EFFECTIVE SOURCE SEPARATION ALGORITHM USING NMF AND SPARSE NMF FOR ACOUSTIC FALL DETECTION

A Thesis presented to
the Faculty of the Graduate School
at the University of Missouri

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

by
Yanlin Wang
Dr. Dominic Ho, Thesis Supervisor
DEC 2014
The undersigned, appointed by the Dean of the Graduate School, have examined the dissertation entitled:

EFFECTIVE SOURCE SEPARATION ALGORITHM USING NMF AND SPARSE NMF FOR ACOUSTIC FALL DETECTION

presented by Yanlin Wang,
a candidate for the degree of Master of Science and hereby certify that, in their opinion, it is worthy of acceptance.

__________________________________________

Dr. Dominic Ho

__________________________________________

Dr. Guilherme DeSouza

__________________________________________

Dr. Mihail Popescu
ACKNOWLEDGMENTS

This work would not be possible without the help, guidance, and encouragement of my advisor, colleagues, friends, and family. First, I would like to thank Dr Ho as my advisor for his inspiring teaching and encouraging me to find my own research path. Your support and suggestion have guided me through the most difficult moments of my research.

I am thankful to Dr DeSouza and Dr Popescu for being on my dissertation committee. Many thanks to Dr Zare for generously providing me the source code for the project. Without her code and paper, my initial proposal cannot develop it into a true body of work.

Above all, I send my thanks to all my families and friends. Thank you to my friends in Boston, California, and elsewhere. Thank you to my cousin, Zhaoxiang Jin, who shared many of the same experiences of graduate school, albeit years and distance apart. Finally, thank you to my mother and father, Huimin Jin and Liming Wang, to whom this thesis is dedicated.
# TABLE OF CONTENTS

**ACKNOWLEDGMENTS** ........................................... ii

**LIST OF TABLES** ........................................... vi

**LIST OF FIGURES** .......................................... vii

**ABSTRACT** .................................................... x

**CHAPTER**

1 Introduction .................................................. 1

1.1 Background of NMF ......................................... 1

1.2 Challenges for NMF ......................................... 4

1.3 Application ................................................ 5

1.3.1 Text Mining ........................................... 5

1.3.2 Speech processing ....................................... 6

1.3.3 Data clustering .......................................... 6

1.3.4 Bioinformatics .......................................... 6

1.3.5 Hyperspectral Data Analysis ............................ 7

1.3.6 Image Processing ....................................... 7

1.3.7 Fall Detection .......................................... 8

1.4 Previous work on NMF and sparse NMF .................... 8
1.5 Contribution of the Thesis ................................................. 10
1.6 Outline of the Thesis ....................................................... 11

2 NMF .............................................................. 12
  2.1 Basic concept of NMF ................................................... 12
  2.2 Different Algorithms for NMF .......................................... 14
    2.2.1 Multiplicative Update Algorithms ............................... 14
    2.2.2 Gradient Descent Algorithm .................................. 21
    2.2.3 Alternating Least Square Algorithm ........................... 23
  2.3 Stopping Criteria ....................................................... 25
  2.4 Performance Comparison .............................................. 27
  2.5 Summary ............................................................... 32

3 Sparse NMF .......................................................... 33
  3.1 Basic concept of Sparse NMF ......................................... 33
  3.2 Algorithms for Sparse NMF ........................................... 35
    3.2.1 Projected Gradient Descent .................................. 36
    3.2.2 Alternating Nonnegativity-constrained Least Squares ....... 38
    3.2.3 Performance Comparison ....................................... 48
  3.3 Summary ............................................................... 55

4 Classification Techniques ............................................. 57
  4.1 Features for Audio Signal Processing ............................... 57
    4.1.1 Spectral Features ............................................. 58
    4.1.2 Mel-frequency Cepstral coefficients [59] ..................... 60
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>Description of the Dataset</td>
</tr>
<tr>
<td>6.2</td>
<td>10-fold cross validation result for single channel, 2-channel, 3-channel and 4 channel NMF based source separation at different SNR when SIR=5dB</td>
</tr>
<tr>
<td>6.3</td>
<td>10-fold cross validation result for single channel, 2-channel, 3-channel and 4 channel sparse NMF based source separation at different SNR when SIR=5dB</td>
</tr>
</tbody>
</table>
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>NMF with a fall signal mixed with TV interference and white Gaussian noise with K=60. Notice how the matrix $H$ extract the time onsets of the fall signal.</td>
</tr>
<tr>
<td>2.2</td>
<td>NMF interpretation. The matrix $V$ is approximated as a sum of matrix layers.</td>
</tr>
<tr>
<td>2.3</td>
<td>Comparison of the convergence property for Multiplicative Update, Gradient Descent and Inexact Alternating Least Square algorithms.</td>
</tr>
<tr>
<td>2.4</td>
<td>The non-convergence case for Gradient Descent with large step rate.</td>
</tr>
<tr>
<td>3.1</td>
<td>Illustration of various degrees of sparseness. Four vectors are presented with sparseness equal to 0, 0.2, 0.52 and 1. Each bar denotes the value of one element of the vector.</td>
</tr>
<tr>
<td>3.2</td>
<td>Sparse NMF using sparsity promoted ICE of a fall signal mixed with TV interference and white Gaussian noise when K=60 $\mu=0.004$ and $\Gamma=0.01$.</td>
</tr>
<tr>
<td>3.3</td>
<td>Comparison of the convergence property for the Projected Gradient Descent, ANLS and the Sparsity promoted ICE.</td>
</tr>
</tbody>
</table>

vii
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4</td>
<td>The change of the RSS, SSD and SPT terms over the iterations.</td>
<td>51</td>
</tr>
<tr>
<td>3.5</td>
<td>Change of the reconstruction error for different sparse NMF algorithms</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td>using different number of basis.</td>
<td></td>
</tr>
<tr>
<td>3.6</td>
<td>Influence of $\Gamma$ on the number of basis found by sparsity promoted ICE</td>
<td>53</td>
</tr>
<tr>
<td>3.7</td>
<td>Influence of $\Gamma$ on the factorization accuracy</td>
<td>54</td>
</tr>
<tr>
<td>3.8</td>
<td>The actual number of basis found by sparsity promoted ICE when</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>increasing the initial number of basis</td>
<td></td>
</tr>
<tr>
<td>4.1</td>
<td>Steps for computing MFCCs</td>
<td>60</td>
</tr>
<tr>
<td>4.2</td>
<td>Example of kNN classification</td>
<td>64</td>
</tr>
<tr>
<td>5.1</td>
<td>A mixture signal containing fall, TV audio and Gaussian white noise</td>
<td>71</td>
</tr>
<tr>
<td></td>
<td>signal.</td>
<td></td>
</tr>
<tr>
<td>5.2</td>
<td>Plots of some selected normalized row of matrix $H$. (a)$k=17$, (b)$k=31$, (c)$k=1$, (d)$k=41$</td>
<td>72</td>
</tr>
<tr>
<td>5.3</td>
<td>Separation of the fall basis using thresholding</td>
<td>74</td>
</tr>
<tr>
<td>5.4</td>
<td>NMF source separation example</td>
<td>75</td>
</tr>
<tr>
<td>5.5</td>
<td>Sparse NMF source separation example</td>
<td>76</td>
</tr>
<tr>
<td>5.6</td>
<td>The time domain mixture signal and separated fall signal using the</td>
<td>77</td>
</tr>
<tr>
<td></td>
<td>proposed single channel NMF source separation</td>
<td></td>
</tr>
<tr>
<td>5.7</td>
<td>Reconstructed spectrogram for the fall signal and non-fall signal.</td>
<td>81</td>
</tr>
<tr>
<td>5.8</td>
<td>Mean square error for separated fall signal when SIR=$5$dB and $K=60$</td>
<td>82</td>
</tr>
<tr>
<td>6.1</td>
<td>10-fold cross validation ROC curves of fall detection using NMF and</td>
<td>89</td>
</tr>
<tr>
<td></td>
<td>sparse with single channel, 2-channel ,3-channel and 4-channel when</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SIR=$5$dB and SNR='clean'</td>
<td></td>
</tr>
</tbody>
</table>
6.2 10-fold cross validation ROC curves of fall detection using NMF and 
sparse with single channel, 2-channel, 3-channel and 4-channel when 
SIR=5dB and SNR=5dB  

6.3 10-fold cross validation ROC curves of fall detection using NMF and 
sparse with single channel, 2-channel, 3-channel and 4-channel when 
SIR=5dB and SNR=0dB  

6.4 10-fold cross validation ROC curves of fall detection using NMF and 
sparse with single channel, 2-channel, 3-channel and 4-channel when 
SIR=5dB and SNR=-5dB  

6.5 10-fold cross validation ROC curves of fall detection using NMF and 
sparse with single channel, 2-channel, 3-channel and 4-channel when 
SIR=5dB and SNR=-10dB
ABSTRACT

In applications such as audio denoising, it is favorable to decompose a recording into its respective sources. Two commonly used methods are principal component analysis (PCA) and independent component analysis (ICA). However, such approaches perform poorly when no training data is given.

To overcome these issues, we propose another promising and effective algorithm, which is based on non-negative matrix factorization (NMF). The method works by decomposing the time-frequency domain of the signal into separated parts which correspond to respective sources.

The sparsity constraint is introduced to address the number of the bases. The sparse NMF is then proposed to find the appropriate number of bases during the factorization, which yields more convincing and accurate model of the original data.

To verify the advantage of the proposed method, we develop an acoustic fall detection system. For the simulated tasks, we find that sparse NMF based source separation is better than its NMF based counterpart in terms of approximation accuracy and detection performance. Overall, the result demonstrates that our proposed approach is a powerful separation method and motivates further work in source separation.
Chapter 1

Introduction

1.1 Background of NMF

With the rapid developments of the technology, the quantities of data have increased dramatically. Processing these massive amounts of data for representation and dimension reduction has created significant interests in related areas. Most of the time, the received data are not exact due to the corruption caused by the noise and interference or the finite bandwidth of the sensors. Reducing the inexactness through the data reconstruction is an important procedure before any valid processing can be applied.

Besides, different kinds of data require different schemes to capture the underlying features of the data. Implementing the correct model can not only save time but also provide a better representation of the original data. The ideal model may not exist since there is always a better model to be found. For example, dominant
subspace with the Singular Value Decomposition (SVD) [1] is believed to be the best approach to reduce the complexity of data, which produces the least error with the same complexity, compared with other methods. However, it is not the only appropriate algorithm since other classical approaches [2] have been extensively used as well.

In many situations, the collected data can be represented by the integration of several independent variables interacting with one another. If these variables could be found, a dimension reduced system can be implemented to replace the original data through the subspace approximation. PCA and ICA are two of the many popular low-dimension approximation methods to reduce the number of variables and detect the structure among the variables used in a wide range of applications.

Oftentimes the data to be analyzed are non-negative, for example, the pixel intensity of an image or the spectrogram of an audio signal. To avoid the physical contradiction, the low-dimension representation should be nonnegative as well. Classical method like PCA or ICA cannot guarantee the non-negativity.

Besides, for the low-dimension representation, the presence or absence of a basis creates composites for making an object. Whether a basis exists or not is represented by a set of nonnegative numbers. Because of that, nonnegativity is a crucial characteristic which needs to obtain during the analysis. Decomposing nonnegative objects with general methods like the SVD significantly compromises the physical constraint of the data. Other analysis tools like the PCA require some features that nonnegative data hardly possess, such as orthogonality.

This gave us enough motivations to focus on a better algorithm for nonnegative data decomposition, which is nonnegative matrix factorization (NMF). This technique
allows us to approximate nonnegative data, stored in columns of a nonnegative matrix \( V \), by the product of two other nonnegative matrices \( W \) and \( H \):

\[
V = WH
\]  

(1.1)

where \( V \) is \( M \times N \), \( W \) is \( M \times K \) and \( H \) is \( K \times N \). This factorization retains all key features of the original data. Columns of \( W \) define the extracted bases and rows of \( H \) describe the contributions of these bases.

The idea of approximating a nonnegative matrix \( V \) by the product of two nonnegative matrices is not new. In fact, it is a generalized method of the well-known K-Means method applied to nonnegative data. Suppose we have \( n \) nonnegative data vectors \( v_1, ..., v_n \) and \( r \) initial centroids \( c_1, ..., c_r \) representing \( r \) clusters \( C_1, ..., C_r \), what the K-Means method does is to repeat the following two steps until convergence:

1. For each \( v_i \), assign it to \( C_j \) if \( c_j \) is the nearest centroid to \( v_i \).
2. For each \( c_j \), replace it with the mean of all \( v_i \) in \( C_j \).

We can construct a matrix \( W \) by putting all the vectors \( v_j \) in the columns of \( W \) and create a matrix \( H \) such that

\[
H_{i,j} = \begin{cases} 
1 & \text{if } v_j \in C_i \\
0 & \text{otherwise}
\end{cases}
\]  

(1.2)

The K-Means method tries to minimize the Euclidean distance between matrices \( V \) and \( WH \). Mathematically, we solve:

\[
\min \ ||V - WH||_F^2
\]  

(1.3)
To some extent NMF is the same as K-Means method except the values of the matrix $W$ and $H$. Instead of a binary matrix, in NMF, $H$ contains a set of normal nonnegative numbers. The difference offers more flexibility and difficulties to NMF solve the two subproblems optimally, which are

$$\min_{W \in \mathbb{R}^M \times K} \| V - WH \|_F^2$$  \hspace{1cm} (1.4)

$$\min_{H \in \mathbb{R}^N \times K} \| V - WH \|_F^2$$  \hspace{1cm} (1.5)

Obtaining nonnegativity during the analysis of the nonnegative data preserves important properties of the data. The cost of losing mathematical precision because of the nonnegativity constraint is worthwhile since the factorization result can give us meaningful representation of the original data.

### 1.2 Challenges for NMF

The first challenge for NMF is the ambiguity for the decomposition result. Since there is no unique solution for the NMF, a matrix $P$ and its inverse $P^{-1}$ can be used to transform the two factorization matrices

$$V = WH = WPP^{-1}H$$  \hspace{1cm} (1.6)

as long as the transformed matrix $\tilde{W} = WP$ and $\tilde{H} = P^{-1}H$ are both nonnegative matrices. Depending on the problem, the columns of $W$ or rows of $H$ are constrained to unity sum to solve the ambiguity issue.
Another even more important challenge affecting NMF is how to choose the value of K, which is critical in practice and often problem dependent. Given the same factorization accuracy, a lower dimension NMF result provides more compact and effective result. Automatic dimension detection is a difficult task in application. For conventional NMF, the value of K is chosen based on some analysis of the experiment data rather than automatic determination during the factorization process. In this thesis, the spasticity will be imposed on the classical NMF to determine the appropriate dimension automatically, which would be discussed in length in Chapter Three.

