# Advances in Nonnegative Matrix Factorization with Application on Data Clustering

Submitted by

Jing Wang

for the degree of Doctor of Philosophy

of the

Bournemouth University

15, September, 2017

# Abstract

Clustering is an important direction in many fields, e.g., machine learning, data mining and computer vision. It aims to divide data into groups (clusters) for the purposes of summarization or improved understanding. With the rapid development of new technology, high-dimensional data become very common in many real world applications, such as satellite returned large number of images, robot received real-time video streaming, large-scale text database and the mass of information on the social networks (i.e., Facebook, twitter), etc, however, most existing clustering approaches are heavily restricted by the large number of features, and tend to be inefficient and even infeasible. In this thesis, we focus on finding an optimal low dimensional representation of high-dimensional data, based nonnegative matrix factorization (NMF) framework, for better clustering. Specifically, there are three methods as follows:

- **Multiple Components Based Representation Learning**

Real data are usually complex and contain various components. For example, face images have expressions and genders. Each component mainly reflects one aspect of data and provides information others do not have. Therefore, exploring the semantic information of multiple components as well as the diversity among them is of great benefit to understand data comprehensively and in-depth. To this end, we propose a novel multi-component nonnegative matrix factorization. Instead of seeking for only one representation of data, our approach learns multiple representations simultaneously, with the help of the Hilbert Schmidt Independence Criterion (HSIC) as a diversity term. HSIC explores the diverse information among the representations, where each representation corresponds to a component. By integrating the multiple representations, a more comprehensive representation is then established. Extensive experimental results on real-world datasets have shown that MCNMF not only achieves more accurate performance over the state-of-the-arts using the aggregated representation, but also interprets data from different aspects with the multiple representations, which is beyond what current NMFs can offer.

- **Ordered Structure Preserving Representation Learning**

Real-world applications often process data, such as motion sequences and video clips, are with ordered structure, i.e., consecutive neigh-bouring data samples are very likely share similar features unless a sudden change occurs. Therefore, traditional NMF assumes the data samples and features to be independently distributed, making it not proper for the analysis of such data. To overcome this limitation, a novel NMF approach is proposed to take full advantage of the ordered nature embedded in the sequential data to improve the accuracy of data representation. With a $L_{2,1}$-norm based neighbour penalty term, ORNMF enforces the similarity of neighbouring data. ORNMF also adopts the $L_{2,1}$-norm based loss function to improve its robustness against noises and outliers. Moreover, ORNMF can find the cluster boundaries and get the number of clusters without the number of clusters to be given beforehand. A new iterative updating optimization algorithm is derived to solve ORNMF's objective function. The proofs of the convergence and correctness of the scheme are also presented. Experiments on both synthetic and real-world datasets have demonstrated the effectiveness of ORNMF.

- **Diversity Enhanced Multi-view Representation Learning**

Multi-view learning aims to explore the correlations of different information, such as different features or modalities to boost the performance of data analysis. Multi-view data are very common in many real world applications because data is often collected from diverse domains or obtained from different feature extractors. For example, color and texture information can be utilized as different kinds of features in images and videos. Web pages are also able to be represented using the multi-view features based on text and hyperlinks. Taken alone, these views will often be deficient or incomplete because different views describe distinct perspectives of data. Therefore, we propose a Diverse Multi-view NMF approach to explore diverse information among multi-view representations for more comprehensive learning. With a novel diversity regularization term, DiNMF explicitly enforces the orthogonality of different data representations. Importantly, DiNMF converges linearly and scales well with large-scale data. By taking into account the manifold structures, we further extend the approach under a graph-based model to preserve the locally geometrical structure of the manifolds for multi-view setting. Compared to other multi-view NMF methods, the enhanced diversity of

both approaches reduce the redundancy between the multi-view representations, and improve the accuracy of the clustering results.

- **Constrained Multi-View Representation Learning**

To incorporate prior information for learning accurately, we propose a novel semi-supervised multi-view NMF approach, which considers both the label constraints as well as the multi-view consistence simultaneously. In particular, the approach guarantees that data sharing the same label will have the same new representation and be mapped into the same class in the low-dimensional space regardless whether they come from the same view. Moreover, different from current NMF-based multi-view clustering methods that require the weight factor of each view to be specified individually, we introduce a single parameter to control the distribution of weighting factors for NMF-based multi-view clustering. Consequently, the weight factor of each view can be assigned automatically depending on the dissimilarity between each new representation matrix and the consensus matrix. Besides, Using the structured sparsity-inducing, $L_{2,1}$-norm, our method is robust against noises and hence can achieve more stable clustering results.

