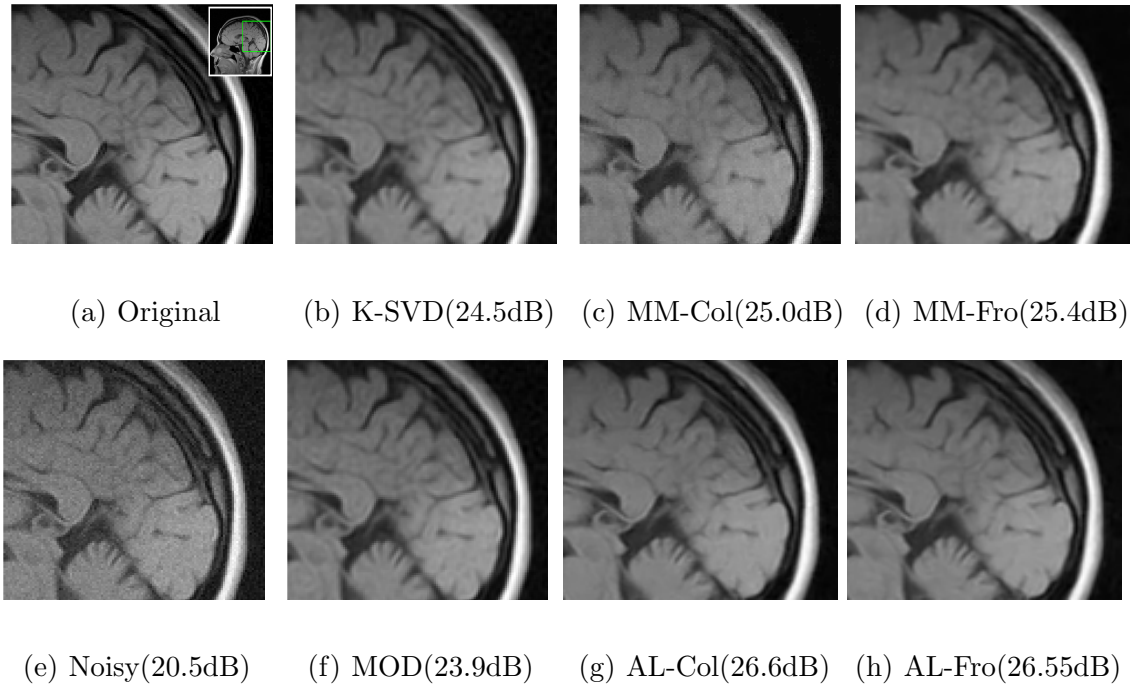
We compare the convergence rate of the algorithms with respect to computation time in Fig. 4.6. The algorithms are implemented in MATLAB R2012a and run on a Intel Xeon processor with 34 GB of RAM. We observe that the proposed scheme

converge in around 15 seconds, while KSVD and MOD algorithms take around 300 seconds. The MM algorithm is observed to be the slowest. To make the competition fare, the sparsity parameter in KSVD and MOD algorithms are set to four, while the λ parameter in DL-MM and DL-AL algorithms are chosen such that the average sparsity of the reconstructions is around four. We observe that the proposed DL-AL scheme is much faster and capable of providing improved reconstructions; this may be attributed to the quality of the learned dictionaries.

To test the sensitivity to initialization, we consider four different dictionary initializations of size 36×36. They are the 2D DCT matrix, the identity matrix, a random matrix with gaussian entries, and the Karhunen-Loeve Transform(KLT). Fig. 4.7 shows the objective function over computation time of the algorithm for different initializations. The objective function converges very fast and has nearly identical final values for all cases. This indicates that out algorithm is robust to initialization.

The next experiment is to evaluate the sensitivity of the proposed algorithm to parameter choosing. There are three parameters needed to set: sparsity regularization parameter (λ), patch size in square dictionary, and dictionary atom number(K). We vary one parameter at a time with other parameters fixed at nominal values in previous experiments. The top three figures in Fig. 4.8 show the results for column-norm constraints and the rest ones for Frobenius-norm constraints. Fig. 4.8(a) plotted SNR and average sparsity level versus λ. The sparsity level decreases as a function of λ. On the other hand, SNR is increasing until $\lambda = 65$ and the corresponding sparsity is around 4. The poorer performance at the low λ, that is high sparsity levels such as