1.3 Application

1.3.1 Text Mining

The smallest meaningful units of texts are words. Usually, each document is about a specific topic or category. Topics are characterized by a list of keywords, which describe their semantics. In reality, the topic of a document is often not pure, meaning each document can belong to a number of topics. NMF can be applied to text mining [3], where a term-by-message matrix is constructed by the weighted word-frequency information. This matrix is then factorized into a term-feature matrix, and a feature-message matrix using NMF, where features are extracted from the message, and the feature-message matrix interpolates the data clusters for related messages.
1.3.2 Speech processing

Speech processing has been a lasting topic in audio signal processing area which includes speech separation [4-5], speech denoising [6-8], and robust automatic speech recognition [9-10]. The key idea is that speech signal can be represented by a speech dictionary, but the noise cannot. The general step for speech processing is to extract the clean speech signal which is shown below. Given a noisy speech, the magnitude of the Short-Time Fourier Transform (STFT) is calculated and separated into two parts via NMF. Using the speech parts can help to obtain the estimated clean speech.

1.3.3 Data clustering

Some previous researchers [11] show that NMF is a variant of K-means clustering where the matrix $W$ corresponds to cluster centroids, and matrix $H$ contains membership for different centres. In [12-14], consensus clustering and semi-supervised clustering using NMF have been proven to be simple to implement and the convergence is also guaranteed.

1.3.4 Bioinformatics

Recently, NMF has been successfully applied to bioinformatics [15-16]. With the introduction of microarray technology, many classification methods have been proposed to diagnose and classify cancer types via gene expression patterns. Compared with the traditional methods, which is based on the clinical evidence, this approach is not subject to the pathologists personal experience. In [15] and [16], sparse NMF has been applied to cancer class discovery. The result indicates that sparse NMF
improves cancer detection rate. NMF has also been applied to areas like gene expression data [17-20], microarray comparative genomic hybridization [21] and functional characterization of gene lists [22].

1.3.5 Hyperspectral Data Analysis

Increasing amount of hyperspectral datasets are available and conventional analysis like mixture of Gaussians is not good enough to extract the astonishing volume of information. In [23] the authors develop an effective NMF algorithm for unmixing spectral reflectance data for space object identification and classification, whose result indicates NMF can provide more accurate endmembers and thus enhancing the identification performance.

1.3.6 Image Processing

Each digital image is a rectangular matrix of pixels. Moreover, each pixel is represented by its light intensity. Since the light intensity is measured by a nonnegative value, each image can be represented as a nonnegative matrix, where each element is a pixel. In [27], the authors extract a list of valid features using NMF. The method guarantees not only the nonnegativity of the features but also small reconstruction error. Therefore, together with the coefficients of each feature in the image, the composition for each image is found easily.
1.3.7 Fall Detection

Automatic fall detection has attracted research attention recently since the older lead a longer and more independent lives and falls have become the leading cause of death for the elderly according to a Center for Disease Control report [24]. Due to the low cost of the acoustic sensors and the robustness in occluded environments, acoustic sensors have been used wildly for the fall detection problem. In [25] and [26] authors proposed a NMF based multi-channel fall detection system and a single channel sparse NMF based fall detection system whose results reveal that both NMF and sparse NMF can reduce interference and improve performance of acoustic-based fall detection.

1.4 Previous work on NMF and sparse NMF

First proposed by Paatero and Tappwe [28], NMF used to be called positive matrix factorization. After that Lee and Seung [27] found an interesting property of NMF, which is NMF’s ability to decompose the objects in meaningful parts when they applied NMF to image processing. It is the paper that popularizes the NMF. Since then efforts have been devoted to the analysis and application of NMF algorithms in various disciplines.

Many researchers have studied different form of the cost function to evaluate the performance difference. Lee and Seung [29] provide a formulation based on the Kullback-Leibler (KL) divergence, which leads to some variant approaches, such as Cichocki’s $\phi$-divergence based cost function [30]. Wang et al. [31] propose a framework based on Fisher linear discriminant analysis to improve the determination of spatially
localized features.

Various minimization strategies for the solution of the cost function have also been proposed to speed up convergence of the conventional NMF algorithm [27]. [32] has recently proposed the projected gradient bound-constrained optimization method that is less time-consuming and has better convergence property than the standard multiplicative update approach. [33] proposes accelerating the standard approach based on an interior-point gradient method. [34] proposes a quasi-Newton optimization approach with the projection during each iteration.

There are many studies related to convergence of the standard NMF algorithm. In [27], the authors claim that their algorithm can converge to a local minimum using the gradient and properties of continual non-increase, which is disproved by [32-33, 35] because they forget to preclude the possibility that the solution descends to a saddle point. Most of the NMF algorithms update the $\mathbf{W}$ and $\mathbf{H}$ using an alternating scheme which is a variant of the optimization technique known as alternating variable, coordinate search or method of local variation. Although the global convergence for the general case has not been proven, some special cost functions are proven to be convergent, such as convex quadratic function [36-37]. Extra attention should be paid to balance the tradeoff between the convergence property and the computational complexity. Some of the algorithms, such as the nonnegative constrained least squares, guarantee the convergence to a local minimum but greatly increases the cost per iteration. In practice, researchers often choose the speed and sacrificing the convergence property. Nevertheless, this tradeoff seems worthwhile. Some experiments show that the saddle point can also provide reasonable results [38-40].

In the conventional NMF algorithm, $\mathbf{W}$ and $\mathbf{H}$ are initialized with random non-
negative values. Various efforts have been focused on methods to initialize the algorithm in order to speed up convergence performance. For example, [41-42] employs a spherical k-means clustering approach to initialize $W$. [43] uses an SVD-based initialization. Effective initialization remains to be an open problem deserving further attention.

Many of the previous researches on sparse NMF focus on the application, such as hyperspectral unmixing [44], music instrument processing [45-46], bioinformatics [15-16] and face feature extraction [47]. Sparse NMF provides better result than conventional methods, which can be contributed to the sparsity imposed to $W$ or $H$ that offers a more faithful representation of the original data.

1.5 Contribution of the Thesis

The main topic of this thesis is the investigation of NMF, sparse NMF and its application to source separation for the fall detection. The specific contributions are:

- A new sparse NMF based source separation algorithm that approximates the building blocks of the original data.
- A multichannel based source separation that reduces the noise to accomplish better factorization accuracy.
- The application of the sparse NMF for human fall detection with strong interferences for achieving better results.
1.6 Outline of the Thesis

The thesis is organized as follows. Chapter Two and chapter Three introduce the basic concept, methodology and algorithms for both NMF and sparse NMF and compare the pros and cons for these different algorithms. Chapter Four presents the classification techniques which includes the features and classifier used in the project. Chapter Five describes the source separation using both NMF and sparse NMF for single channel processing and multichannel processing. Chapter Six provides a detailed description of the fall detection application including the parameter setting, experiment procedures and the final result. A brief summary and conclusion is listed in Chapter Seven along with some suggested future work.
Chapter 2

NMF

2.1 Basic concept of NMF

NMF is first introduced by Paatero and Tapper in [28]. But it is the works of Lee and Seung [27] that makes it popular. They argue that the nonnegativity is crucial in human perception and give examples to learn parts of faces and semantic features of text using NMF. Given an M by N nonnegative matrix $V$ and a reduced dimension $K$, the NMF problem is to find two nonnegative matrices $W$ and $H$ that approximate $V$, ie $V = WH$. Typically, $K \ll \min(M, N)$, yields a compressed and low-dimension approximation of the data $V$. Furthermore, we can say the following:

- $V$: original non-negative input data matrix.
  - Each column is an M dimension data sample.
  - Each row represents a data feature.

- $W$: matrix of basis vectors, basis functions, or dictionary elements.
Each column represents a basis vector.
Each column is not necessarily orthonormal, but commonly normalized to one.

- $H$: matrix of contributions, coefficients or weights.
Each row represents the contribution of a corresponding basis vector.
Each row is not necessarily orthonormal, but sometimes normalized to one.

The columns of $V$ can be approximated by a linear combination of $K$ nonnegative basis vectors which is columns of $W$

$$V(:,i) = \sum_{j=1}^{K} W(:,j) H_{j,i} = WH(:,i)$$  \hspace{1cm} (2.1)

There are several ways to measure the difference between the original data $V$ and the approximation result $WH$. One of the most widely used measurements is the Frobenius norm:

$$F(V, WH) = \frac{1}{2} ||V - WH||_F^2 = \frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{N} (V_{i,j} - [WH]_{i,j})^2$$  \hspace{1cm} (2.2)

which is also referred to as the Euclidean Distance. Another well-known method is the KL divergence.

$$D(V, WH) = Vln\left(\frac{V}{WH}\right) + WH - V = \sum_{i=1}^{M} \sum_{j=1}^{N} V_{i,j}ln\left(\frac{V_{i,j}}{[WH]_{i,j}}\right) + [WH]_{i,j} - V_{i,j}$$  \hspace{1cm} (2.3)

Different cost functions assign different penalization. There are a large varieties of divergence measures which have been successfully applied to NMF problems, such as Bregman divergences [48], Renyi’s information measure [49], Csizsar’s divergences
Kompass’ divergence [50], the $\alpha$-divergence [51] or the Itakura-Saito divergence [52] which will not be discussed here.

2.2 Different Algorithms for NMF

There are three main algorithms for the NMF: multiplicative update algorithm, the alternating least square algorithm and the gradient descent algorithm. The earliest algorithm is the alternating least square algorithm proposed by Paatero [28] for what they called positive matrix factorization, which can be view as the predecessor of the NMF. However, the focus has shifted since the introduction of the multiplicative update of Lee and Seung [27]. The problem was then rephrased as NMF. The multiplicative update algorithm is simple to use and offers good interpretation of the original data. Gradient descent method is one of the most common and straightforward methods for the optimization problem, but typically are slow to converge.

2.2.1 Multiplicative Update Algorithms

Updating Equation

The most popular algorithm for NMF is the multiplicative update rule suggested by Lee and Seung [27]. The method works by decomposing the optimization problem into two subproblems, fixing one of the matrices $W$ or $H$ to minimize the cost function with respect to the other and repeating this process to obtain an acceptable solution. To formulate the updating equation, we choose the Frobenius norm cost function to
illustrate how this algorithm works. The cost function can be expressed as:

$$\frac{1}{2}||V - WH||_F^2 = \frac{1}{2} \sum_{i=1}^{N} ||V_{:,i} - WH_{:,i}||_2^2$$

(2.4)

We can minimize the cost function with respect to each column of H individually, which generates a series of quadratic problems:

$$\min_{H \in \mathbb{R}_+} F(H) \quad \text{where} \quad F(H) = \frac{1}{2}||V - WH||_2^2$$

(2.5)

Consider an approximation $\bar{H} > 0$ of the solution and formulate the following problem:

$$\min_{H \in \mathbb{R}_+} \tilde{F}(H) \quad \text{where} \quad \tilde{F}(H) = \frac{1}{2} [||V - WH||_2^2 + (H - \bar{H})^T U_H (H - \bar{H})]$$

(2.6)

where $U_H = D_x - W^TW$, $D_x$ stands for a diagonal matrix with the vector $x$ on its main diagonal and $x$ is equal to $\frac{w^TWH}{H}$. It can be proven that $U_H$ is positive semidefinite, therefore $\tilde{F}(H) \geq F(H)$ for all $H$. Moreover, the function is also convex.

We set the derivative of $\tilde{F}$ equal to zero, i.e

$$\frac{\partial \tilde{F}(H)}{\partial H} = W^T WH - W^T V + U_H (H - \bar{H}) = 0$$

(2.7)

to obtain the optimized value $H^*$ for $H$:

$$(W^TW + U_H)H^* = W^TV + U_H\bar{H} = 0$$

(2.8)
Since $W^TW + U_H = D_W^TWHD_H^{-1}$ and $U_HH = 0$, we get

$$H^* = H\frac{W^TV}{W^TWH} \quad (2.9)$$

Since $H^*$ is the global minimizer of $\tilde{F}(H)$, we have $\tilde{F}(H^*) \leq \tilde{F}(H)$. Moreover, $\tilde{F}(H)$ is constructed to satisfy $\tilde{F}(H) \geq F(H)$ for all $H$. This implies that we have a descent on the cost function. The updating scheme for $W$ can be derived using the same manner. The updates provide two alternating steps in Algorithm 1.

---

**Algorithm 1** MU-KL Divergence Update Procedure

Require:
- $V \in \mathbb{R}^{M \times N}$, $W \in \mathbb{R}^{M \times K}$, $H \in \mathbb{R}^{K \times N}$, where $K$ is the number of basis.

1: loop
2:
   $$W_{i,j} \leftarrow \frac{W_{i,j}}{\sum_{n=1}^{N} H_{j,n} \sum_{n=1}^{N} \frac{V_{i,n}WH_{j,n}}{|WH|_{i,n}}} \quad (2.10)$$
3:
   $$H_{i,j} \leftarrow \frac{H_{i,j}}{\sum_{m=1}^{M} W_{m,i} \sum_{m=1}^{M} \frac{V_{m,j}WH_{m,i}}{|WH|_{m,j}}} \quad (2.11)$$
4: end loop

---

**Algorithm 2** MU-Frobenius Norm Update Procedure

Require:
- $V \in \mathbb{R}^{M \times N}$, $W \in \mathbb{R}^{M \times K}$, $H \in \mathbb{R}^{K \times N}$, where $K$ is the number of basis.

1: loop
2:
   $$W_{i,j} \leftarrow W_{i,j} \frac{[VH^T]_{i,j}}{[WHH^T]_{i,j}} \quad (2.12)$$
3:
   $$H_{i,j} \leftarrow H_{i,j} \frac{[WTV]_{i,j}}{[W^TWH]_{i,j}} \quad (2.13)$$
4: end loop
Convergence Properties

Lee and Seung used the claim that the multiplicative update algorithm converges to a local minimum, which was later shown to be incorrect. There are two main problems to investigate the convergence property of the multiplicative update algorithm. The first one is that these multiplicative rules fail to descend the cost function. Rewrite the equation as:

\[ H' = H \frac{W^TV}{W^TWH} \]  (2.14a)
\[ = H \frac{W^TV + W^TWH - W^TWH}{W^TWH} \]  (2.14b)
\[ = H(1 + \frac{W^TV - W^TWH}{W^TWH}) \]  (2.14c)
\[ = H + \frac{H}{W^TWH}[W^T(V - WH)] \]  (2.14d)

With this updating rule, a necessary condition for a sufficient descent is that the eigenvalues of matrix \( \frac{H}{W^TWH} \) must be away from zero, which is not true in general. The limit point of the algorithm might not be stationary.