**Keywords**: Nonnegative matrix factorization, representation learning, low rank, multi-component, multi-view, constraint

*To my parents*

# Declaration

I, Jing Wang, declare that the work contained in this thesis has not been submitted for any other award and that it is all my own work. I also confirm that this work fully acknowledges opinions, ideas and contributions from the work of others.

Jing Wang
15, September, 2017

# Acknowledgement

The doctoral research is coming to an end. Looking back over the last three years, my greatest achievement is not limited to the experience of scientific research and academic outcomes, more importantly, the good friendship with my supervisors and classmates becomes a beautiful unforgettable experience in my life. Your excellence and hard working makes me deeply aware of my own shortcomings and inspires me never slack off. Your kindness and sincerity gives me endless support and power, making me feel fearless and courageous. I would like to take this opportunity to express my great thanks to all those who have helped for me.

First of all, I would like to thank my first supervisor Prof. Feng Tian, who showed me the path to the research area of computer vision and machine learning. Prof. Tian's rigorous research attitude and profound knowledge gives me great help, so that I can participate in high-level scientific research work. He also provides me a good research platform and environment, so that I can publish papers and complete the doctoral thesis successfully. Moreover, he has always been providing me insightful advises and valuable ideas, and shaping me to be a qualified researcher.

I would like to thank my second supervisor Prof. Changhong Liu and third supervisor Dr. Hongchuan Yu. Their supports and advices have guided me through the most difficult moments of my research. Prof. Liu has given a lot of valuable suggestions on my research and helped me correcting my paper over and over again even for the details of changes. His serious attitude of research inspires my enthusiasm for my future research. Dr. Yu has a keen insight and a profound understanding of research, which pointed out the direction of my future research. He also encourages me to communicate with peers, which not only establish my research self-confidence, but also greatly enriches my vision.

I would like to thank Dr. Xiao Wang for collaboration. We have always communicated, discussed and exchanged ideas in matrix factorization issues for three years. He have given me a very patient and detailed guidance for my research, so that I can get deeper understanding of the research topics and propose more novel research methods. We have conquered several research problems together.

I would like to thank my classmate Chi Zhang. We always have discussion, inspire each other and progress together. Her high motivated research attitude influences me greatly. Thank all my friends in BU, we often have meals, do excises and chat together. Thanks for their accompanies, my life beyond the research is more joyful and colorful.

I would also like to thank Dr. Changqing Zhang, Dr. Kun Zhan and Dr. Liang Yang for their kind academic guidance during my doctoral study.

Finally, I would like to thank my parents. Thanks to their unconditional support and constant encouragement, so that I have passion and motivation to continue my doctoral study. Thanks to my grandfather, as an older generation of educator, his diligence, honesty and integrity has a tremendous impact on me.

# Publications

## A. Conference paper

- **Jing Wang**, Feng Tian, Xiao Wang, Chang Hong Liu, Hongchuan Yu, Liang Yang. Multi-component Nonnegative Matrix Factorization[C]. The 26th International Joint Conference on Artificial Intelligence (IJCAI) 2017.

- **Jing Wang**, Feng Tian, Xiao Wang, Chang Hong Liu, Hongchuan Yu, Xianchao Tang. Robust Nonnegative Matrix Factorization with Ordered Structure Constraints[C]. The 30th International Joint Conference on Neural Networks (IJCNN) 2017.

- **Jing Wang**, Xiao Wang, Feng Tian, Chang Hong Liu, Hongchuan Yu, Yanbei Liu. Adaptive Multi-View Semi-Supervised Nonnegative Matrix Factorization[C]. The 23rd International Conference on Neural Information Processing (ICONIP) 2016.

- **Jing Wang**, Feng Tian, Changhong Liu, Xiao Wang. Robust Semi-supervised Nonnegative Matrix Factorization[C]. The 28th International Joint Conference on Neural Networks (IJCNN) 2015.

- Xiao Wang, Peng Cui, **Jing Wang**, Jian Pei, Wenwu Zhu, Shiqiang Yang. Community Preserving Network Embedding[C]. The 31st AAAI Conference on Artificial Intelligence (AAAI) 2017.