10, is due to learning aliasing artifacts and noise in the dictionary training. At low sparsity levels, the algorithm loses resolution and more information of data thereby degrading performance. The performance with respect to patch size is improved when the patch size is increased from 4×4 to 6×6. However, when the size is higher than 6×6, the SNR becomes decreasing. Also, the increase in patch size increases average sparsity level. Therefore, using large patch size, we need to choose greater λ to enforce lower sparsity to get good results. The SNR and average sparsity level versus dictionary atom number are plotted in Fig. 4.8(c). The SNRs for dictionaries of sizes 36×16 to 36×196 are lower than SNR with squre dictionary of size 36×36, but the change in sparsity is rather small. However, to get good denoising results, we need to make the average sparsity lower since the dictionary with large number of atom may contain more information of both original image and noise. Comparing Fig. 4.8(c) wirh Fig. 4.8(f), we notice that the algorithm with Frobenius-norm constraints performs slightly better in SNR than that with column-norm constraints when the atom number is high. To show the reason of this observation, Fig. 4.9 shows the structure of learned dictionaries for both constraints when dictionary atom number is $K = 144$. We find that the dictionary learned from column-norm constraints is more noisy while the dictionary with Frobenius-norm constraints contains more information from original image.



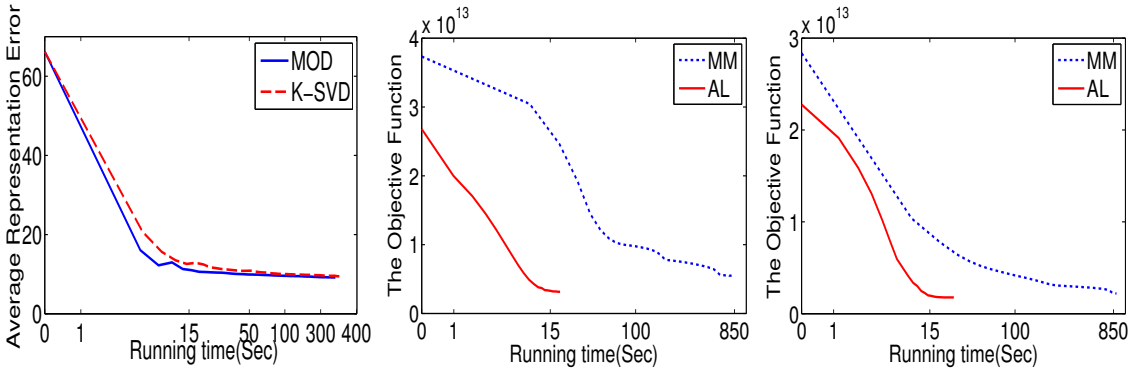
(a) Original (b) K-SVD(24.5dB) (c) MM-Col(25.0dB) (d) MM-Fro(25.4dB)

(e) Noisy(20.5dB) (f) MOD(23.9dB) (g) AL-Col(26.6dB) (h) AL-Fro(26.55dB)

Figure 4.5: Comparison of the denoising performance of different algorithms. (a) and (e) are the actual and noisy brain MR images, respectively. (b), (c), (d), (f), (g) and (h) show the reconstructions using K-SVD, DL-MM with column-norm constaints, DL-MM with Frobenius-norm constaints, MOD, DL-AL with column-norm constaints, and DL-AL with Frobenius-norm constaints, respectively. Note that the DL-AL method provides improved reconstructions, suggesting better learned dictionaries.

4.2.2 3D Image Denoising

In this experiment, we implement our algorithm to dynamic MRI denoising which is important for many clinical exams such as cardiac, perfusion, and functional



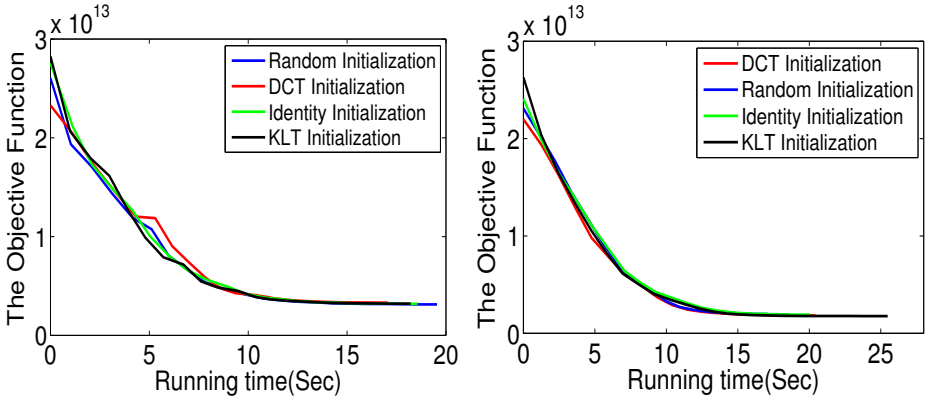
(a) ℓ_0 norm (b) ℓ_p norm(Col) (c) ℓ_p norm(Fro)