Another obstacle is the appearance of zeros in \( W \) and \( H \). If the initial set \((W, H)\) converge to \((W^*, H^*)\), then \( \frac{\partial F}{\partial W}(W^*, H^*) = 0 \) and \( \frac{\partial F}{\partial H}(W^*, H^*) = 0 \). These can be proven by using the additive updating rules. Suppose a limit point for \( H \) has reached. From the Eq.2.14(d), we know for the \((i, j)^{th}\) element of \( H \)

\[ \frac{H_{i,j}}{[W^TWH]_{i,j}}([W^TV]_{i,j} - [W^TWH_{i,j}]) = 0 \]  (2.15)

Since \( H_{i,j} > 0 \), which implies \([W^TV]_{i,j} = [W^TWH_{i,j}]\), and thus we can get
When we combine the equations together

\[ \frac{\partial F}{\partial W_{i,j}} = 0. \]

the Karush-Kuhn-Tucker optimality conditions is satisfied.

Actually we can rewrite the NMF as a standard nonlinear optimization problem:

\[
\min_{W \in \mathbb{R}_-, H \in \mathbb{R}_-} 2 \frac{1}{2} \| V - WH \|_F^2
\]

The Lagrangian function is

\[
L(W, H, \mu, \nu) = \frac{1}{2} \| V - WH \|_F^2 - \mu \ast W - \nu \ast H
\]

where \( \mu \) and \( \nu \) are two matrices containing the Lagrange multipliers constraints \( W > 0 \) and \( H > 0 \). According to the famous KKT conditions, if \((W, H)\) is a local minimum,
then there exist $\mu_{i,j} \geq 0$ and $\nu_{i,j} \geq 0$ such that:

\begin{align*}
\mathbf{W} &\geq 0, & (2.19a) \\
\mathbf{H} &\geq 0, & (2.19b) \\
\frac{\partial L}{\partial \mathbf{W}} &= 0, & (2.19c) \\
\frac{\partial L}{\partial \mathbf{H}} &= 0, & (2.19d) \\
\mu \mathbf{W} &= 0, & (2.19e) \\
\nu \mathbf{H} &= 0. & (2.19f)
\end{align*}

Expanding the Eq.2.19(c) and Eq.2.19(d), we have

\begin{align*}
\mu &= \mathbf{VH}^T + \mathbf{WHH}^T, & (2.20a) \\
\nu &= \mathbf{V}^T \mathbf{W} - \mathbf{H}^T \mathbf{W}^T \mathbf{W} & (2.20b)
\end{align*}

Combining all these equations together, we have

\begin{align*}
\mathbf{W} &\geq 0, & (2.21a) \\
\mathbf{H} &\geq 0, & (2.21b) \\
(\mathbf{WH} - \mathbf{V})\mathbf{H}^T &\geq 0, & (2.21c) \\
\mathbf{W}^T(\mathbf{WH} - \mathbf{V}) &\geq 0, & (2.21d) \\
(\mathbf{WH} - \mathbf{V})\mathbf{H}^T \times \mathbf{W} &= 0, & (2.21e) \\
\mathbf{W}^T(\mathbf{WH} - \mathbf{V}) \times \mathbf{H} &= 0. & (2.21f)
\end{align*}
We want to avoid the situation like
\[
\frac{\partial F}{\partial H_{i,j}} < 0 \text{ while } H_{i,j} = 0 \tag{2.22}
\]

Since the initial value of \( \mathbf{W} \) and \( \mathbf{H} \) are positive and with the updating rule
\[
H_{i,j} \frac{|W^T V|_{i,j}}{|W^T W H|_{i,j}} > 0
\]

It seems that the problem will not happen. However, in practice, because of the finite precision, the truncation error might force some small numbers to be zeros. In this case, the multiplicitive update rule can no longer make the value of \( \mathbf{W} \) or \( \mathbf{H} \) positive. Then the algorithm would trap into a potential non-stationary point.

In short, the multiplicitive update algorithm might give us an non-stationary point. And even we can get a stationary point, it is still difficult to determine whether it is local minima or not. Hence, the convergence property of this algorithm remains to be an open question.

**Computational Complexity**

Various studies have shown that the multiplicitive update algorithm is extremely slow to converge. Each iteration requires \( 6O(N^3) \) matrix-matrix multiplications and \( 6O(N^2) \) component-wise operations. In order to overcome the shortcomings, researchers have proposed some variants of the original algorithm. Some of the algorithms is faster while still has the same convergence issues like [33]. Other algorithm is guaranteed to converge to the stationary point with heavier computational load [53].
2.2.2 Gradient Descent Algorithm

Known as steepest descent or the steepest descent, gradient descent is a first-order optimization algorithm. Gradient descent is based on the observation that if the function $F(x)$ is differentiable in a neighborhood of a point $a$, the direction of the negative gradient of $F$, which is $-\nabla F(a)$ is the fastest to decrease. Given a positive value $\gamma$, if $b = a - \gamma \nabla F(a)$, then $F(a) \geq F(b)$. If we start with a random guess of $x_0$ for the local minima of function $F$ and construct a sequence $x_0, x_1, x_2, \ldots$, such that

$$x_{n+1} = x_n - \gamma_n \nabla F(x_n)$$

We have

$$F(x_0) \geq F(x_1) \geq F(x_2) \cdots \geq F(x_n)$$

If the function $F$ is convex and with a particular choice of $\gamma$, the convergence to a local minimum can be proven. Also note that if a function is convex, all its local minima are global minima as well. In this case, the gradient descent algorithm yields the global solution.

The cost function of NMF is not convex with respect to both $W$ and $H$, but it is convex if either of the parameters to be estimated is fixed. Then we can consider NMF as a linear optimization problem on a convex nonnegative orthant. In this case, we can implement the gradient descent framework to solve the minimization problem. The basic idea of applying gradient descent to NMF is to carry out a three-step calculation in each iteration:

- Calculating the gradient $\nabla F(x^{(k)})$
• Choosing the step size $\gamma^{(k)}$ for the $k^{th}$ iteration.

• Projecting the update result on the nonnegative set:

$$x^{(k+1)} = [x^{(k)} - \gamma^{(k)} \nabla F(x^{(k)})]_+$$

The algorithms of gradient descent is shown below:

**Algorithm 3 Gradient Descent Update Procedure**

**Require:**

- $V \in \mathbb{R}_+^{M \times N}$, $W \in \mathbb{R}_+^{M \times K}$, $H \in \mathbb{R}_+^{K \times N}$, where $K$ is the number of basis.

**loop**

$$W \leftarrow W - \alpha \frac{\partial F}{\partial W} = W - \alpha (WH - V)H^T$$ (2.23)

$$H \leftarrow H - \alpha \frac{\partial F}{\partial H} = H - \alpha W^T(WH - V)$$ (2.24)

**end loop**

How to choose the step size $\alpha$ is a tricky question since the rate of convergence depends on step size. Large step size means faster convergence to the neighborhood of the local minima, but the result of the equation could oscillate around the local minima. Small step size is much stable, but it takes much more iteration to converge. Some previous algorithms just set the initial step size $\alpha$ to 1 and reduce it by half for each iteration. Another approach is combining the line search to find the locally optimized step size $\alpha$, which is an extremely time-consuming method. Another problem concerning this algorithm is that there is no guarantee to keep the elements of $W$ and $H$ from being negative. The simplest way to address this issue is to project the updated matrices to a nonnegative orthant by forcing all the negative values in $W$ and $H$ to zero.
However, the choice of step size combined with the projection makes it difficult to analyze the convergence property of the gradient descent algorithm. Since most of the implementation of the gradient descent for NMF use the geometric rule for step size, such as scaling by a constant at each iteration, the result of these algorithms is not far away from the initial matrices with the random initialization. Therefore, they are sensitive to the initialization of $W$ and $H$.

2.2.3 Alternating Least Square Algorithm

The first algorithm proposed to solve the NMF was the alternating least squares method [28]. When either $W$ or $H$ is fixed, the problem becomes a least squares problem with nonnegativity constraint. In this algorithm, a least squares procedure is followed by another least squares procedure in an alternating manner. Given one matrix, the other matrix can be found using a least square computation. The ALS algorithm goes like this:

**Algorithm 4** Exact Alternating Least Square

**Require:**
$V \in \mathbb{R}^{M \times N}$, $W \in \mathbb{R}^{M \times K}$, $H \in \mathbb{R}^{K \times N}$, where $K$ is the number of basis.

**loop**

\[
W \leftarrow \min_{W \in \mathbb{R}^+} \frac{1}{2} ||V - WH||_F^2. \tag{2.25}
\]

\[
H \leftarrow \min_{H \in \mathbb{R}^+} \frac{1}{2} ||V - WH||_F^2. \tag{2.26}
\]

**end loop**

Although the solution of the algorithm can provide a stationary point of the NMF, the speed of the algorithm is far from satisfaction. Even with the fastest
implementation, it cannot match any other algorithms. In this context, a modification has been proposed to replace the exact solution of this nonnegative least square problem. The result generated by an inexact solution is projected on the nonnegative orthant to speed up the algorithm at the cost of the convergence. The algorithm is shown below:

**Algorithm 5 Inexact Alternating Least Square**

**Require:**
- $V \in \mathbb{R}^{M \times N}_+$,
- $W \in \mathbb{R}^{M \times K}_+$,
- $H \in \mathbb{R}^{K \times N}_+$, where $K$ is the number of basis.

**loop**
- Solve $HH^T W^T = HV^T$.
- $W = [W]_+$.
- Solve $W^T WH = W^T V$.
- $H = [H]_+$.

**end loop**

As we have discussed in the previous section, the simplest method for ensuring the nonnegativity is used in this algorithm, which is to set all the negative element in $W$ and $H$ to be zero. Since we can set some of the elements in $W$ and $H$ manually to 0, this algorithm can promote the sparsity of the result. However, some disadvantages still exist. Once an element in $W$ or $H$ becomes zero, its value must remain to be zero, meaning that once the algorithm heads towards a poorly fixed point, it would never move out. The general convergence characteristic has not been proven for the alternating updating scheme. However, some research has studied the convergence for some particular classes of the objective function such as convex quadric functions. Furthermore, it has been shown that alternating least square can converge to a local minimum with the nonnegative constraint using the nonnegative least square algorithm. The greatest challenge for the nonnegative least square is that solving the least square problem with the nonnegative constraint would dramatically increase the
computation by some orders. In fact, most of the researchers choose the speed using a simple projection enforcing the nonnegativity rather than the convergence property.

2.3 Stopping Criteria

For these methods, several stopping conditions are implemented in the literature. The first criterion is the decrease of the objective function. The algorithm would stop when the objective function fails to descend over the iterations at a certain threshold:

$$F(W^{(k)}, H^{(k)}) - F(W^{(k-1)}, H^{(k-1)}) < \epsilon$$

or

$$\frac{F(W^{(k-1)}, H^{(k-1)}) - F(W^{(k)}, H^{(k)})}{F(W^{(k-1)}, H^{(k-1)})} < \epsilon$$

Because the algorithm might stop at a point very far from a stationary point, it is not an ideal option for the general cases. Another stopping criterion is the maximum number of iterations. However, a fixed number of iterations is not an attracting way since the appropriate value for the maximum number of iterations is problem-dependent. Both of these methods do not reveal whether the solution is close to a stationary point or not.

One of the first paper mentioning the stopping criteria for NMF is [32] where the author use the norm of the projected gradient as the stopping criteria. The condition to check if a point $x^{(k)}$ is close to a stationary point in bounded constrained
optimization is the following:

$$||\nabla^P f(x^{(k)})||_F \leq \epsilon ||\nabla f(x^{(0)})||_F$$

(2.27)

where $\nabla^P f(x^{(k)})$ is the project gradient define as

$$\nabla^P f(x^{(k)}) = \begin{cases} 
\nabla f(x^{(k)}) & \text{if } x > 0, \\
\min(0, \nabla f(x^{(k)})) & \text{if } x = 0,
\end{cases}$$

(2.28)

This is equivalent to the KKT condition for the bounded problems, which $||\nabla^P f(x)|| = 0$. For NMF, the stopping criteria becomes:

$$||\nabla^P f(W^{(k)}, H^{(k)})||_F \leq \epsilon ||\nabla f(W^{(0)}, H^{(0)})||_F$$

(2.29)

For alternating least squares, each sub-problem requires a stopping criterion. The following stopping conditions are implemented as the stopping criteria for the sub-problems. The updating algorithm would terminate respectively if these conditions are satisfied:

$$||\nabla^P_W f(W^{(k)}, H^{(k)})||_F \leq \bar{\epsilon}_W$$

(2.30a)

$$||\nabla^P_H f(W^{(k)}, H^{(k)})||_F \leq \bar{\epsilon}_H$$

(2.30b)

where $\bar{\epsilon}_W = \bar{\epsilon}_H = \max(10^{-3}, \epsilon)||\nabla f(W^{(0)}, H^{(0)})||_F$, which is decided by some numerical evaluations.

If the projected gradient stops at the beginning of the iteration, decrease the stopping threshold can yield a better result. Some numerical experiments reveal that
A factor of 10 is an appropriate option for the general cases, which means:

$$
\tilde{\epsilon}_W \leftarrow \frac{\epsilon_W}{10}
$$

$$
\tilde{\epsilon}_H \leftarrow \frac{\epsilon_H}{10}
$$

Also, we need to notice that the gradient $\nabla_W f$ and $\nabla_H f$ are scale-dependent in NMF, and any gradient-based stopping criteria would be affected by the scaling. To alleviate the effect, the scaling is imposed after each iteration that ensures the same Frobenius norm for both $W$ and $H$ to avoid the numerical unstable. The scaling follows these procedures:

$$
\tilde{W} \leftarrow WT
$$

$$
\tilde{H} \leftarrow HT^{-1}
$$

where $T$ is a positive diagonal matrix whose diagonal elements equal to $T_{ii} = \sqrt{\frac{||H_{i,\cdot}||_2}{||W_{\cdot,i}||_2}}$

### 2.4 Performance Comparison

In this example, NMF is used to represent the spectrogram or the magnitude of short Time Fourier Transform (STFT) of the signal when it is applied to audio signal processing. Transform the original signal into the time-frequency domain using the STFT, take its magnitude or power as the matrix $V$, and then factorize the result as $V = WH$. In such manner, NMF approximates spectrogram as a linear combination of several frequencies basis over time.
This process can be seen in Fig. 2.1, where the spectrogram of a mixed signal is shown with the factorization result. The mixed signal is composed of a 10 second long TV audio signal, a fall signal performed by some stunt actors and the Gaussian white noise to simulate the noise. The number of the basis K is typically chosen manually and (M,N) are the function of the overall signal length and STFT parameters such as the frame size and the overlap rate for the neighboring frames. As is shown in Eq. 2.31,

\[ V = \sum_{k=1}^{K} W_{:,k} H_{k,:} \]  

(2.31)

the columns of \( V \) are represented as the weighted sum of basis vectors. The Fig. 2.2 provides a visual representation of this theory.