## B. Journal paper

- **Jing Wang**, Feng Tian, Hongchuan Yu, Chang Hong Liu, Kun Zhan, Xiao Wang. Diverse Nonnegative Matrix Factorization for Multi-view Data Representation[J]. IEEE Transactions on Cybernetics (TCYB) 2017.

- **Jing Wang**, Xiao Wang, Feng Tian, Chang Hong Liu, Hongchuan Yu. Constrained Low-Rank Representation for Robust Subspace Clustering[J]. IEEE Transactions on Cybernetics (TCYB) 2016.

- Kun Zhan, Jinhui Shi, **Jing Wang***, Feng Tian. Graph-regularized concept factorization for multi-view document clustering[J]. Journal of Visual Communication and Image Representation (JVCIR) 2017.

- Yanbei Liu, Kaihua Liu, Changqing Zhang, **Jing Wang**, Xiao Wang. Unsupervised Feature Selection via Diversity-induced Self-representation[J]. Neurocomputing 2016.

- Xianchao Tang, Tao Xu, Xia Feng, Guoqing Yang, **Jing Wang**, Qiannan Li, Yanbei Liu, Xiao Wang. Learning Community Structures: Global and Local Perspectives[J]. Neurocomputing 2017.

- Liang Yang, Meng Ge, Di Jin, Dongxiao He, Huazhu Fu, **Jing Wang**, Xiaochun Cao, Advertisement Exploring the roles of cannot-link constraint in community detection via Multi-variance Mixed Gaussian Generative Model[J]. PloS One 2017.

# Contents

# List of Figures

# List of Tables

# List of Notation

| | |
|---|---|
| $\mathbf{X}, x_i$ | input data matrix, data vector |
| $\mathbf{X}^{(v)}, \mathbf{x}_i^{(v)}$ | input data matrix, data vector of $v$th view |
| $\mathbf{W}$ | basis matrix |
| $\mathbf{H}$ | representation matrix |
| $\mathbf{A}$ | label constraint matrix |
| $\mathbf{R}$ | ordered constraint matrix |
| $tr$ | trace form |
| $\boldsymbol{\eta}$ | Lagrange multiplier matrix |
| $\alpha$,$\beta$,$\gamma$ | tradeoff parameter |
| $m, n$ | dimensionality, number of samples |
| $k$ | number of clusters |
| $\|\cdot\|_F, \|\cdot\|_{2,1}$ | Frobenius norm, $L_{2,1}$-norm |
| $\circ$ | Kronecker (element-wise) product |
| $\odot$ | dot product |
| $J$ | objective function |
| $V$ | number of components/views |

# Chapter 1

# Introduction

## 1.1 Research Background

In the past decade, we have witnessed the explosion and the messiness of data across numerous fields of pattern recognition, computer vision and machine learning. With the fast-growing amount of data, partitioning them into different groups or clusters is of great practical importance to understand data in-depth. For relatively small collections, it may be possible to partition them manually. But given large volumes of data, it would be extremely time consuming and difficult to partition them into different meaningful groups. Thus, clustering which aims to group data automatically so that data in the same group are with high similarity, is becoming one of the most important techniques in data analysis [26, 1]. As an unsupervised technique, clustering which needs no annotation of training data but considers nature of data only has been widely used in a wide range of applications [52, 89, 104]. For example, in business analysis, clustering customers can characterize features of different groups for targeted marketing; in text mining, clustering documents into specific categories can form semantic topics. In genomic analysis and cancer study, clustering can find common patterns in the patients' gene expression profiles that correspond to cancer subtypes and offer personalized treatments. Besides, clustering can be used as a foundation for many directions in computer science. Such as in anomaly detection, clustering can find the exception points that are not related to each category [57]; in dictionary-based expression learning, clustering can find class centers to build dictionaries [83, 92]. Overall, clustering is crucial in the field of data mining, machine learning, pattern recognition, computer vision, etc.