Figure 4.6: Comparison of the convergence of the different algorithms: (a) shows the average representation error of K-SVD and MOD. (b) shows the objective functions of DL-MM and DL-AL with column-norm constraints. (c) shows the objective functions of DL-MM and DL-AL with Frobenius-norm constraints. Note that K-SVD and MOD aims to minimize the representation error with a fixed sparsity, while DL-MM and DL-AL uses the cost function specified (3.6).The plots show that the running time of the proposed algorithm is much smaller than that of the competing algorithms.

imaging. Firstly, we represent the 3D dataset as a $\mathbf{M} \times \mathbf{N}$ matrix $\mathbf{Y}$

$$\mathbf{Y}_{\mathbf{M}\times\mathbf{N}} = \begin{bmatrix} y(\delta_1, t_1) & \cdot & \cdot & \cdot & y(\delta_N, t_1) \\ y(\delta_1, t_2) & \cdot & \cdot & \cdot & y(\delta_N, t_2) \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ y(\delta_1, t_M) & \cdot & \cdot & \cdot & y(\delta_N, t_M) \end{bmatrix}$$

where $\mathbf{M}$ and $\mathbf{N}$ represent the number of voxels in one frame image and the number of image frames in the dataset, respectively. From the structure of Y, the i^{th} row corresponds the reshaped i^{th} frame image and the i^{th} column corresponds the i^{th}



(a) Column-norm constraints (b) Frobenius-norm constraints

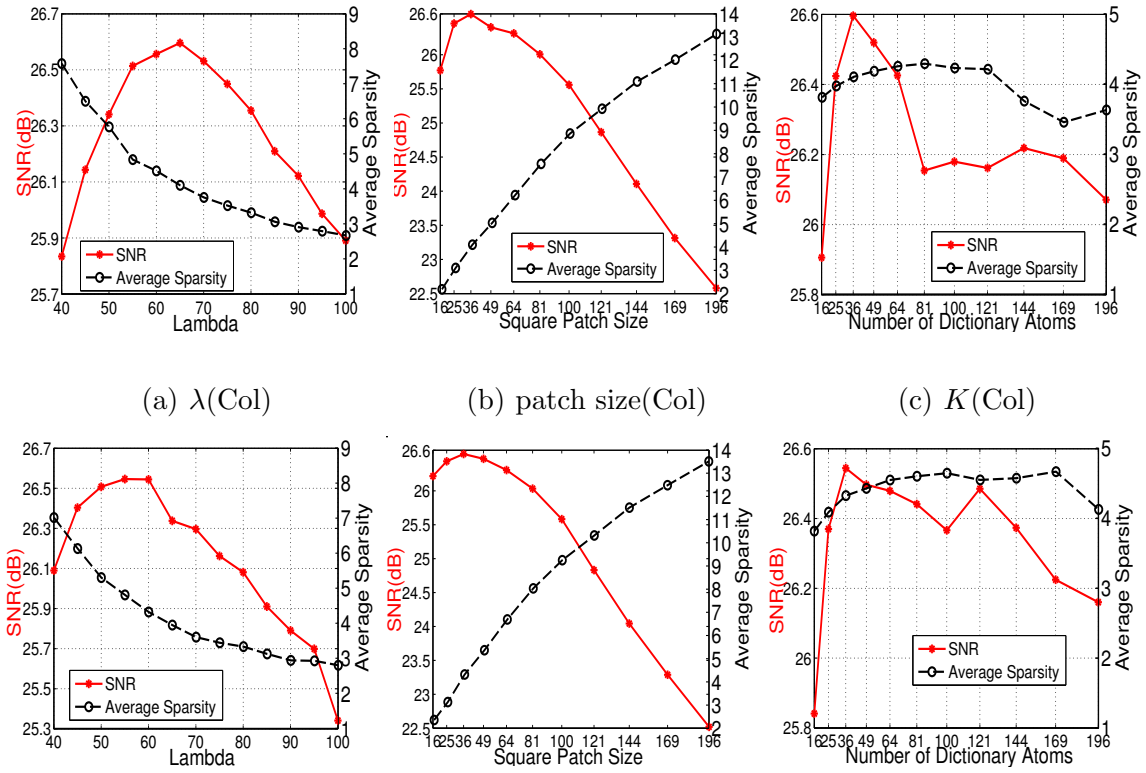
Figure 4.7: The objective function versus computation time using DCT, KLT, identity and random dictionary initializaitions.