Figure 2.1: NMF with of a fall signal mixed with TV interference and white Gaussian noise with K=60. Notice how the matrix \( H \) extract the time onsets of the fall signal.
Figure 2.2: NMF interpretation. The matrix $V$ is approximated as a sum of matrix layers.

Fig.2.3 is a typical example of the convergence performance of the Multiplicative Update, the Gradient Descent and the Inexact Alternating Least Squares.
The number of basis $K$ is set to 60 and reconstruction error is defined as $||V - WH||^2_F$. This figure reveals some interesting properties of these three algorithms.

First, the multiplicative update algorithm yields the best result in the experiment. It just takes about 100 iterations to convergence and also provides a relative small reconstruction error.

Second, for the gradient descent, the convergence rate is much slower. This is because the choice of step rate or learning rate is a challenging task for the NMF using gradient descent. The speed of convergence depends on two factors: the step
rate $\alpha$ and the gradient value at the current point. Recall that the updating equation for $W$ is

$$W \leftarrow W - \alpha(WH - V)H^T.$$ 

The initial elements of $W$ and $H$ are both random numbers generated by the Matlab function `rand()`, which are small numbers. With large $\alpha$, the projection step would enforce all the elements in $W$ or $H$ to zero in an alternating manner and repeat this process until some stopping criteria reached. Therefore, as is shown in the Fig.2.4, the cost function will never converge to a stationary point.

![Figure 2.4: The non-convergence case for Gradient Descent with large step rate](image-url)
However, if we choose a small $\alpha$, then it will take much more iteration to converge compared with other algorithms.

Third, the Inexact ALS algorithm does not always make a descent update although it is much more computational effective than the exact ALS, which verifies the result in [51].

2.5 Summary

In this chapter, NMF is introduced as a technique to decompose the non-negative matrix into the product of two matrices. The approximated basis matrix $W$ and the corresponding contribution matrix $H$ are both constrained to be nonnegative. Three basic algorithms for NMF, multiplicative update, the alternating least squares and the gradient descent, are also discussed in this chapter. The multiplicative update algorithm is the most popular algorithm among three of them and balance the factorization accuracy and running time. The exact version of the alternating least squares is guaranteed to converge to a stationary point, its computational complexity makes it difficult to be applied in practice. However the inexact version of the the alternating least squares cannot make a descent update even it is more computational effective.
Chapter 3

Sparse NMF

3.1 Basic concept of Sparse NMF

Since the sparse coding has been shown to be a useful method to reveal the underlying features of the original data, we borrow the idea of sparsity to improve the performance of NMF. The concept of sparse coding refers to a generative model where only a handful of unit is used to represent the data. In fact, this representation means most of the units are close to zero while only a few take non-zero values.

Depending on the application, the NMF can produce a sparse representation of the data. Such a model encodes much of the data using a few active components, which makes the encoding easy to illustrate. However, most of the sparsity brought by the NMF algorithm can be viewed as a by-product, and the spareness of the components cannot be controlled explicitly. Moreover, there exist data sets where NMF doesn’t factorize the original data into the bases corresponding to the building blocks of the
data. For example, in [54] NMF yields global feature rather than local features of the facial images. In most of the applications, more direct control of the sparsity is needed to obtain better performance.

There are many approaches to measure the sparseness of a vector. One of the commonly used methods based on the relationship between the $L_1$ norm and the $L_2$ norm is described here [55]:

$$sparseness(x) = \frac{\sqrt{n} - \frac{\sum_{i=1}^{n} |x_i|}{\sqrt{\sum_{i=1}^{n} x_i^2}}}{\sqrt{n} - 1},$$

where $n$ is the dimension of the vector $x$. This measure maps from $\mathbb{R}^n$ to $\mathbb{R}$, which quantify the energy compactness of a vector. Using this expression, the sparsest vector, where only a single element is non-zero, should have a sparseness of one, whereas a vector with equivalent elements should have a sparseness of zero. An illustration of various sparseness is provided in Fig.3.1.

![Figure 3.1: Illustration of various degrees of sparseness. Four vectors are presented with sparseness equal to 0, 0.2, 0.52 and 1. Each bar denotes the value of one element of the vector.](image)

Our purpose is to constrain NMF to find solutions with desired levels of sparseness. Imposing the sparseness on the matrix $W$ or $H$ is crucial to the performance of the sparse NMF. There is no general answer to the question, meaning that it all depends
on the particular application. Since usually the column vector of $V$ is regarded as data sample, the column of $W$ is considered as the basis function or dictionary list and the row of $H$ is treated as the contribution or weights of the corresponding bases. Most of the literature add the sparse constraint to the matrix $H$.

Given the sparseness measurement, our job is to find solutions of the NMF with the accepted degrees of sparseness, which is given the nonnegative matrix $V \in \mathbb{R}^{M \times N}$ and a reduced dimension $K$, find the nonnegative matrix $W \in \mathbb{R}^{M \times K}, H \in \mathbb{R}^{K \times N}$ which minimize the cost function

$$\frac{1}{2} \| V - WH \|_F^2$$

with the constraint

$$\text{sparseness}(W_{:,i}) = S_W$$

$$\text{sparseness}(H_{i,:}) = S_H,$$

where $S_W$ and $S_H$ denote the desired sparseness of $W$ and $H$. Mathematically, we have

$$(W, H) = \arg \min_{W \in \mathbb{R}_+^{M \times K}, H \in \mathbb{R}_+^{K \times N}} \frac{1}{2} \| V - WH \|_F^2$$

$$\text{s.t} \quad \text{sparseness}(W_{:,i}) = S_W \quad \text{sparseness}(H_{i,:}) = S_H,$$

(3.1)

### 3.2 Algorithms for Sparse NMF

There are several different types of algorithms for sparse NMF: the gradient descend based method, alternating least square based algorithms and the iterated constrained
endmember (ICE) based approach. The first two algorithms are the extension of the conventional NMF algorithms, which add the penalty term to the objective function to achieve appropriate sparseness. The third algorithm, however, is designed for determining the number of endmembers or basis. The meaning of the sparse here is different from the other two algorithms. During the updating process, the algorithms would prune the unnecessary basis and just keep the important ones. In such manner, the algorithm can detect the correct number of the bases $\tilde{K}$ of the original data even starting with a large number of the dimension chosen randomly.

### 3.2.1 Projected Gradient Descent

Proposed by [55], the projected gradient descent algorithm is similar to the gradient descent algorithm discussed in the previous chapter, except that during each iteration, the algorithm project the elements of the updated matrices to be non-negative and adjust its $L_2$ norm to achieve desired sparseness, which is shown below:
Algorithm 6 Projected Gradient Descent

Require:
\[ V \in \mathbb{R}^{M \times N}_+, \quad W \in \mathbb{R}^{M \times K}_+, \quad H \in \mathbb{R}^{K \times N}_+, \]
where K is the number of basis.

1: loop
2: if Sparseness constrains is applied to \( W \) only then
3: \[ W \leftarrow W - \alpha \frac{\partial F}{\partial W} = W - \alpha (WH - V)H^T. \]
4: Projection Procedure (Columns of \( W \))
5: else
6: Take the standard multiplicative update.
7: end if
8: if Sparseness constrains is applied to \( H \) only then
9: \[ H \leftarrow H - \alpha \frac{\partial F}{\partial H} = H - \alpha W^T(WH - V). \]
10: Projection Procedure (Rows of \( H \))
11: else
12: Take the standard multiplicative update.
13: end if
14: end loop

The method to project the vector with a fixed \( L_2 \) norm and set the \( L_1 \) norm to some designed value will be presented here:
function PROJECTION PROCEDURE(Vector x, $L_1$ norm, $L_2$ norm)

$s_i \leftarrow \frac{L_1 - \sum_i x_i}{\text{dimension}(x)}$

$Z \leftarrow \{\}$

loop

$m_i \leftarrow \begin{cases} 
\frac{L_1}{\text{dimension}(x) - \text{size}(Z)} & \text{if } i \notin Z \\
0 & \text{if } i \in Z 
\end{cases}$

$s \leftarrow m + \alpha(s - m)$, where $\alpha \geq 0$ is selected to satisfy the $L_2$ norm.

if All components in s are nonnegative then

Return s

else

$Z \leftarrow Z \cup \{i; s_i \leq 0\}$

$s_i \leftarrow 0, \forall i \in Z$

$c \leftarrow \frac{\sum_i s_i - L_1}{\text{dimension}(x) - \text{size}(Z)}$

$s_i \leftarrow s_i - c \forall i \notin Z$

end if

end loop

end function

3.2.2 Alternating Nonnegativity-constrained Least Squares

Two similar algorithms SNMF/L and SNMF/R [56] are designed for the sparseness of $W$ and $H$. They are based on the alternating non-negativity constrained least squares to enforce sparsity on $W$ or $H$. The sparsity is measured by the $L_1$ norm minimization and and the minimization is divided into two sub-problems which are solved by a fast non-negativity constrained least squares algorithm that is improved on the active set based nonnegative least squares method.
To apply sparseness constraints on $W$, we formulate the following SNMF/L optimization problem:

$$\min_{W \in \mathbb{R}^{M \times K}, H \in \mathbb{R}^{K \times N}} \frac{1}{2} \|V - WH\|_F^2 + \eta \|H\|_F^2 + \alpha \sum_{i=1}^{M} \|W_{i,:}\|_1^2$$  \hspace{1cm} (3.2)

where $\eta > 0$ is a parameter to compress the Frobenius of matrix $H$ and $\alpha > 0$ is a regularization terms to balance the sparseness of $W$ and the approximation accuracy.

Starting with an initial nonnegative value of $H$. The algorithm iterates following ANLS until convergence [56]:

$$W = \arg\min_{W \in \mathbb{R}^{M \times K}} \left\| \left( \frac{H^T}{\sqrt{\alpha 1_{1 \times K}}} \right) W^T - \left( \frac{V^T}{0_{1 \times M}} \right) \right\|_F^2$$ \hspace{1cm} (3.3a)

$$H = \arg\min_{H \in \mathbb{R}^{K \times N}} \left\| \left( \frac{W}{\sqrt{\eta I_K}} \right) H - \left( \frac{V}{0_{K \times N}} \right) \right\|_F^2$$ \hspace{1cm} (3.3b)

where $1_{1 \times K}$ is a row vector with all components equal to one and $0_{1 \times M}$ is a zero vector, $I_K$ is an identity matrix of size $K \times K$, and $0_{K \times N}$ is a $K \times M$ zero matrix.

The equation 3.3.a can be written as

$$W = \arg\min_{W \in \mathbb{R}^{M \times K}} \left\| H^T W_{1,:) - V_{1,:} \right\|_2^2 + \alpha \left( \sum_{j=1}^{K} W_{1,j}^T \right)^2 + \cdots + \left\| H^T W_{M,:) - V_{M,:} \right\|_2^2 + \alpha \left( \sum_{j=1}^{K} W_{M,j}^T \right)^2$$ \hspace{1cm} (3.4)

Since all elements in $W$ are non-negative, we obtain the following formulation by
the definition of $L_1$-norm of a vector:

$$ W = \arg\min_{W \in \mathbb{R}^{M \times N}_+} \left\| H^T W^T_{1,:} - V^T_{1,:} \right\|_2^2 + \alpha \left\| W^T_{1,:} \right\|_1^2 + \cdots + \left\| H^T W_{M,:} - V^T_{M,:} \right\|_2^2 + \alpha \left\| W^T_{:,M} \right\|_1^2 $$

(3.5)

**SNMF/R**

To apply sparseness constraints on $H$, we formulate the following SNMF/R optimization problem:

$$ \min_{W \in \mathbb{R}^{M \times K}_+, H \in \mathbb{R}^{K \times N}_+} \frac{1}{2} \left\| V - WH \right\|_F^2 + \eta \left\| W \right\|_F^2 + \beta \sum_{j=1}^{M} \left\| H_{:,j} \right\|_1^2 $$

(3.6)

where $\eta > 0$ and $\beta$ are the same as before.

Starting with an initial nonnegative value of $W$. The algorithm iterates following ANLS until convergence [56]:

$$ H = \arg\min_{H \in \mathbb{R}^{K \times N}_+} \left\| \left( \frac{W}{\sqrt{\beta} 1_{1 \times K}} \right) H - \left( \frac{V}{0_{1 \times N}} \right) \right\|_F^2 $$

(3.7a)

$$ W = \arg\min_{W \in \mathbb{R}^{M \times K}_+} \left\| \left( \frac{H^T}{\sqrt{\eta} I_K} \right) W^T - \left( \frac{V^T}{0_{K \times M}} \right) \right\|_F^2 $$

(3.7b)

where notations are the same as that of the SNMF/L. The previous equation minimizes $L_1$ norm of columns of $H$ which enforces sparsity on $H$. 

40
Iterated Constrained Endmember

The iterated constrain endmember (ICE) algorithm [57] is designed for autonomous endmember detection, which is a difficult task in various fields. The ICE algorithm perform a least square minimization of the residual summation of the squares (RSS) based on the geometry model which mathematically has the form:

$$V_{:,j} = \sum_{i=1}^{K} W_{:,i} H_{i,j},$$

where $j=1,2,\cdots,N$.

The following condition holds to satisfy the constraint that the overall weights for the different bases sum to one, \(i.e\),

$$\sum_{j=1}^{K} H(i,j) = 1, \quad H(i,j) > 0, \quad j = 1, 2, \cdots, N \tag{3.8}$$

The RSS is defined as:

$$RSS = \sum_{j=1}^{N} (V_{:,j} - \sum_{i=1}^{K} W_{:,i} H_{i,j})^{T} (V_{:,j} - \sum_{i=1}^{K} W_{:,i} H_{i,j}) \tag{3.9}$$

which is the same as the Frobenius norm cost function for the NMF.