In reality, high dimensional data has become very common. For example, an image dataset may contain a huge number of pixels that correspond to dimensions; a document consists of a sequence of words, each of which can be regarded as a dimension; a gene expression microarray may have thousands of dimensions and each of them corresponds to an experimental condition, etc. Traditional clustering approaches which calculate similarity between high-dimensional data samples directly to perform results, tend to be inefficient and even infeasible, because the results are greatly affected by noises and may not be robust [23]. Therefore, it is more reasonable and effective to reduce the dimensions of original data for clustering, so that noisy data and redundancy of data can be alleviated. To do so, quite a few matrix factorization techniques [23, 30, 44, 90, 64, 62, 108, 17] have been proposed to factorize data from the input space to several low-dimensional matrices. The most popular methods include Principal Component Analysis (PCA) [45], Singular Value Decomposition (SVD) [23] and Vector Quantization [31]. However, the factorizing matrices in these methods can have negative entries, which makes it hard or impossible to obtain physical interpretations from the factorizing results. This is because many real-world data (e.g. images and texts) are nonnegative and the corresponding hidden parts convey physical meanings only when the nonnegative condition holds. These methods are therefore called "holistic" approaches. In contrast, Nonnegative Matrix Factorization (NMF) [55] as a "parts-based" approach, has been receiving more and more attention. It imposes the nonnegativity constraint on all the factorizing matrices, allowing only additive but not subtractive combinations during the factorization. Such nature can exactly discover the hidden parts that have specific structures and physical meanings, such as each original face image can be approximately represented by additively combining several "parts" (eyes, noses, lips, etc.).

To summarize, finding a useful low-dimensional representation to achieve satisfactory clustering performance based on the NMF framework is of great significance. Based on previous works, this dissertation improves NMF from three aspects, i.e., extraction of semantic information, exploration of diverse feature information, as well as constraints fusion, and conducts clustering on the proposed methods with a wide range of real-world datasets including images, network, texts and video sequences.

## 1.2 Contributions of the Thesis

This thesis studies NMF from both theoretical analysis and application aspects, as shown in figure 3.2. In theory, different from existing single-view NMFs that find a single representation based on global feature only, we explore embedded multiple components of data with capturing more comprehensive information and interpreting data from different perspectives at a sematic level. Considering that sequential data contain ordered nature, we also incorporate such nature to enhance clustering performance. Since data often consist of multiple features which describe data from different perspectives, we further study NMF in multi-view setting by feature fusion. Instead of learning a consensus representation across different views as existing multi-view NMFs, we emphasize the diversity of each view so as to capture more comprehensive information among views. Finally, we incorporate prior information in multi-view clustering effectively to guide the clustering process. From application perspectives, the study of single-view NMF is based on image or video sequence clustering, while we explored multi-view NMF on a wider range of data, including images, texts and networks. In more details, my main research work include the following three works:

1. **Multiple Components-based NMF**

Given that data contain different subsets of features, namely components, we obtain corresponding representations and achieve multiple clustering results. A diversity regularization is introduced to enhance the independency between different representations. By integrating different representations, a comprehensive representation with diverse information is established. The main contributions are

- This work is the first to explore components of data and achieve multiple representations where each representation corresponds to a component.
- A diverse term is introduced to explore the diverse information among the representations so as to capture comprehensive information.
- A novel multiplicative updating rule is derived to solve the objective function, along with its convergence proof correctness analysis.
- We conduct clustering experiments on image datasets. The results have demonstrated that the proposed approach exploits semantic meaning of