voxel in different frames. Using the matrix factorization idea, we model $\mathbf{Y}$ as

$$\underbrace{\begin{bmatrix} y(\delta_1,t_1) & \cdot & \cdot & \cdot & y(\delta_N,t_1) \\ y(\delta_1,t_2) & \cdot & \cdot & \cdot & y(\delta_N,t_2) \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ y(\delta_1,t_M) & \cdot & \cdot & \cdot & y(\delta_N,t_M) \end{bmatrix}}_{\mathbf{Y_{M\times N}}} = \underbrace{\begin{bmatrix} d_1(t_1) & \cdot & \cdot & \cdot & d_K(t_1) \\ d_1(t_2) & \cdot & \cdot & \cdot & d_K(t_2) \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ d_1(t_M) & \cdot & \cdot & \cdot & d_K(t_M) \end{bmatrix}}_{\mathbf{D_{M\times K}}} \times \underbrace{\begin{bmatrix} x_1(\delta_1) & \cdot & \cdot & \cdot & x_1(\delta_N) \\ x_2(\delta_1) & \cdot & \cdot & \cdot & x_2(\delta_N) \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ x_K(\delta_1) & \cdot & \cdot & \cdot & x_K(\delta_N) \end{bmatrix}}_{\mathbf{X_{K\times N}}}$$

where K is the atom number of dictionary, the i^{th} column of $\mathbf{D}$ ($\mathbf{d}_i(\mathbf{t})$) represents the i^{th} temporal basis function and the i^{th} row of $\mathbf{X}$ ($\mathbf{x}_i(\delta)$) represents the i^{th} spatial weight.

Our goal is to reconstruct the dynamic image of size $128 \times 128 \times 70$ from the

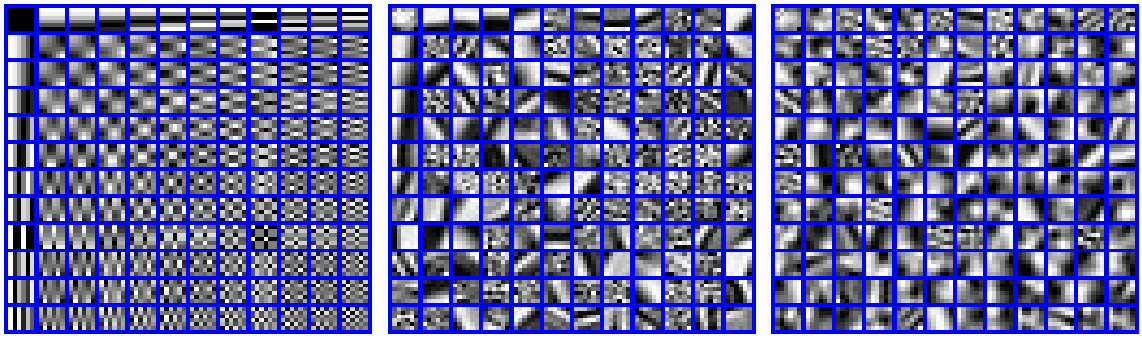


(a) λ(Col) (b) patch size(Col) (c) K(Col)

(d) λ(Fro) (e) patch size(Fro) (f) K(Fro)

Figure 4.8: Parameter evaluation. (a) SNR and average sparsity level versus λ. (b) SNR and average sparsity level versus patch size for square dictionary. (c) SNR and average sparsity level versus overcompleteness of dictionary.

noised image which is contaminated by a random zero-mean Gaussian noise. Fig 4.10 shows the comparisons on the perfusion MRI dataset with the standard deviation of the measurement noise $\sigma = 10$. We observe that our algorithm provides about 4dB improvement in SNR compared to K-SVD and MOD algorithms. The reconstructed images by K-SVD and MOD suffer from noisy artifacts and blur, while the errors in the our scheme are smaller and less concentrated at the edges, which represents much



(a) DCT (b) Col (c) Fro

Figure 4.9: Structure of dictionaries of size 36×144. (a) the initialized dictionary using DCT. (b) the dictionary trained from DL-AL with column-norm constraints. (c) the dictionary trained from DL-AL with Frobenius-norm constraints. From the figure (b), it appears that some of the atoms obtained are very noisy and uninformative using column-norm constraints on dictionary. However, We the dictionary in (c) contains more atoms that correspond to the textured regions in the original image, indicating that the dictionary adapts well to the content of interest.