By minimization the RSS with the constraint Eq.3.9, we can obtain the desired result. However, the authors of [57] pointed out that the minimizer of the cost function is not unique. To solve this issue, the sum of squared distance (SSD) term is added
to the cost function, i.e.,

$$SSD = \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} (W_{i;i} - W_{i;j})^T (W_{i;i} - W_{i;j})$$

(3.10)

This term is proportional to the variance of the basis, which is to constrain the size of the basis space. Therefore, by adding this extra penalty term to the cost function, basis that can fit the original data tightly can be found. In [57], authors prove that

$$SSD = (K - 1)KA$$

(3.11)

where A is the sum of the variance over the basis. To make the cost function independent of the number of the basis K, A is used instead.

Therefore, the objective function for the original ICE algorithm is

$$RSS_{reg} = (1 - \mu) \frac{RSS}{N} + \mu A$$

(3.12)

where $\mu$ is a small positive value called the regularized parameter that balance the squared errors and the degrees of the sparseness.

A standard method to promote a particular constraint during a least squares minimization process is to add a penalty term to the objective function. In the sparsity constraint case, a weighted decay term is added to prevent the $H_{i,j}$ values from becoming large [58], i.e.,

$$RSS_{reg}^* = (1 - \mu) \frac{RSS}{N} + \mu A + SPT$$

(3.13)

where SPT is the penalty term that promote the sparseness of the matrix $H$, which
has the form:

\[ SPT = \sum_{i=1}^{K} \gamma_i \sum_{j=1}^{N} |H_{i,j}| \]  

(3.14)

Since all the elements in matrix \( \mathbf{H} \) are nonnegative, we can rewrite it as:

\[ SPT = \sum_{i=1}^{K} \gamma_i \sum_{j=1}^{N} H_{i,j} \]  

(3.15)

and we take

\[ \gamma_i = \frac{\Gamma}{\sum_{j=1}^{N} H_{old,i,j}} \]  

(3.16)

where \( \Gamma \) is the sparsity constant that controls the sparseness of the matrix \( \mathbf{H} \). The advantage of this dynamic expression of \( \gamma_j \) is that, as the contributions of the bases change during the iteration, the corresponding weights would adjust automatically.

If the sum of a particular basis’s contribution becomes small, then weight \( \gamma_j \) for that basis becomes larger. This dynamic weight adjustment accelerates the minimization of the contribution value.

Incorporating all the terms together, the cost function for the sparsity promoted ICE algorithms is [58] :

\[ \text{RSS}_{reg}^* = \frac{1-\mu}{N} \sum_{j=1}^{N} (V_{:,j} - \sum_{i=1}^{K} W_{:,i} H_{i,j})^T (V_{:,j} - \sum_{i=1}^{K} W_{:,i} H_{i,j}) + \mu A + \sum_{i=1}^{K} \gamma_i \sum_{j=1}^{N} H_{i,j} \]

\[ = \frac{1-\mu}{N} \left[ \sum_{j=1}^{N} (V_{:,j} - \sum_{i=1}^{K} W_{:,i} H_{i,j})^T (V_{:,j} - \sum_{i=1}^{K} W_{:,i} H_{i,j}) + \frac{N}{1-\mu} \sum_{i=1}^{K} \gamma_i \sum_{j=1}^{N} H_{i,j} \right] + \mu A \]  

(3.17)
In fact, considering a simplified case where the cost function is

\[ \text{RSS}_{\text{simple}} = -\frac{1}{2} \sum_{j=1}^{N} (V_{:,j} - \sum_{i=1}^{K} W_{:,i} H_{i,j})^2 - \sum_{i=1}^{K} \gamma_i \sum_{j=1}^{N} |H_{i,j}| \]  

(3.18)

Minimization the previous equation is equivalent to minimize the equation \( L \), which is:

\[
L = \ln \left[ e^{-\frac{1}{2} \sum_{j=1}^{N} (V_{:,j} - \sum_{i=1}^{K} W_{:,i} H_{i,j})^2 - \sum_{i=1}^{K} \gamma_i \sum_{j=1}^{N} |H_{i,j}|} \right] 
\]

\[
= \ln \left[ e^{-\frac{1}{2} \sum_{j=1}^{N} (V_{:,j} - \sum_{i=1}^{K} W_{:,i} H_{i,j})^2} e^{-\sum_{i=1}^{K} \gamma_i \sum_{j=1}^{N} |H_{i,j}|} \right] 
\]

(3.19)

The previous equation can be written as

\[ \ln(p(V|H)p(H)) \]

which is the logarithm of two products, where

\[ p(V|H) = e^{-\frac{1}{2} \sum_{j=1}^{N} (V_{:,j} - \sum_{i=1}^{K} W_{:,i} H_{i,j})^2} \]

\[ \propto \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2} |\sum_{j=1}^{N} (V_{:,j} - \sum_{i=1}^{K} W_{:,i} H_{i,j})|^2} \]

(3.20)

is the probability density of the original data given the parameter matrix \( H \) and \( p(H) \) is the prior distribution of the parameter. In [56], the author use a Laplacian distribution to promote the sparsity, which is what we have implemented in the section.

The minimization of the objective function can be done using quadratic programming. As a special kind of mathematical optimization method, quadratic programming is to optimize a quadratic function of several variables with respect to some linear constraints on these variables. The problem can be formulated as: Assume \( x \)
and \(c\) are \(n\)-dimensional vector and \(Q\) is a symmetric \(n \times n\) matrix.

\[
x = \arg\min_x f(x) = \frac{1}{2} x^T Q x + c^T x
\]

\[\text{s.t } Ax \leq b \quad \text{(Inequality constrain)}
\]

\[Ex = d \quad \text{(Equality constrain)}
\]

The cost function of the sparsity promoted ICE Eq.3.17 can be formulated as [58]:

\[
RSS^*_{reg} = \frac{1 - \mu}{N} \sum_{j=1}^{N} (V_{:,j} - \sum_{i=1}^{K} W_{:,i} H_{i,j})^T (V_{:,j} - \sum_{i=1}^{K} W_{:,i} H_{i,j}) + \mu A + \sum_{i=1}^{K} \gamma_i \sum_{j=1}^{N} H_{i,j}
\]

\[
= \frac{1 - \mu}{N} \sum_{j=1}^{N} (V_{:,j}^T V_{:,j} - V_{:,j}^T W H_{:,j} - H_{:,ij}^T W^T V_{:,j} + H_{:,ij}^T W^T W H_{:,j}) + \mu A + \sum_{i=1}^{K} \gamma_i \sum_{j=1}^{N} H_{i,j}
\]

\[
= \sum_{j=1}^{N} \frac{1}{2} H_{:,j}^T \frac{2(1 - \mu)}{N} W^T WH_{:,j} + (\gamma - 2 \frac{1 - \mu}{N} W^T V_{:,j})^T H_{:,j} + e
\]

where \(\gamma\) is a \(K\) dimensional column vector whose elements are \([\gamma_1, \gamma_2, \ldots, \gamma_K]\), and \(e\) is constant that is independent of the \(H_{:,j}\). It has the quadratical form and subjects to a linear constrain. We can use the quadratical programming to optimize the cost function.

The ICE algorithm minimizes the cost function iteratively. For the first iteration of the algorithm, matrix \(W\) is set to randomly chosen values. After solving matrix \(H\) using the quadratic programming, matrix \(W\) is calculated using the current estimate of matrix \(H\), which is:

\[
W = \{(HH^T + \lambda(I_K - \frac{11^T}{K}))^{-1}HV^T\}^T
\]
where $I_K$ is the $K \times K$ identity matrix, $1$ is the $K$-dimensional all-one vector and
\[ \lambda = \frac{N \mu}{(K-1)(1-\mu)}. \]
This iteration stops when the value of the cost function is smaller than the tolerance value, or the maximum number of iterations is reached. Of course, there is no guarantee that the result obtained from the iteration is strictly nonnegative. Therefore, we use the same approach as we did in the Alternating Least Squares algorithm for NMF, which is to project the elements of $W$ to the nonnegative orthant.

During the iterative minimization process, bases can be pruned as their contributions drop below a specific threshold. After every iteration, the maximum contribution values for every basis can be obtained by

\[ MAX(H_i) = \max_j H_{i,j} \]  \hspace{1cm} (3.24)

If the maximum contribution for a basis drops below the threshold, then this basis can be pruned from the candidate set. The algorithm of the sparsity promoted ICE is shown below [58]:
Algorithm 7 Sparsity promoted ICE

Require:
\[ V \in \mathbb{R}^{M \times N}, W \in \mathbb{R}^{M \times K}, H \in \mathbb{R}^{K \times N}, \text{where } K \text{ is the number of basis.} \]

1: **loop**
2: \( H \leftarrow \arg\min_{W} \text{RSS}^{*} \text{reg} \) using quadratic programming.
3: \[ W \leftarrow \{[HH^{T} + \lambda(I_{K} - \frac{1}{K}I^{T})^{-1}HV]^{T} \}
4: \( W = [W]_{+} \)
5: **loop**
6: Find the \( \text{MAX} (H_{i}) \)
7: if \( \text{MAX} (H_{i}) < \text{threshold} \) then
8: Prune the basis
9: else
10: Retain the basis
11: end if
12: **end loop**
13: Renew the basis matrix \( W \), weights matrix \( H \), the number of basis \( K \), \( I_{K} \), \( 1 \) and the values of \( \lambda \).
14: **end loop**

By penalizing the cost function when a large number of basis is used, the sparsity-promoting term enforces the weights of the unnecessary basis to zero so that a more compact way to model the original data is found. There are still a number of parameters such as the regularization parameter \( \mu \) and the sparsity constant \( \Gamma \) to be set when the algorithm is initialized. The choice of these parameters are often problem dependent, which requires the experiment beforehand. Setting these parameters automatically remains to be an open question. Moreover, since the introduction of the projection step theoretical analysis of the convergence property of the algorithm is difficult to achieve.
3.2.3 Performance Comparison

As we did in the last chapter, we apply the sparse NMF to audio signal processing to represent the spectrogram of the signal. The result can be seen in Fig. 3.2, where the spectrogram of a mixed signal is shown with the factorization result. The mixed signal is a 10 second long fall signal performed by some stunt actors adding with TV audio signal and Gaussian white noise to simulate the interference and noise. The result is conducted by the sparsity promoted ICE when the initial number of basis $K = 60$, the regularization parameter $\mu = 0.004$ and sparsity constant $\Gamma = 0.01$.

Figure 3.2: Sparse NMF using sparsity promoted ICE of a fall signal mixed with TV interference and white Gaussian noise when $K = 60$, $\mu = 0.004$ and $\Gamma = 0.01$.

Fig. 3.3 is a typical example of the convergence performance of the Projected Gradient Descent, the Alternating Nonnegative-constrained Least Squares and the Sparsity promoted Iterated Constrain Endmember algorithm. Number of the initial basis $K$
is set to 60 as the previous case. The reconstruction error is defined as $||V - WH||_F^2$.

Some interesting properties of these three algorithms are shown in this figure.

First, as the extension of the conventional gradient descent algorithm, the projected gradient descent is easy to implement but it converges much slower than the other two algorithms as is shown in the figure.

Second, compared with the inexact alternating least squares algorithm which we have discussed in Chapter Two, the alternating nonnegative constrained least squares is more stable, and theoretical proof has been given concerning its convergence prop-
tery. However, this algorithm would significantly increase the computation when guaranteeing convergence to a local optimum. With the same number of iterations, the projected gradient would take about 6 seconds on an Intel 2.2GHz i7 core laptop with 6GB of RAM while the ANLS would take approximately 11 seconds.

The third intriguing feature of the figure is that the reconstruction error of the sparsity promoted ICE is not monotonically decreasing. The reconstruction error drops dramatically at the first few iterations, then slightly increases and saturates at a fixed level. Because the cost function for the sparsity promoted ICE is

\[ RSS_{reg}^* = \frac{(1-\mu)RSS}{N} + \mu A + SPT \]

During the iterations, the algorithm minimizes the cost function and balance the residual sum squares, the sum of squared distances and the sparsity promoting term. The objective of the sparsity promoted ICE is to find an appropriate number of basis to model the original data rather than accomplishing perfect factorization. Fig.3.4 illustrates how these three terms change over the iteration.
In this setting, the sparsity term dominates the cost function.

Finally, although the reconstruction error of the sparsity promoted ICE is the largest among three algorithms at the end of the iteration, this algorithm generates more convincing model to represent the original data. Fig.3.5 illustrates the change of the reconstruction error when different number of bases are implemented to reconstruct the original data $V$. All the bases for three algorithms are arranged in descent order in terms of their contributions. Notice that the reconstruction error of the sparsity promoted ICE decreases faster than the other two algorithms. Moreover, it is a three-segment piecewise function. Recall that the original signal contains fall signal,
audio interference and background noise. We can assume that the bases within each segment can be used to reconstruct fall signal, audio interference and background noise. Therefore, the sparsity promoted ICE is better in the sense that it provides a generative model which can be used as the building blocks of the original data.

![Figure 3.5: Change of the reconstruction error for different sparse NMF algorithms using different number of basis.](image)

We can change the parameters of the sparsity promoted ICE to adjust the weights for different terms. For example, a small $\mu$ fits the low noise data since it increases the penalty for the imprecise approximation while greater sparsity requires larger $\Gamma$. An exploration of the effect of the sparsity constant $\Gamma$ is presented below:
Fig. 3.6 reveals the actual number of basis $\tilde{K}$ found by the sparsity promoted ICE, which is taken over 50 ensemble runs. Notice that when the sparsity constant $\Gamma$ increases, the number of the bases found by the sparse NMF decrease. This phenomenon verifies the theoretical analysis about the effect of the $\Gamma$ that larger sparsity constant yields greater sparsity.

However, when the number of basis drops below a certain threshold, which means that we use an insufficient number of basis to model the original data, the approximation accuracy would decrease as is shown in Fig.3.7. How to choose $\Gamma$ to accommodate the particular application remains to be an open question and most of the time re-
Fig. 3.7: Influence of $\Gamma$ on the factorization accuracy

Fig. 3.8 shows the actual number of bases $\tilde{K}$ found by the sparsity promoted ICE when the initial number of the basis $K$ varies. The values are the average from the 50 trials. Using the sparsity promoted ICE, sparse NMF is able to determine the number of bases automatically and even when the initial number of basis keeps increasing, it remains to settle at the value of 31 or 32.
Figure 3.8: The actual number of basis found by sparsity promoted ICE when increasing the initial number of basis

3.3 Summary

In this chapter, sparse NMF is introduced as a technique to decompose a non-negative matrix into the product of two matrices which might be meaningful components. Three basic algorithms for sparse NMF, gradient descent, alternating least squares and sparsity promoted ICE, are also presented in this chapter. The first two algorithms are the extension of the traditional NMF algorithms. All three algorithms achieve particular sparseness by adding the penalty term to the objective function
and update the matrices $W$ and $H$ iteratively. However, the sparsity promoted ICE would prune the unnecessary bases by thresholding during each iteration to find out the appropriate number of bases. Although sparsity promoted ICE based sparse NMF is slower to run than the other two algorithms, it generates more meaningful result that can be used as the building blocks of the original data.
Chapter 4

Classification Techniques

We use audio signal as an example to illustrate classification. The purpose of the audio signal classification is to extract dependent features from a sound and to use these features to identify which set of classes the sound is most likely to suit. The feature extraction and classifier can be quite diverse depending on specific application. This chapter introduces the general concept about the audio signal processing. The introduction of some popular features and classifiers for the audio signal processing are discussed later.