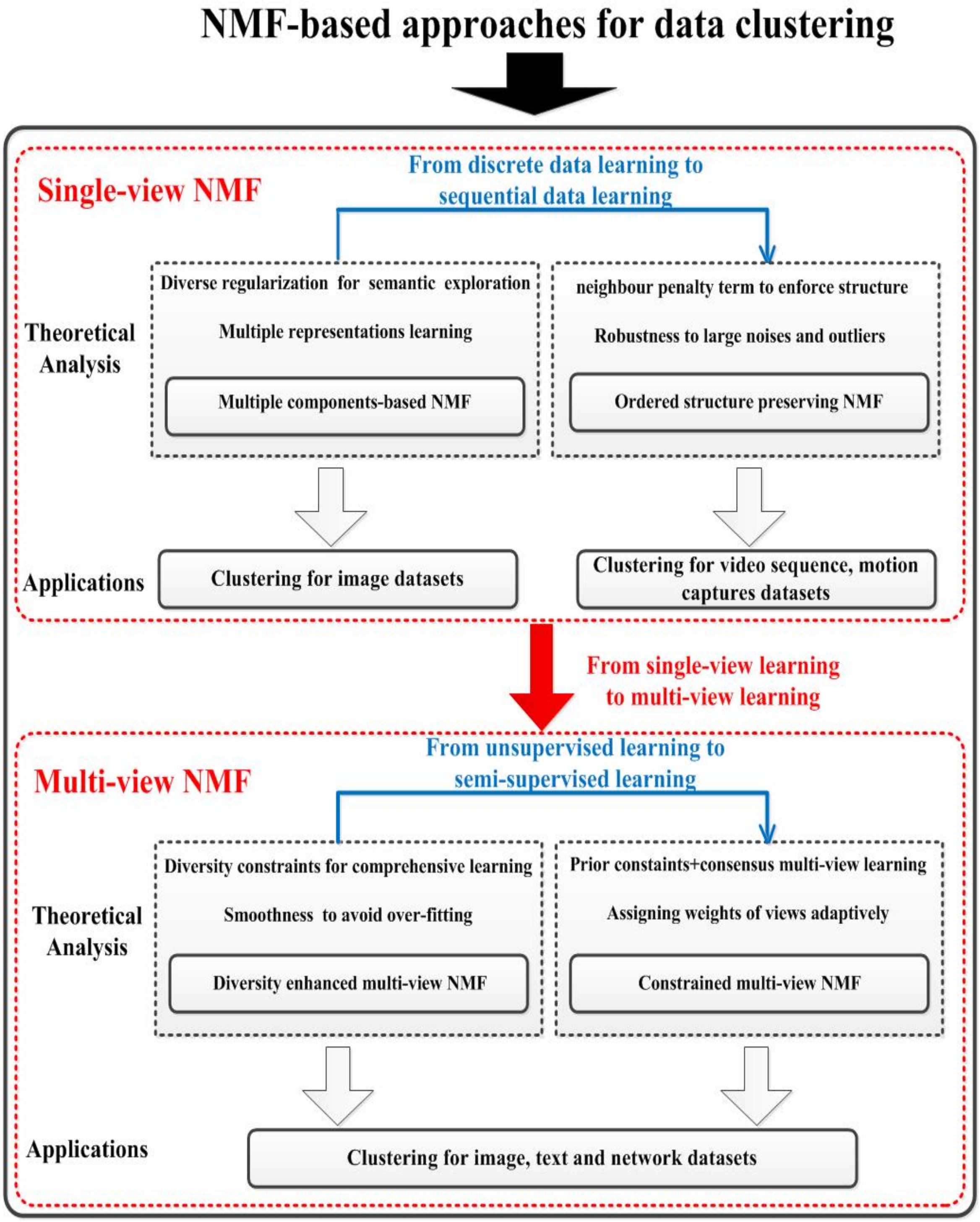

NMF-based approaches for data clustering
Single-view NMF
From discrete data learning to sequential data learning
Theoretical Analysis
Diverse regularization for semantic exploration
Multiple representations learning
Multiple components-based NMF
neighbour penalty term to enforce structure
Robustness to large noises and outliers
Ordered structure preserving NMF
Applications
Clustering for image datasets
Clustering for video sequence, motion captures datasets
From single-view learning to multi-view learning
Multi-view NMF
From unsupervised learning to semi-supervised learning
Theoretical Analysis
Diversity constraints for comprehensive learning
Smoothness to avoid over-fitting
Diversity enhanced multi-view NMF
Prior constaints+consensus multi-view learning
Assigning weights of views adaptively
Constrained multi-view NMF
Applications
Clustering for image, text and network datasets

Figure 1.1: The framework of the thesis.

data by interpreting data from different aspects with the multiple representations, which is beyond what current NMFs can offer.

**2. Ordered Structure Preserved NMF**

Different from existing NMFs assume the data samples and features to be independently distributed, the approach captures ordered nature embedded in the sequential data, such as video sequences. Based on a novel neighbour penalty term, it enforces the similarity of neighbouring data to improve the discriminating power of the data representations. The main contributions are

- With a novel neighbour penalty term, it enforce the similarity of the consecutive data representations by incorporating the ordered structure as additional constraints.
- In ideal cases, it can correctly find the cluster boundaries and get the number of clusters without needing the number of clusters beforehand.
- A $L_{2,1}$-norm based loss function is adopted to improve its robustness against noises and outliers.
- A new iterative updating optimization scheme is derived to solve ORNMF's objective function, along with its convergence and correctness proofs.