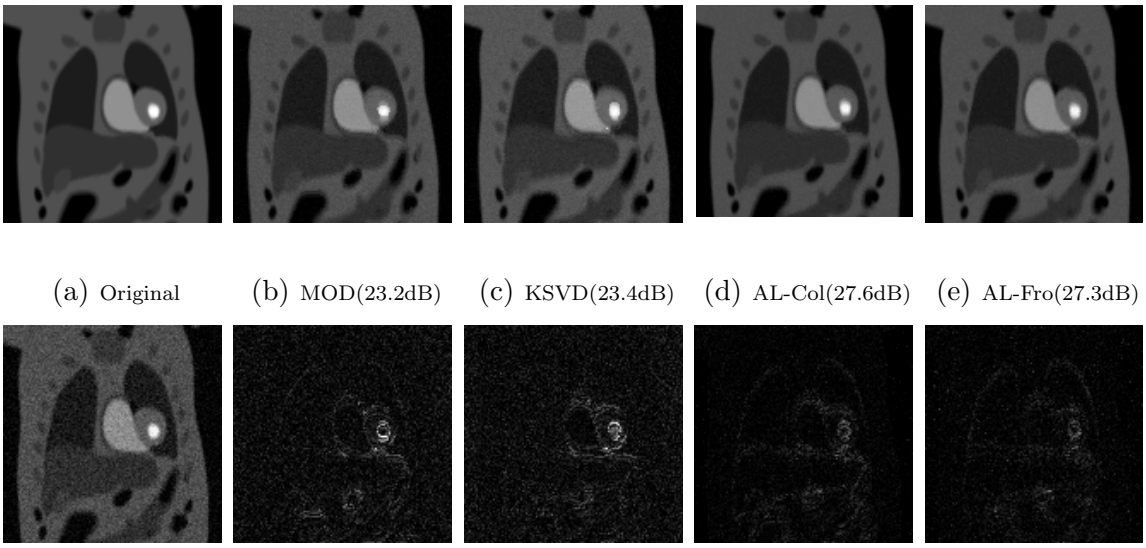
edge detailed information is preserved.

Fig. 4.11 compares the convergence rate for different algorithms. It shows our algorithm converges to a high SNR within 100 Sec for both column-norm constraints and Frobenius-norm constraints. However, K-SVD and MOD need almost 4 times running time, but even provide a worse SNR.

We study the relationship between SNR and dictionary atom number for both column-norm and Frobenius-norm constrains in Fig. 4.12. We choose the optimal λ such that the SNR for any K is optimal. When noise level is low ($\sigma = 10$) in Fig.

4.12 (a), around 15 basis functions are needed to get the best SNR (also shown in Fig. 4.13) and the reconstructions are insensitive to the dictionary atom number beyond 15. Algorithms with both constraints give similar results in this case. However, Fig. 4.12 (b) shows that the algorithm with Frobenius-norm constraints performs much better than the other one under heavy noise ($\sigma = 50$). This is due to modeling with noisy basis functions. Note that the column-norm constraints enforce that all the basis functions are ranked equally (see Fig. 4.13 (a)). In contrast, in the Frobenius-norm constraints, the energy of the learned bases functions can vary considerably, which means that the noisy and insignificant basis functions are enforced to very small values (see Fig. 4.13 (c)).

We also implement the proposed algorithm with joint sparsity constrains on this dynamic dataset. Fig. 4.14 shows the SNR convergence as a function of running time. Compare with the results in Fig. 4.12 (b) at $K = 45$, the algorithm with ℓ_1-ℓ_2 norm constraints presents similar results as Frobenius-norm since they both can get rid of insignificant dictionary atoms.



(a) Original (b) MOD(23.2dB) (c) KSVD(23.4dB) (d) AL-Col(27.6dB) (e) AL-Fro(27.3dB)

(f) Noisy(15.5dB) (g) MOD (h) KSVD (i) AL-Col (j) AL-Fro

Figure 4.10: Comparisons of the proposed scheme with different methods on one of spacial frame when $\sigma = 10$. (a) and (f) are the actual and noisy images, respectively. (b), (c), (d) and (e) show the reconstructions using MOD, K-SVD, DL-AL with column-norm constraints and DL-AL with Frobenius-norm constraints, respectively. (g), (h), (i) and (j) show the corresponding error images. Note that the DL-AL method performs better in denoising.