4.1 Features for Audio Signal Processing

Usually, the first step in the audio signal classification problem is feature extraction. Most audio signal contains much redundancy, and important features are embedded in the original signal. The feature extraction refers to the process of discovering a few essential components which are unique to each sample. The input for the
audio signal processing systems is an audio signal in the form of voltages. The real important information is usually features like frequency, spectral content, etc. These features can be physical based on measurable variables or perceptual based on how human beings perceive.

In some application, pre-processing signal has been used to eliminate unnecessary information and enhance the original signal. For example, in automatic fall detection system, the signal enhancement achieved by the source separation is implemented to filter out the noise and the interference. Extra care should be put to the pre-processing process since it has the potential to remove useful information for classification.

The features that are used in audio signal processing systems are usually divided into two classes: perceptual features and physical features. Perceptual features are based on how human being hear sound, which include pitch, timbre and rhythm. While physical features concentrate on mathematical and statistical characteristic of signals that include spectral features and Mel-frequency Cepstral Coefficients (MFCCs) [59]. Since we apply the audio signal processing to fall signal detection, and most of the fall signal is burst like, we will focus on the discussion of the physical features.

4.1.1 Spectral Features

The spectrum of a signal describes the frequency distribution of the signal. It has been shown that the human performs the spectral analysis when we process the audio signal to some extent. Since humans extract spectrum information when we perceive the audio signal, it is reasonable to design an audio signal processing system using spectrum information. Apart from that, spectral techniques have been used to analyze
and classify sound for a long time.

Most of the spectral features are extracted on a frame by frame basis. Without proper window function or just implementing a simple rectangle window function, the frequency leakage would occur. This phenomenon is caused by the fact that spectrum of the rectangle has relative wide side-lobe which could spread the original frequency and might mask the important spectrum elements at even lower levels. To prevent this phenomenon, some of the well-known window functions are introduced when taking the frames or the audio signal.

Depending on the classification purpose, many physical characteristics of the spectrum of a signal can be used for classification. One of the most fundamental spectral measures is bandwidth, which is a measure of what range of frequency components are present in the signal. This feature can be used to classify speech and music, where music typically occupies larger bandwidth than speech does.

Other general feature called harmonicity is also used as a feature in various classification application, which refers to the relationships between peaks in the spectrum. If an object vibrates in a resonant manner, such as a musical instrument or human voice, its frequency components will peak at evenly spaced intervals over the spectrum. The harmonicity of an audio signal can be used to discriminate voiced and unvoiced speech and to identify different musical instrument.

Another widely used feature is the spectrogram, which is the time-varying spectrum of a signal. To generate a spectrogram, the signal is broken into frames. The spectrum is calculated on each frame and these elements are displayed as a time-varying spectrum. The result is a measurement how the frequency contents of the signal change over time. They are used extensively in the areas of music, sonar, radar,
speech processing and seismology, etc.

4.1.2 Mel-frequency Cepstral coefficients [59]

The Mel-frequency Cepstral coefficients (MFCCs) is one of the most commonly used features for the speech recognition application, which takes the properties of human’s acoustic perception sensitivity with respect to frequencies into account.

Usually, the computation for MFCC can be decomposed into 6 stages which is shown in the Fig.4.1

The description of these blocks will be presented below:

1. Pre-emphasis:

   The purpose of the pre-emphasis is to compensate the high-frequency component suppressed during the sound production. Moreover, it can also amplify the high-frequency components. It is implemented by a first order high-pass filter which has the form:

\[ y(n) = x(n) - \eta x(n - 1) \]  \hspace{1cm} (4.1)
where usually $\eta \in [0.95, 1]$

2. Segmentation and windowing:

The original audio signal is segmented into several frames with 20-30 ms length at certain overlap rate between $\frac{1}{3}$ and $\frac{1}{2}$. Usually, the frame size is equal to the power of two in order to accommodate the FFT. If this is not the case, zeros are padded to the nearest length of power of two. After separating, every frame will be multiplied by a window function (usually Hamming window) to minimize the boundary effect caused by the segmentation.

3. FFT:

The Fast Fourier Transform (FFT) is implemented to transform the time domain signal to frequency spectral domain features which is applied to each frame separately.

4. Mapping and Filtering:

After windowing, the magnitude frequency response of each frame is multiplied by a set of $L$ triangular bandpass filters to get its log energy. The positions of these filters are equally separated in Mel frequency, which is related to the linear frequency $f$ by the following equation [59]:

$$mel(f) = 1125ln(1 + \frac{f}{700}). \quad (4.2)$$

where $f$ denotes the linear frequency ranging from 0 to the highest frequency of the signal. The log energy $E_l$ for the $l$th band pass filter is generated where $l = 1, 2, \cdots, L$
5. Discrete Cosine Transform (DCT) [60]:

We apply the DCT on the J log energy $E_l$ obtained from the triangular bandpass filters to generate J MFCCs for the $i^{th}$ frame which is shown in the equation below:

$$C_{j,i} = \sum_{l=1}^{L} \cos \left( \frac{(j + 1)(l - 0.5\pi)}{L} \right) E_l$$  \hspace{1cm} (4.3)

where $j=1, 2, \cdots, J$.

6. MFCCs matrix:

Repeating the procedure 4 and 5 for all frames to obtain the MFCC matrix $C$ for one audio sample:

$$C = \begin{bmatrix} C_{1,1} & \cdots & C_{1,P} \\ \vdots & \ddots & \vdots \\ C_{J,1} & \cdots & C_{J,P} \end{bmatrix}$$  \hspace{1cm} (4.4)

where $P$ is the total number of the frame of the signal.

### 4.2 Classifier

After the feature extraction, the next task is to decide in which of these classes a particular feature sample lies, which is known as classification. In a typical classification system, there are many samples to be grouped a small number of categories. The target of the classification is assigning the label to a particular sample, which help us define the origin of the signal. A classifier generates decision boundary in the feature space to separate different classes from one another.

Classifier design is a broad research topic. It is important to use an efficient classifier that discriminates the different audio signal with less computation and pre-
serve the accuracy at the same time. A brief overview of the classifier implemented in the thesis, k-nearest neighbor (kNN), is presented below.

4.2.1 k-nearest neighbour(kNN)

The k nearest neighbor method is to assign the label of the training sample to the unlabelled testing sample which is nearest in the feature space. In kNN, a training set is used to determine the class of a previously unknown sample X. An appropriate distance measure in the feature space is used to determine k elements in the training set closest to a particular testing sample. If most of these k nearest neighbors belong to a particular class, then this testing sample gets assigned accordingly. This classification framework generates a nonlinear decision boundary. Furthermore, the number of samples used in the training set can be considerably reduced for faster implementation since only the examples that are close to the decision boundary are required. This can be illustrated by the following example.
Referring to Fig. 4.2, each of the samples has been labeled either A or B marked by dots or stars respectively. The unlabeled data sample $x$, marked by a circle, needs to be classified using the kNN classifier takes the $k$ nearest neighbors around it and uses them to assign a label. This is usually done by a majority-voting rule.

For example if $k=1$, the sample that is nearest to the sample $x$ belongs to class B. Then the unknown sample $x$ is assigned as B. But if $k=5$, then there are four samples of A and one sample of B that have the shortest distance to the sample $x$. Now the sample $x$ is assigned as A by the majority. Therefore, the value of $k$ is critical in order to assign an unknown sample by its nearest neighbors. Note that all the neighbors
around testing sample are assumed to have the identical influence on classification irrespective of the relative distance. Although the only sample of B is the closest to the point x among all the five neighbors, it still gets assigned to the class A regardless of the distance difference.

There exist two main problems in this classifier that has to be addressed. The first is to find a suitable distance measure that which sample is closer to the sample to the sample x.

There are many distance metrics that are used to calculate the distance between the samples. The first one is the Euclidian distance, which is defined as:

\[
dist(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}
\]  
(4.5)

where x and y are the feature vectors and n is the corresponding dimension.

Another option is the \(L_1\) norm distance, which is defined as:

\[
dist(x, y) = \sum_{i=1}^{n} |x_i - y_i|
\]  
(4.6)

The second problem is the choice of k, choosing a large k results in a linear classifier however small k yields a nonlinear case, which influences the generalization capability of the kNN classifier. The optimal k is usually problem dependent. The disadvantage of this method is its large computing, since for classifying an unknown sample all the pairwise distance between the training set and the testing has to be calculated.
4.3 Summary

In this chapter, we introduce some of the well-known approaches to extract the features of the audio signal and several popular classifier to do classification. Some features like energy, zero-crossing rate describe the basic physical properties of the audio signal. They are easy to implement but demonstrate very little about the essence of the signal. However, for the spectral features and the Mel-frequency Cepstral coefficients, since we analyze the audio signal based on its spectral characteristics, it resembles the acoustic system of human beings. They are widely used in various applications in audio signal processing. Followed by the feature extraction, the classification is to assign a label to a testing sample, which can help us define the origin of the signal. The k-nearest neighbor is a commonly used classifier. It assigns the label of the training sample to the unlabelled sample that is nearest in the feature space based on some distance measurements. It is simple to use and sensitive to the number of k.
Chapter 5

Source Separation using NMF Vs Sparse NMF

The objective of source separation is to recover the signal components in a mixture. A famous example of the source separation problem is the cocktail party problem, where several people talk simultaneously, and the listener attempts to extract one of the discussions from the mixture of sound sources. The human brain can handle this type of source separation problem subconsciously, but it is a difficult task in audio signal processing because the characteristic of the signal might vary and a general source separation algorithm is hard to find.

Several different methods have been proposed for this problem and most of the approaches are developed for a particular application where the number of sources, the characteristics of the sources and how these signals are mixed together are all known to us. Approaches like PCA and ICA are two most widely used methods for source separation. By minimal correlating the sources or maximal independence
of the sources, the mixed signal is separated using these two methods. They have relatively good performance when the problem is simplified with no delays present. Another well-known approach is the NMF, which imposes the nonnegative constraints on the source signals to find the corresponding basis. A common extension of this approach is to impose some low-complexity constraint on the signal, such as sparsity to enhance the performance. This method can be particularly effective if only the prominent features for a particular source is required.

In this chapter, the proposed technique for the source separation using NMF and sparse NMF is presented, and the performance will be evaluated using some simulated data acquired from the real application.

### 5.1 Single-channel Source Separation

The single channel source separation for the audio signal processing can be divided into five steps: Frame Segmentation and STFT, NMF, Basis Separation, Overlap and add IFFT and signal reconstruction, which are presented below.

#### 5.1.1 Frame Segmentation and STFT

The first stage for the NMF based source separation is to construct a data matrix $V$ for decomposition. The matrix $V$ is the spectrogram or the magnitude of STFT of the mixed signal. Segment the original signal into a series of frames. Let $L$ be the total length of the signal, $F$ be the frame size and $Q$ be the overlap length between neighboring frames. The number of frames which equals to the columns of the matrix...
\( \mathbf{V} \) is

\[
N = \left\lceil \frac{L - F}{F - Q} \right\rceil + 1 \tag{5.1}
\]

Here the symbol \( \lceil * \rceil \) stands for the largest integer value no greater than *. 

After segment the original signal into several frames, the STFT is applied to each frame to get the columns of \( \mathbf{V} \)

\[
\mathbf{V}_{k,i} = \left| \sum_{n=0}^{F-1} s_i(n)w(n)e^{-j\frac{2\pi kn}{F}} \right| \tag{5.2}
\]

where \( w(n) \) is the window function to keep the continuity of the first and the last points in the frame, \( s_i \) is the \( i^{th} \) frame of the original mixture signal and \( \lvert * \rvert \) stands for the magnitude of a complex number. Because of the symmetry of the STFT, only the first half of the result is retained for decomposition. In such manner, the column of \( \mathbf{V} \) is the magnitude of STFT of the mixture signal.

### 5.1.2 NMF

Recall that mathematically, NMF or sparse NMF has the form:

\[
\mathbf{V} = \mathbf{WH}
\]

where \( \mathbf{V} \) is \( M \times N \), \( \mathbf{W} \) is \( M \times K \) and \( \mathbf{H} \) is \( K \times N \). The columns of matrix \( \mathbf{W} \) can be viewed as the frequency basis, and the row of matrix \( \mathbf{H} \) is regarded as the corresponding contribution at different frames. This equation can also be rewritten as:

\[
\mathbf{V} = \sum_{i=1}^{K} \mathbf{W}_{i,:} \mathbf{H}_{i,:} = \mathbf{W}_{1,:} \mathbf{H}_{1,:} + \mathbf{W}_{2,:} \mathbf{H}_{2,:} + \cdots + \mathbf{W}_{K,:} \mathbf{H}_{K,:}
\]
Since each source for the mixed signal has different frequency composite which can be discriminated into P different classes, and each corresponds to unique source. If we can group these K matrices into P classes, \( i.e., \)

\[
V = WH = \begin{bmatrix} W_1, & W_2, & \cdots, & W_P \end{bmatrix} \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_P \end{bmatrix} = W_1H_1 + W_2H_2 + \cdots + W_PH_P \tag{5.3}
\]

where

\[
W_p = \begin{bmatrix} W_{i,1p}, & W_{i,2p}, & \cdots, & W_{i,Kp} \end{bmatrix} \tag{5.4}
\]

\[\sum_{p=1}^{P} Kp = K \] and \( W_{i,i_p}, i = 1, 2, \cdots, Kp \) is the rearranged frequency basis for the \( p^{th} \) source, then we can use the corresponding \( W_pH_p \) to approximate the spectrogram of the \( p^{th} \) source which ultimately leads to the time domain signal of the \( p^{th} \) source.

### 5.1.3 Basis Separation

This section will introduce the process how to separate the corresponding basis for a single channel source separation using NMF for the fall signal source separation problem. The performance of source separation using NMF greatly depends on how well we can classify each basis. An effective unsupervised technique [25] using the characteristic of the fall signal is proposed for the separation of basis vectors.