**3. Diversity Enhanced Multi-View NMF**

With a novel penalty term, the information among different views are constrained to be diverse enough to each other, thus the mutually redundant information among views are reduced. The proposed approach can scales well with large-scale data due to its linear computational time. The main contributions are

- A novel diversity regularization term is proposed to enforce the orthogonality of different data representations, so that the diversity among views are enhanced and mutually redundancy are reduced.
- The proposed approach is computationally linear thus has good scalability to large-scale datasets.
- By taking into account manifold structures of data in each view, we further extend the proposed approach by incorporating local geometry information which leads to further improved performances.

4. **Constrained Multi-View NMF**

The approach takes both prior information and the consistency of multiple views into account. It learns a consensus representation across multiple views jointly with a constrained label matrix. Moreover, the weights of representations which reflect the importance of different views are learnt adaptively. The main contributions are

- Incorporating label information as hard constraints to enhance the discriminating power of new representations so that all data with the same label are clustered together regardless of their views.
- A single parameter is introduced to learn the weight of each view adaptively. Each weight is assigned automatically depending on the dissimilarity between each new representation matrix and the consensus matrix.
- Using $L_{2,1}$-norm to measure the approximation errors, it is robust against noises and hence can achieve more stable clustering results.
- The clustering experiments are conducted in well-known real-world datasets, which have demonstrated its effectiveness and robustness in comparison to the state-of-the-arts.

## 1.3 Thesis Outline

- Chapter 1 introduces the research background and the main contributions of this thesis.
- Chapter 2 focuses on unsupervised single-view representation learning. We present a novel multi-component nonnegative matrix factorization approach, which interprets data from different aspects through seeking for multiple representations simultaneously.
- Chapter 3 focuses on unsupervised single-view representation learning for sequential data. A novel ordered structure preserved nonnegative matrix factorization approach is proposed to enforce the similarity of data presentations.

- Chapter 4 focuses on unsupervised multi-view representation learning. We propose a diversity enhanced approach to enhance diverse information and reduce the redundancy among multi-view representations for more comprehensive learning.
- Chapter 5 focuses on semi-supervised multi-view representation learning. We propose a constrained approach, which considers both the label information as well as the multi-view consistence simultaneously.
- Chapter 6 summarizes the work of this thesis and discusses future works of nonnegative representation learning.

# Chapter 2

# Nonnegative Matrix Factorization

Nonnegative Matrix Factorization (NMF) is a popular matrix decomposition method with various applications in e.g. machine learning, data mining, pattern recognition, and signal processing. The nonnegativity constraints have been shown to result in parts-based representation of the data, and such additive property can lead to the discovery of datas hidden structures that have meaningful interpretations. In this chapter, we will first give a brief introduction to NMF before reviewing NMF-based approaches.

## 2.1 Introduction

Nonnegative Matrix Factorization was first introduced by Paatero and Tapper [81], and gained popularity by the works of Lee and Seung [55] published in Nature in 1999. Mathematically, given a $n$ data matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n] \in \mathbb{R}_+^{m \times n}$ where each column is a $m$-dimensional data vector, NMF [55] aims to find two nonnegative matrices $\mathbf{W} \in \mathbb{R}_+^{m \times k}$ and $\mathbf{H} \in \mathbb{R}_+^{k \times n}$, where the product of the two matrices can well approximate the original matrix, represented as

$$\mathbf{X} \approx \mathbf{WH}. \tag{2.1}$$

Here $\mathbb{R}_+$ denotes the set of nonnegative real numbers. The columns of $\mathbf{W}$ form a basis of a latent space and are called basis vectors. The representation matrix $\mathbf{H}$

contains coefficients that reconstruct the input matrix $\mathbf{X}$ by linear combinations of the basis vectors, and the product term $\mathbf{WH}$ is called the compressed version of the $\mathbf{X}$ or the approximating matrix of $\mathbf{X}$. Typically we have $k \ll \min(m, n)$, i.e., the original data points in the $m$-dimensional space are reduced to a much lower-dimensional space of dimension $k$. An appropriate selection of the value $k$ is critical in practice, but its choice is usually problem dependent.