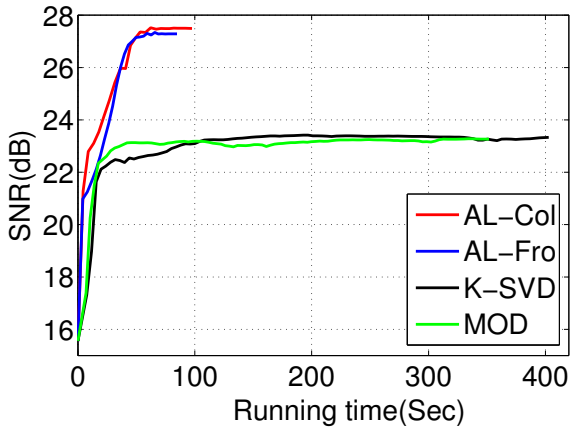
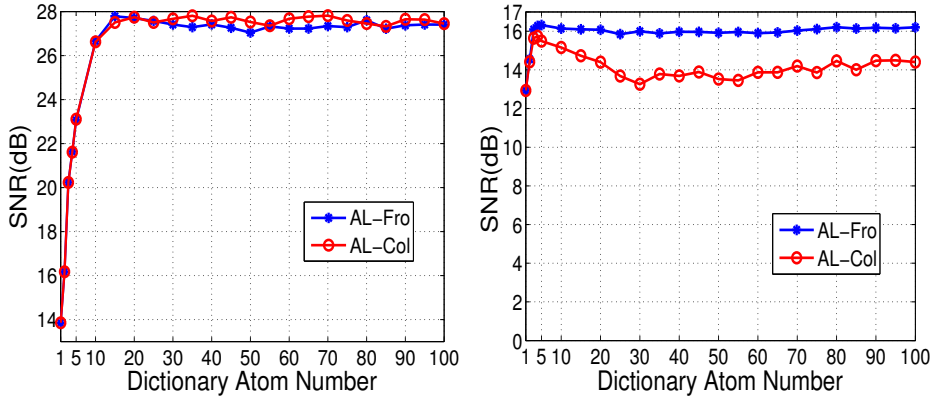
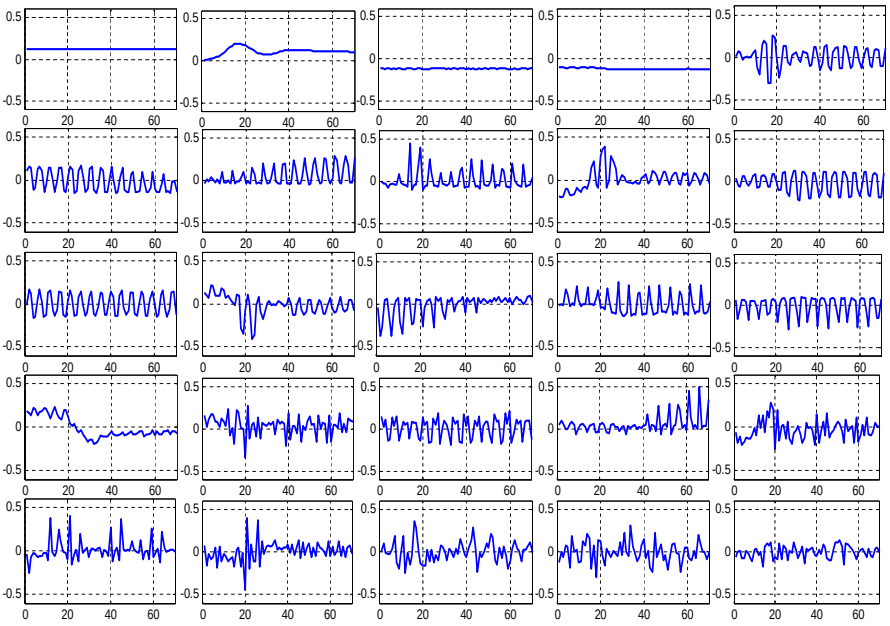


Figure 4.11: Comparisons of Convergence in the case of dictionary atom size $K = 45$. This figure shows SNR as a function of running time. Compared to K-SVD and MOD, our scheme converges much faster.