The row of matrix \( H \) normalized by its \( L_2 \) norm is used to separate the fall signal. Recall that the row of matrix \( H \) corresponds to the contribution of a particular basis over the frames. Since most of the fall signals happen within a short period (usually
less than 1 second) as illustrated in Fig.5.1.

Figure 5.1: A mixture signal containing fall, TV audio and Gaussian white noise signal.

Theoretically, the energy of the fall basis should concentrate when the fall happens and has relatively small values otherwise. Fig.5.2 presents four of such normalized bases using the multiplicative update algorithm with KL divergence where the number of bases is 60. The top two figures correspond to the fall signal and the latter correspond to the non-fall signal. As is shown in Fig.5.2, (a) and (b) reveal an obvious impulse like peak where the fall occurs. Meanwhile, due to the continuity of the TV audio interference and the Gaussian white noise, (c) and (d) display a random
pattern. As long as we can capture the spiky features in the normalized row of matrix H, we can separate the fall signal and non-fall signal.

Figure 5.2: Plots of some selected normalized row of matrix H. (a)k=17, (b)k=31, (c) k=1, (d) k=41

The selection of the proper basis is based on the energy ratio. The energy ratio is defined as the Euclidean norms of the rows of H during the occurrence of the fall, which is formulated as

$$E_k = \sum_{n=1}^{\Delta n_L + \Delta n_R} \tilde{H}_{k,N_0-\Delta n_L+n}^2$$

(5.5)

where $\tilde{H}_{i,j}$ is the $j^{th}$ element of the $i^{th}$ row of H normalized by its $L_2$ norm, which
mathematically equals to $\tilde{H}_{i,:} = \frac{H_{i,:}}{\sum_{j=1}^{N} H_{i,j}}$. In Eq.5.5, the $N_0$ is the estimated frame index when the fall occurs, $\Delta n_L$ and $\Delta n_R$ are the corresponding bias by which the window size is decided to calculate the energy within certain portion.

However, in practice, the exact fall portion is not a known prior. A sliding window is required to calculated the energy ratio for all the candidate fall portion. We assume that the fall portion always lies in the middle of a sliding window. The procedure to calculate the energy ratio for a particular row of $H$ is presented below:

**Algorithm 8 Energy Ratio Calculation**

**Require:**
- Normalized row vector $\tilde{H}_{k,:}$.

1: loop
2: $E_{k,i} = \sum_{n=1}^{\Delta n_L + \Delta n_R} \tilde{H}_{k,i-n}^2$, where $i = \Delta n_L, \Delta n_L + 1, \cdots, N - \Delta n_R$
3: end loop
4: Find the maximum value of $E_{k,i}$ and take it as $E_k$

By thresholding the energy ratio, the fall bases can be separated from the non-fall bases. This task is accomplished through a discriminator with a positive threshold $\Theta$, whose decision rule is expressed as:

$\tilde{H}_{k,:}$ is \begin{align*}
\begin{cases}
\text{a fall basis} & \text{if } E_k \geq \Theta \\
\text{a nonfall basis} & \text{if } E_k < \Theta
\end{cases}
\end{align*}

(5.6)

The value of $\Theta$ can be manually chosen or adjusted by calculating the energy of the non-fall frames before any possible fall signal is detected.
In Fig.5.3, the threshold Θ is chosen as 0.75. In this example, 16 bases are selected to reconstruct the fall signal.

After selection of the fall basis using the energy ration, the magnitude of STFT for the framed fall signal $V_f$ can be built by the approximation $V_f = W_f H_f$ as is shown in the Fig.5.4 and Fig.5.5.
Figure 5.4: NMF source separation example
Since the characteristic of the fall signal depends on its magnitude description, therefore phase of the STFT for the fall signal can be approximated by the phase of the mixture signal $\angle S_{mixture}$ which acquired during the frame separation and STFT process. Then the STFT of the fall signal can be formulated as $S_f = V_f / S_{mixture}$. After that, the time domain fall signal can be reconstructed using the inverse Fast Fourier Transform (IFFT) for an individual frame and the overall fall signal is generated by the overlap and add. Fig.5.6 illustrates the separated waveform of the fall
signal together with the mixture signal.

Figure 5.6: The time domain mixture signal and separated fall signal using the proposed single channel NMF source separation

5.2 Multichannel Source Separation

Given the source position, the conventional delay-and-sum beamformer (DSB) can reduce the noise of the separated source signal. We want to extend the proposed single-channel NMF source separation to a multichannel array, which applies a conventional signal channel NMF to factorize the multichannel signal jointly. In an ideal
case the NMF result of the a particular channel is the same as that from another channel except the relative time delay. However, due to the existence of the noise, this case will never happen. Factorizing the signals using NMF jointly over multiple channels can reduce the noise to accomplish better factorization accuracy since the noises from different channels are independent of one another. Most of the procedures for the multichannel NMF source separation are the same as we discussed in the single channel separation. Except after reconstruct the signal for each channel, DSB is then implemented to enhance the result.

5.2.1 Formulating Multichannel Factorization Matrix

To perform the joint decomposition, we define the matrix $V^{(j)}$, which is generated by the magnitude of the STFT of the $j$th channel signal $s_j(t)$, as

$$V^{(j)} = [V_{i;1}^{(j)}, V_{i;2}^{(j)}, \ldots, V_{i;N}^{(j)}]$$ (5.7)

The matrix $V$ with $J$ channels for factorization is formulated as follows:

$$V = [V_{i;1}^{(1)}, V_{i;1}^{(2)}, \ldots, V_{i;1}^{(J)}, V_{i;2}^{(1)}, V_{i;2}^{(2)}, \ldots, V_{i;N}^{(J)}, \ldots, V_{i;1}^{(2)}, V_{i;2}^{(2)}, \ldots, V_{i;N}^{(2)}, \ldots, V_{i;1}^{(J)}, V_{i;2}^{(J)}, \ldots, V_{i;N}^{(J)}]$$ (5.8)

where $V$ is $M \times NJ$ for a $J$-channel joint factorization. After NMF, $W$ remains to be $M \times K$ while $H$ is $K \times NJ$. The proposed basis separation is utilized to jointly separate the fall basis from $H$. The columns of the matrix $H$ can be partitioned into $J$ submatrices by picking the columns of $H$ at a interval of $J$. Then the basis
contribution matrix $H^{(j)}$ for the $j^{th}$ channel is

$$H^{(j)} = \left[ H_{:,j}, H_{:,J+j}, \ldots, H_{:,N-1+j+j} \right], \text{ where } j = 1, 2, \ldots J$$

(5.9)

Therefore, we can reconstruct the separated source signal $s_j(t)$ for the $j^{th}$ channel using the selected bases from $H^j$. The enhanced fall signal can be represented as

$$\tilde{s}(t) = \frac{1}{J} \sum_{j=1}^{J} s_j(t + \tau_j)$$

(5.10)

where $\tau_j$ stands for the relative time delay at the $j^{th}$ channel. The time delay $\tau$ is calculated by the relative distance between the source position and the microphone array.

The source position is estimated using the source localization algorithm. One of the well-known algorithms for source localization is called steered response power with the phase transform (SPR-PHAT) [61]. The general idea of this algorithm is that the source position can maximize the output power of a DSB beamformer, which is also the summation of the pairwise cross-correlation functions across all microphones.

Using the spherical coordinate, the range is defined as $r$, elevation as $\phi$, and azimuth as $\theta$. Then the relative delay between two microphones can be expressed as:

$$\tau_{i,j}(r_s, \phi_s, \theta_s) = \frac{1}{c} \left[ d_{s,j}(r_s, \phi_s, \theta_s) - d_{s,i}(r_s, \phi_s, \theta_s) \right]$$

(5.11)

where $c$ is speed of the sound, $d_{s,j}$ stands for the distance between the sound source and microphone $j$ and $(r_s, \phi_s, \theta_s)$ stands for the spherical coordinates of the source. Here is the simplified version of the SPR-PHAT algorithm for source localization:
Algorithm 9 SPR-PHAT Source Localization

Require:
- Initialize the coordinates for source location and set $S^{SPR-PHAT}(r_s, \phi_s, \theta_s)$ to 0.

1: loop
2: for all candidate source position
3: Look up $\tau_{i,j}(r_s, \phi_s, \theta_s) \leftarrow \tau_{i,j}(r_s, \phi_s, \theta_s)$
4: $S^{SPR-PHAT}(r_s, \phi_s, \theta_s) \leftarrow S^{SPR-PHAT}(r_s, \phi_s, \theta_s) + \sum_{j=0}^{N_f-1} G_{x_i,x_j}(k) e^{j2\pi kr_s(r_s, \phi_s, \theta_s)}$ 
5: end loop
6: $(\tilde{r}_s, \tilde{\phi}_s, \tilde{\theta}_s) \leftarrow \text{argmax}_{(r, \phi, \theta) \in L} S^{SPR-PHAT}(r_s, \phi_s, \theta_s)$

where $L = L_r \times L_\phi \times L_\theta = \{r_0, r_1, \cdots, r_{N_r}\} \times \{\phi_0, \phi_1, \cdots, \phi_{N_\phi}\} \times \{\theta_0, \theta_1, \cdots, \theta_{N_\theta}\}$ is the candidate location space.

As a conventional method, solving the problem requires traversing the three-dimensional location space, which is computational intensive. A modified version of this algorithm is introduced in [57], where only a one-dimensional relative delay space is traversed, which is a computationally viable implementation of the original approach.

Notice that most of the time delay is not necessarily an integer, the implementation of the beamforming is usually implemented in the frequency domain before the IFFT.

5.2.2 Performance Evaluation using NMF vs Sparse NMF

The purpose of this section is to validate the proposed source separation approach for enhancing the fall signal by reducing the interference and noise. What Fig. 5.7 reveals is the reconstructed spectrogram for the fall signal and non-fall signal. The figure (c) and (d) are the result for the NMF, (e) and (f) are the result for the sparse NMF. Both approaches can reconstruct the fall signal quite effectively. However, sparse
NMF is better in terms of the fall signal residue. Although the NMF can reconstruct the fall signal as well, it seems that a portion of fall signal has leaked into the non-fall section. On the contrary, the result of sparse NMF is better.

Figure 5.7: Reconstructed spectrogram for the fall signal and non-fall signal.

The Fig.5.8 present the performance comparison as the SNR increases, when SIR=10dB and K=60. The mean square error (MSE) of the vertical axis is defined
as:

\[
MSE = \frac{\sum_{t=1}^{T} (\tilde{s}(t) - s(t))^2}{\sum_{t=1}^{T} s^2(t)}
\]  

(5.12)

It is a term which measures the relative reconstruction error for the fall signal. As is shown in the Fig. 5.8, sparse NMF always yields better results for the same number of iterations unless the SNR is below -14 dB, sparse NMF always yields better result.

Figure 5.8: Mean square error for separated fall signal when SIR=5dB and K=60
5.3 Summary

In this chapter, we introduce the NMF based source separation technique. The objective of source separation is to recover every signal component in a mixture. For single channel source separation, starting with a mixture audio signal $s$, we segment the original signal into several frames. Then we take the magnitude of spectrogram to generate the data matrix for factorization. After NMF, the fall bases are collected based on its energy ratio to generate the spectrogram of the fall signal. Finally, the overlap and add IFFT is applied to reconstruct the time domain estimated fall signal. For multichannel source separation, the procedures are approximately the same, except that we need to mix the spectrogram of the multiple channels together to generate the combined data matrix for factorization and when we reconstruct the time domain signal, the delay and sum approach should be applied to correct the time delay among different channels. The performance evaluation reveals that the sparse NMF based source separation achieves better performance in terms of the mean square error since the result of the sparse NMF provides better representation of the original data.
Chapter 6

Fall Detection Application

In this chapter, we would like to evaluate the performance of the fall detection using NMF and sparse NMF based source separation algorithm. A detailed description of the fall detection application including the data measurement, data simulation, experiment configuration and the final result will be presented. The configuration of the experiment and the description is provided below:

6.1 Data Measurement

We have obtained approval by the Institutional Review Board, University of Missouri, for this research project. The experimental data consists of falls and non-falls performed by three stunt actors and collected in a laboratory room. The stunt actors fall on the mattress and generate 120 segments of fall signals. Apart from fall signals, the stunt actors also produce 120 non-fall sounds from different actions. All the fall signals and non-fall signals contain low noise and lasts about 1 second. There are 20
types of falls and 20 types of non-falls, and each is performed in six trials. Table 6.1 [25] gives the description of dataset.

Table 6.1: Description of the Dataset

<table>
<thead>
<tr>
<th>Falls (format: ‘type’-‘trend’)</th>
<th>#files</th>
<th>Non-falls</th>
<th>#files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Balance-Forwards</td>
<td>6</td>
<td>Closing window</td>
<td>6</td>
</tr>
<tr>
<td>Balance-Backwards</td>
<td>6</td>
<td>Typing keyboard</td>
<td>6</td>
</tr>
<tr>
<td>Balance-Left</td>
<td>6</td>
<td>Key shaking</td>
<td>6</td>
</tr>
<tr>
<td>Balance-Right</td>
<td>6</td>
<td>Machine noise</td>
<td>6</td>
</tr>
<tr>
<td>Loose consciousness-Forwards</td>
<td>6</td>
<td>Phoneringing</td>
<td>6</td>
</tr>
<tr>
<td>Loose consciousness-Backwards</td>
<td>6</td>
<td>Knocking door</td>
<td>6</td>
</tr>
<tr>
<td>Loose consciousness-Left</td>
<td>6</td>
<td>Talking</td>
<td>6</td>
</tr>
<tr>
<td>Loose consciousness-Right</td>
<td>6</td>
<td>Lying on a bed</td>
<td>6</td>
</tr>
<tr>
<td>Trip and fall-Forwards</td>
<td>6</td>
<td>Lying on a couch</td>
<td>6</td>
</tr>
<tr>
<td>Trip and fall-Sideways</td>
<td>6</td>
<td>Sitting on a chair</td>
<td>6</td>
</tr>
<tr>
<td>Slip and fall-Forwards</td>
<td>6</td>
<td>Normal walking</td>
<td>6</td>
</tr>
<tr>
<td>Slip and fall-Sideways</td>
<td>6</td>
<td>Slow walking</td>
<td>6</td>
</tr>
<tr>
<td>Slip and fall-Backwards</td>
<td>6</td>
<td>Rubbing</td>
<td>6</td>
</tr>
<tr>
<td>Reach fall (chair)-Forwards</td>
<td>6</td>
<td>Dropping books</td>
<td>6</td>
</tr>
<tr>
<td>Reach fall (chair)-Left</td>
<td>6</td>
<td>Dropping balls</td>
<td>6</td>
</tr>
<tr>
<td>Reach fall (chair)-Right</td>
<td>6</td>
<td>Dropping cans</td>
<td>6</td>
</tr>
<tr>
<td>Slide fall-Forwards</td>
<td>6</td>
<td>Dropping wood cases</td>
<td>6</td>
</tr>
<tr>
<td>Slide fall-Backwards</td>
<td>6</td>
<td>Dropping plastic bottles</td>
<td>6</td>
</tr>
<tr>
<td>Couch fall-Upper body first</td>
<td>6</td>
<td>Rolling a can</td>
<td>6</td>
</tr>
<tr>
<td>Couch fall-Hips first</td>
<td>6</td>
<td>Rolling a plastic bottle</td>
<td>6</td>
</tr>
<tr>
<td>Total # fall files</td>
<td>120</td>
<td>Total # non fall files</td>
<td>120</td>
</tr>
</tbody>
</table>

6.2 Data simulation

A linear microphone array is implemented to enhance the fall signal. The direction of arrival (DOA) for the $i^{th}$ fall signal is $\theta_i$ and the corresponding DOA for the interference is $\Theta_i$. In the experiment, a 10-second long TV audio signal is prerecorded to simulate the interference. The zeros are padded to the fall signal and non-fall signal
to extend them to 10-second to accommodate the interference. Then the fall signals always reside in the middle of the processing window. The single interference is implemented in the experiment where the DOA of the fall signal or non-fall signal is the same as the that of the interference which means that $\theta_i = \Theta_i$. The procedure to simulate the experiment is shown below:

1. Set the Microphone 0 as the reference channel. Specify the DOAs for the $i^{th}$ fall signal and the non-fall signal. Scale the amplitude of the TV audio signal based on the signal-to-interference (SIR). Add the White Gaussian noise (WGN) to simulate the background noise according to the specific signal-to-noise (SNR).