There are several ways to quantify the difference between the data matrix $\mathbf{X}$ and $\mathbf{WH}$. But the most used measure is the Frobenius norm, and the objective function of NMF is

$$\min D_F(\mathbf{X}, \mathbf{WH}) = \min_{\mathbf{W},\mathbf{H}\geq 0} \|\mathbf{X} - \mathbf{WH}\|_F^2 \tag{2.2}$$

where $\|\cdot\|_F$ denotes the Frobenius norm and "$\geq 0$" indicates entrywise nonnegativity.

## 2.2 Multiplicative update algorithms

A wide range of numerical optimization algorithms [93, 15, 16] have been proposed for solving (2.2) . Since (2.2) is nonconvex, in general we cannot expect an algorithm to reach the global minimum but local minimum. This can be found by an iterative procedure alternating between updating one matrix while keeping the other one fixed. The pseudo code to do so is given in Algorithm 2.1. Among

**Algorithm 2.1** Pseudo code for the algorithm NMF

```
Initialize $\mathbf{W} \geq 0$ and $\mathbf{H} \geq 0$
repeat
    Updating $\mathbf{H}$ with fixing $\mathbf{W}^{(v)}$,$D_F(\mathbf{X}, \mathbf{WH}^{new}) \leq D_F(\mathbf{X}, \mathbf{WH}^{old})$ and $\mathbf{H}^{new} \geq 0$
    Updating $\mathbf{W}$ with fixing $\mathbf{H}^{(v)}$, $D_F(\mathbf{X}, \mathbf{W}^{new}\mathbf{H}) \leq D_F(\mathbf{X}, \mathbf{W}^{old}\mathbf{H})$ and $\mathbf{W}^{new} \geq 0$
until converges or the maximum number of iterations is reached.
```

the optimization algorithms, the most popular one is the following multiplicative update rules [55] as it consists of basic matrix computations and thus is very simple to implement.

$$\mathbf{H} \leftarrow \mathbf{H} \odot \frac{\mathbf{W}^T\mathbf{X}}{\mathbf{W}^T\mathbf{WH}} \tag{2.3}$$

$$\mathbf{W} \leftarrow \mathbf{W} \odot \frac{\mathbf{X}\mathbf{H}^T}{\mathbf{W}\mathbf{H}\mathbf{H}^T} \tag{2.4}$$

The multiplicative update algorithm is obtained via the gradient descent by choosing a smart step size. Thus, for the purpose of implementation (e.g, using Matlab), a small constant in each update rule is added to the denominator to avoid division by zero. Besides, since the algorithm suffers from getting stuck in local minimum, it is recommended to run the algorithm several times using different initializations. To prove the convergence property of the algorithm, a common strategy is to measure the decrease of the cost function between successive iterations, and the algorithm stops if the decrease falls below a predefined threshold.

## 2.3 Existing NMF-based approaches

There has been several papers [41, 19, 27, 43, 21, 34, 20, 65, 46] extending and improving the original NMF in the past decade and NMF has been successfully applied to many areas such as image processing, face recognition [61], community detection [112]and document clustering [84], [111]. An elaborate review of NM-based approaches can be found in [105]. According to the types of features of data (to be dealt with), current NMFs could be classified into two categories, single-view NMF and multi-view NMF.

### 2.3.1 Single-view NMF

Single-view NMF means that the approaches can deal with one type of features of data only, such as pixels of images or distributions of words in a documents. Various single-view NMFs [67, 115, 6, 27, 51, 64, 48, 54] have been proposed to find an proper low-dimensional representation. These approaches modify the traditional NMF objective function by either using different norms to measure approximation errors or incorporating auxiliary constraints. Though the forms of constraints are application dependent, they can be characterized by (2.2) as follows:

$$\min_{\mathbf{W},\mathbf{H}\geq 0} D(\mathbf{X}, \mathbf{W}\mathbf{H}) + \alpha J_1(\mathbf{W}) + \beta J_2(\mathbf{H}) \tag{2.5}$$

where $D(\mathbf{X}, \mathbf{WH})$ represents the approximation errors. $J_1(\mathbf{W})$ and $J_2(\mathbf{H})$ are penalty terms regarded as auxiliary constraints. The regularization parameters $\alpha$ and $\beta$ balance the trade-off between the approximation errors and the added constraints.