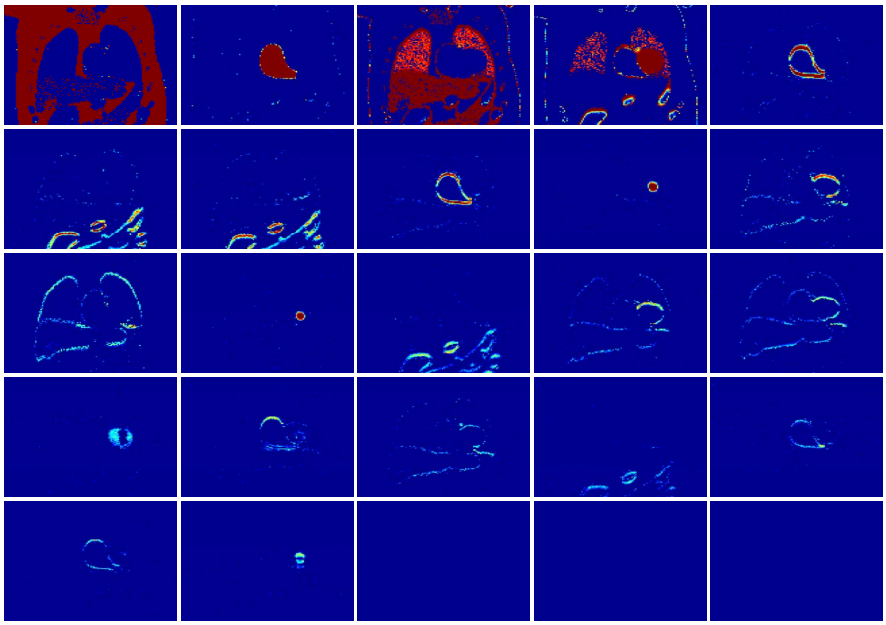


(a) $\sigma = 10$ (b) $\sigma = 50$

Figure 4.12: Evaluation of dictionary atom number K. (a) shows SNR as a function of different dictionary atom sizes for low level noise $\sigma = 10$. (b) shows SNR as a function of different dictionary atom sizes for heavy noise $\sigma = 50$.

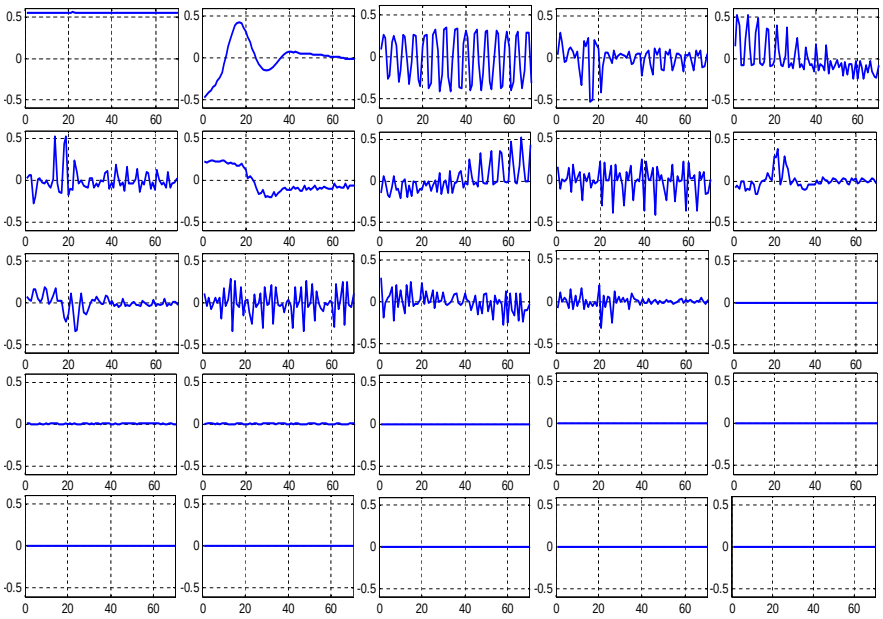


(a) **D**(Col)

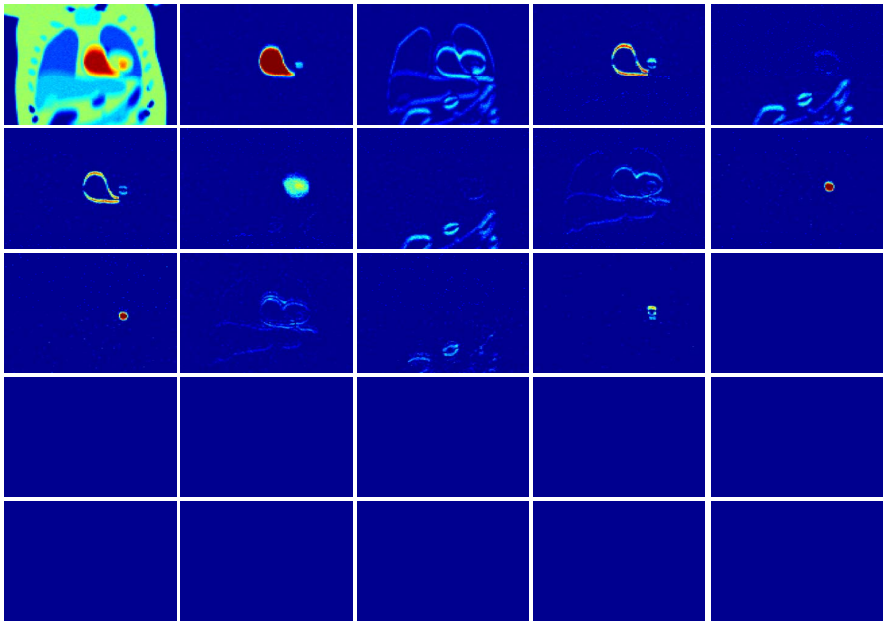


(b) **X**(Col)

Figure 4.13: Dictionary temporal bases and corresponding spatial coefficients ($\sigma = 10$). (a) and (b) show results from column-norm constants case. (c) and (d) show results from Frobenius-norm constants case.