2. Correct the phase of the fall signal, non-fall signal for all other channels.

### 6.3 Experiment Configuration

In the experiment, 120 fall signals and 120 non-fall signals are simulated with interference and noise in a multichannel environment at the 16KHz sampling frequency to evaluate the performance of the fall detection. To generate the matrix $V$ for NMF and sparse NMF to decompose, all the signals are segmented into several frames whose length is 1024 with the 50% overlap ratio. The number of frames is calculated by the Eq.5.1, which is 311. Then the magnitude of STFT is calculated on each frame. Because of the symmetry, only the upper part is required to decomposition and the size of matrix $V$ is $513 \times 311$. To achieve certain approximation accuracy, the number of basis $K$ for NMF or sparse NMF is chosen as 60 according to the experiment analysis. After the basis separation, the factorized matrix $W_f, H_f$ for the estimated fall signal is generated, and its magnitude STFT is produced by $V_f = W_f H_f$. After
the overlap-add IFFT, the MFCCs features are extracted from the time domain fall signal \( s_f \) to do classification. The estimated fall signal is segmented into multiple frames of 25 ms with overlap ratio of 50% of the frame size. A bank of 30 triangular bandpass filters is implemented, and the number of MFCCs is 6. Then size of the MFCCs feature matrix for classification is \( 6 \times 799 \). The nearest neighbor classifier is implemented for classification. It is a special case for the kNN where \( k=1 \). The Frobenius norm is used as the measurement of the distance between a pair of the MFCCs matrix. The hypotheses of a testing sample is selected as the fall signal if

\[
\frac{\min(dist_{test-\text{nonfall}})}{\min(dist_{test-fall})} > \tau
\]

where \( \tau \) is the decision threshold and \( \min(dist_{test-\text{nonfall}}) \) stands for the minimum Frobenius norm between the feature matrix of a testing sample and the feature matrix of the non-fall training set.

### 6.4 Performance Evaluation

The performance of the NMF based fall detection is evaluated by the receiver operating characteristic (ROC) curve, which is a graphical illustration of the performance of a binary classifier. The curve is generated by plotting the detection rate (or known as sensitivity, recall) against the false alarm rate (also known as fall-out, 1-specificity) at different decision threshold. The detection rate is defined as \( \frac{TP}{TP+FN} \) and the false alarm rate is defined as \( \frac{FP}{TN+FP} \). The 10-fold cross validation is implemented to generate the ROC curve. The 120 fall and 120 non-fall are separated into 10 folds evenly. Given a decision threshold, one fold, which contains 12 fall signals and 12 non-fall
signals, is used for testing while other folds are used for training. By repeating this process, detection rate and false alarm rate is calculated over each validation. Taking the average of these values over 10 folds yields the detection rate and false alarm rate for a specific decision threshold at certain noise pattern. To reduce the randomness introduced by the noise, we change the noise and run 5 trials to get the detection rate and false alarm rate for different noise pattern. Averaging them yields the detection rate and false alarm rate at certain decision threshold. Changing the decision threshold generates the ROC curve. To quantity the ROC curve, area under the curve (AUC) is implemented. AUC is between 0 and 1. The larger the AUC, the better classifier works.

6.5 Experiment Result and Analysis

The purpose of this experiment is to validate the advantage of the sparse based source separation over the NMF based source separation for enhancing the fall signal by reducing the interference and noise. The fall detection with single channel, 2-channel, 3-channel and 4-channel using NMF based source separation and sparse NMF based source separation are evaluated at different SNR with a fixed SIR. The simulated data discussed in the previous section is used to evaluated the performance. The SIR is set as 5dB and SNR are specified as ‘clean’, 5dB, 0dB, -5dB and -10dB. By performing the 10 fold cross validation on the dataset, the ROC curves are generated.
Figure 6.1: 10-fold cross validation ROC curves of fall detection using NMF and sparse with single channel, 2-channel, 3-channel and 4-channel when SIR=5dB and SNR='clean'
Figure 6.2: 10-fold cross validation ROC curves of fall detection using NMF and sparse with single channel, 2-channel, 3-channel and 4-channel when SIR=5dB and SNR=5dB
Figure 6.3: 10-fold cross validation ROC curves of fall detection using NMF and sparse with single channel, 2-channel, 3-channel and 4-channel when SIR=5dB and SNR=0dB
Figure 6.4: 10-fold cross validation ROC curves of fall detection using NMF and sparse with single channel, 2-channel, 3-channel and 4-channel when SIR=5dB and SNR=-5dB
Figure 6.5: 10-fold cross validation ROC curves of fall detection using NMF and sparse with single channel, 2-channel, 3-channel and 4-channel when SIR=5dB and SNR=-10dB

Table 6.2: 10-fold cross validation result for single channel, 2-channel, 3-channel and 4 channel NMF based source separation at different SNR when SIR=5dB

<table>
<thead>
<tr>
<th>SNR</th>
<th>Number of Channels</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>'Clean'</td>
<td></td>
<td>0.987</td>
<td>0.995</td>
<td>0.997</td>
<td>0.999</td>
</tr>
<tr>
<td>5dB</td>
<td></td>
<td>0.951</td>
<td>0.971</td>
<td>0.977</td>
<td>0.980</td>
</tr>
<tr>
<td>0dB</td>
<td></td>
<td>0.900</td>
<td>0.919</td>
<td>0.925</td>
<td>0.946</td>
</tr>
<tr>
<td>-5dB</td>
<td></td>
<td>0.826</td>
<td>0.864</td>
<td>0.877</td>
<td>0.884</td>
</tr>
<tr>
<td>-10dB</td>
<td></td>
<td>0.814</td>
<td>0.818</td>
<td>0.864</td>
<td>0.877</td>
</tr>
</tbody>
</table>
Table 6.3: 10-fold cross validation result for single channel, 2-channel, 3-channel and 4 channel sparse NMF based source separation at different SNR when SIR=5dB

<table>
<thead>
<tr>
<th>SNR</th>
<th>Number of Channels</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>'Clean'</td>
<td>0.993</td>
</tr>
<tr>
<td>5dB</td>
<td>0.991</td>
</tr>
<tr>
<td>0dB</td>
<td>0.979</td>
</tr>
<tr>
<td>-5dB</td>
<td>0.979</td>
</tr>
<tr>
<td>-10dB</td>
<td>0.958</td>
</tr>
</tbody>
</table>

Fig.6.1 to Fig.6.5 are the ROC curves when SIR=5dB and SNR changes from 'Clean' to -10dB at an interval of 5dB using NMF based source separation and sparse NMF based source separation. These figures show that the proposed sparse NMF based source separation algorithm achieves better performance at each SNR level and for each method multichannel result is always better than the single channel result, which indicates that beamforming can help reduce the interference and the noise. Table 6.2 and Table 6.3 tabulates the area under the curve from the ROC for different SNR level. Notice that the AUC for the NMF and sparse NMF are similar in a low noise environment. However, when the interference is moderate and background noise is large, the result of the NMF and sparse NMF have great difference. This is because sparse NMF can decompose the original data matrix into meaningful parts that can be used as the building blocks of the data and even at high noise environment it still can separate the fall signal from the noise-plus-interference. However for NMF, some of the factorized bases contains both fall and noise-plus-interference components, which loses some of the spectrum information of the fall signal.
6.6 Summary

In this chapter, we describe the detailed information about the fall detection application using both NMF based source separation and sparse NMF based source separation including the data measurement, data simulation, experiment configuration and the final result. The final result reveals that the proposed NMF based and sparse NMF based source separation are quite effective. Both approaches can achieve good performance when the noise is not large. Notice that the multichannel source separation is more robust to the noise compared with the signal channel method. It is also worth to mention that sparse NMF based method always yields better result than the NMF based method and the improvement is even more significant when we have a high noise environment.
Chapter 7

Conclusion

Since nonnegativity is an important property, we would like to obtain it during data processing. The objective is to find approximation algorithm which yields good approximation accuracy while keeping the physical representation of the data. Then NMF is a good starting point.

In chapter two, we introduce the basic concept about the NMF and three algorithms, multiplicative update, gradient descent and alternating least square, to address the matrix updating problem. The multiplicative update is the most famous algorithm for NMF. The numerical experiment reveals that it is easy to implement and generates the best result among three of them although the convergence property is hard to prove analytically. The gradient descent method is slow to converge and whether it can converge or not greatly depends on the choice of the learning rate $\alpha$. For alternating least square, there are two versions, the exact one and the inexact one. The exact one can converge to a stationary point, but the speed of this method is too slow to accept. However the inexact one is fast to run, it does not always make
a descent update.

Inspired by the success of the sparse coding, we introduce the sparsity constraint into the NMF. In chapter three, we introduce the basic concept about the sparse NMF and three algorithms gradient descent, alternating least square and the sparsity promoted iterated constrained endmember. All three algorithms achieve sparsity by adding the penalty term into the cost function. The previous two are the extension of the methods proposed in chapter two. The sparsity promoted iterated constrained endmember can find the actual number of bases by pruning the unnecessary bases. Although the sparsity promoted iterated constrained endmember is more computational expensive and the approximation accuracy is not as good as other two methods, it still generates more convincing representation of the original data.

We use audio signal as an example to illustrate classification. In chapter four, we introduce the techniques for classification. The purpose of the classification is to extract features from the original data and use these features to identify the origin of the data. The classifier followed by the feature extraction is the common protocol for classification application. The Mel-frequency Cepstral Coefficients and the k-nearest neighbour implemented in the thesis is presented in this chapter.

In chapter five, we introduce the source separation using NMF and sparse NMF for signal channel application and multichannel application. The purpose of the source separation is to recover the individual signal from a mixed one. The source separation is based on the Euclidean norms of the rows of $\mathbf{H}$. The simulation on the fall signal separation reveals that sparse NMF based source separation achieves better result since sparse NMF provide better factorization that can be used to reconstruct the original data with good approximation accuracy.
Chapter six contains detailed description of the fall detection application including the data measurement, data simulation, experiment configuration and the final result. The final result reveals that the both NMF based and sparse NMF based source separation can achieve good performance when the noise is not large. The multichannel source separation is more robust to the noise compared with the signal channel method. Moreover, sparse NMF based method always yields better result and the improvement is more obvious when the SNR becomes small.

For NMF based source separation or sparse NMF based source separation, there are two steps that are important to the final performance. The first one is how to factorize the original data so that the bases don’t contain the mixture of different components. The second one is how to separate the bases into the right category so that each class just include one types of basis. As is shown in the Fig.5.7 both methods are not perfect. For NMF some of the fall signal leaks into the interference-plus-noise one. However, for sparse NMF, the fall signal contains the interference-plus-noise when fall occurs.

Then future works will focus on both of these issues. For sparse NMF, the result depends on the choice of the sparsity constant \( \Gamma \) and the regularization parameter \( \mu \). How to choose these parameters automatically remains to be an open question and ultimately would affect the factorization result. We would also focus on improving the basis separation algorithm to reduce the influence caused by the shared bases. More real world data will be collect to test the performance and robustness of the approach.
Bibliography


[6] Cyril Joder, Felix Weninger, Florian Eyben, David Virette, and Bjorn Schuller. Real-time speech separation by semi-supervised nonnegative matrix factoriza-


[43] Christos Boutsidis and Efstratios Gallopoulos. Svd based initialization: A head
start for nonnegative matrix factorization. Pattern Recognition, 41(4):1350–1362,
2008.

[44] Xiaoqiang Lu, Hao Wu, Yuan Yuan, Pingkun Yan, and Xuelong Li. Manifold reg-
ularized sparse nmf for hyperspectral unmixing. Geoscience and Remote Sensing,

[45] Giovanni Costantini, Massimiliano Todisco, and Giovanni Saggio. A sensor in-
terface based on sparse nmf for piano musical transcription. In Advances in
Sensors and Interfaces (IWASI), 2011 4th IEEE International Workshop on,

[46] Arshia Cont. Realtime multiple pitch observation using sparse non-negative con-
straints. In International Symposium on Music Information Retrieval (ISMIR),
2006.

[47] Ivan Ivek. Supervised dictionary learning by a variational bayesian group sparse

[48] Suvrit Sra and Inderjit S Dhillon. Generalized nonnegative matrix approxima-
tions with bregman divergences. In Advances in neural information processing

[49] Karthik Devarajan and N Ebrahimi. Molecular pattern discovery using nonneg-
native matrix factorization based on renyis information measure. In Proceedings
of the XII SCMA International Conference, volume 24, 2005.


[57] Mark Berman, Harri Kiiveri, Ryan Lagerstrom, Andreas Ernst, Rob Dunne, and Jonathan F Huntington. Ice: A statistical approach to identifying endmem-