(c) **D**(Fro)



(d) **X**(Fro)

Figure 4.13: Continued

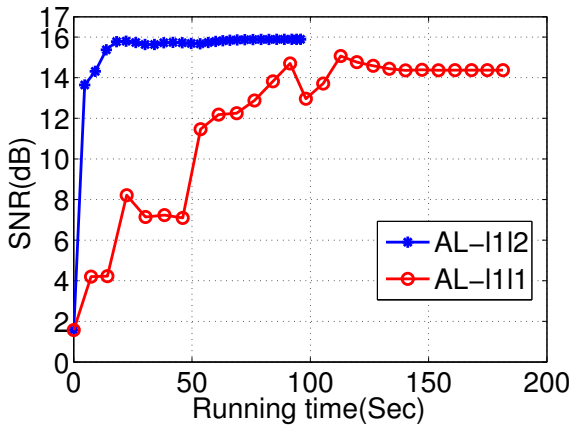


Figure 4.14: Results for joint sparsity constraints when $K = 45$ and $\sigma = 50$.

CHAPTER 5
CONCLUSION

We introduced a novel augmented Lagragian based matrix factorization algorithm for general matrix factorization problems. We focus on applying the proposed scheme on dictionary learning, which is formulated as a sparsity penalized optimization scheme, constrained by different dictionary constraints. And it is easy to extend this algorithm to other matrix factorization problems as shown in this thesis. We used the alternating minimization strategy to decouple the optimization problem into three main sub-problems, each of which has efficient solutions. Comparisons of the proposed scheme with other algorithm showed that the algorithm is capable of recovering the dictionaries at higher sparsity levels. Denoising experiments showed that improved dictionaries translate to improved reconstructions. Numerical experiments also considerably improved computational efficiency.

REFERENCES

[1] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311 –4322, nov. 2006.

[2] M. Angshul and K.W. Rabab. An algorithm for sparse mri reconstruction by schatten p-norm minimization. *Magnetic Resonance Imaging*, 29(3):408 – 417, 2011.

[3] D.L. Donoho and I.M. Johnstone. Ideal denoising in an orthonormal basis chosen from a library of bases. *Comptes Rendus Acad. Sci., Ser. I*, 319:1317–1322, 1994.

[4] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736 –3745, dec. 2006.

[5] K. Engan, S.O. Aase, and J. Hakon Husoy. Method of optimal directions for frame design. In *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, volume 5, pages 2443 –2446, 1999.

[6] E.F. Gonzales and Y. Zhang. Accelerating the lee-seung algorithm for non-negative matrix factorization. *Technical report, Department of Computational and Applied Mathematics, Rice University*, 2004.

[7] S. Haipeng and Z.H. Jianhua. Sparse principal component analysis via regularized low rank matrix approximation. *J. Multivar. Anal.*, 99(6):1015–1034, July 2008.

[8] M.R. Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4:303–320, 1969. 10.1007/BF00927673.

[9] Y. Hu, S.G. Lingala, and M. Jacob. A fast majorize minimize algorithm for the recovery of sparse and low-rank matrices. *IEEE Transactions on Image Processing*, 21(2):742 –753, feb. 2012.

[10] Z. Hui, Trevor H., and Robert T. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15:2006, 2004.

[11] A. Hyvrinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(45):411 – 430, 2000.

[12] A. HyvrinenAapo. Independent component analysis in the presence of gaussian noise by maximizing joint likelihood. *Neurocomputing*, 22(13):49 – 67, 1998.

[13] I.T. Jolliffe. *Principal Component Analysis*. Springer, New York, 2002.

[14] H. Kim and H. Park. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, 23(12):1495–1502, 2007.

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

[16] D.D. Lee and H.S. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562. MIT Press, 2000.

[17] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *NIPS*, pages 801–808. NIPS, 2007.

[18] M.S. Lewicki and B.A. Olshausen. A probabilistic framework for the adaptation and comparison of image codes. *J. Opt. Soc. Am. A*, 16:1587–1601, 1999.

[19] C. Lin. Projected gradient methods for non-negative matrix factorization. Technical report, Neural Computation, 2007.

[20] S. Rayan and Y. Özgür. Sparse recovery by non-convex optimization. *CoRR*, abs/0809.0745, 2008.

[21] X. Wei, L. Xin, and G. Yihong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, pages 267–273, New York, NY, USA, 2003. ACM.

[22] M. Yaghoobi, T. Blumensath, and M.E. Davies. Dictionary learning for sparse approximations with the majorization method. *IEEE Transactions on Signal Processing*, 57(6):2178 –2191, june 2009.