Sparsity is one of important characters of data, that is, all the features or data show the positive effect on the final results but only a few provide meaningful and useful information. Usually, adding sparsity regularization to select the most useful features [41] can improve the generalization of a method, thus avoiding the over-fitting problem [113]. Several approaches are proposed to modify NMF algorithms [40, 24, 41, 47] by penalizing $\mathbf{H}$ or $\mathbf{W}$ which aims to yield a sparse representation [94]. For example, Hoyer[40] proposed a sparseness criterion by leveraging the relationship between the $L_1$ and $L_2$ norm:

$$sparseness(x) = \frac{\sqrt{n} - (\sum |x_i|)/\sqrt{\sum x_i^2}}{\sqrt{n} - 1}, \tag{2.6}$$

where $x$ denotes a given vector with dimension $n$. For instance, the sparseness criterion imposed on a $m \times k$ matrix $\mathbf{W}$ can be formulated as the following penalty term:

$$J_1(\mathbf{W}) = (\alpha \|vec(\mathbf{W})\|_2 - \|vec(\mathbf{W})\|_1) \tag{2.7}$$

where $\alpha = \sqrt{mk} - (\sqrt{mk} - 1)\alpha$ and $vec(\cdot)$ is an operator that transforms a matrix into a vector by stacking its columns. The sparseness in $\mathbf{W}$ is specified by setting $\alpha$ to a value between 0 and 1.

Often, the input $\mathbf{X}$ contains large noises and outliers, which may greatly influence performances. To alleviate this issue, some more robust methods were proposed. For example, Kong et al.[50] adopted $L_{2,1}$-norm based objective function to weaken the influence of data outliers:

$$D(\mathbf{X}, \mathbf{WH}) = \|\mathbf{X} - W\mathbf{H}\|_{2,1}. \tag{2.8}$$

Since the error for each data point is not squared as standard NMF , and thus the large errors due to outliers do not dominate the objective function. Later, Du et al. [22] proposed using the correntropy induced metric to make it insensitive to outliers. Hamza et al. [37] adopted a hypersurface cost function to make the NMF robust to the outliers. Zhang et al. [115] proposed to subtract a sparse outlier matrix from the data matrix to reduce the effect of the outliers. Recent research has shown that data are found to lie on a nonlinear low dimensional manifold embedded in a high dimensional ambient space [2, 85, 60]. However, the standard NMF fails to discover such intrinsic geometrical structure of the data space [6]. To this end, Cai et al. [6] proposed a graph regularized NMF (GNMF) to preserve the local manifold structure with regarding that the data points nearby have more similar data representations than those far away. By constructing a weight matrix $\mathbf{U}$, GNMNF is to solve

$$J_1(\mathbf{H}) = tr(\mathbf{H}\mathbf{L}\mathbf{H}^T), \tag{2.9}$$

where $\mathbf{L} = \mathbf{D}$ - $\mathbf{U}$, $\mathbf{D}$ is a diagonal matrix whose entries $\mathbf{D}_{jj} = \sum_l \mathbf{U}_{jl}$. Based on GNMF, Huang et al. [42] then proposed a robust manifold NMF which simultaneously alleviates noises and preserves geometrical structure. Since the performance of GNMF is known to hinge heavily on the choice of nearest neighbor graph and it is difficult and time consuming to choose a suitable graph. To overcome this limitation, Wang et. al. [100] proposed a multiple graph regularized NMF (MultiGNMF) to approximate intrinsic manifold approximation automatically. Similarly, a relational multi-manifold co-clustering (RMC) approach [58] is proposed to maximally approximate the true intrinsic manifolds of both the sample and feature spaces simultaneously. Later, Wang et al., [102] proposed two GNMF-based methods to learn the graph that is adaptive to the selected features and learned multiple kernels, respectively. In the real world applications, there is certain amount of prior knowledge such as label information, which could be used to improve the performance. NMF-$\alpha$ [14] makes a good combination of NMF and SVM, which utilizes limited labeled samples to achieve the support vectors of large-margin classifiers. Later, Liu et al. [64] proposed a constrained NMF (CNMF),

$$D(\mathbf{X}, \mathbf{WH}) = \|\mathbf{X} - \mathbf{W}\mathbf{Z}\mathbf{A}^T\|_F^2. \tag{2.10}$$

Here $\mathbf{H} = \mathbf{Z}\mathbf{A}^T$, $\mathbf{Z}$ is an auxiliary matrix and $\mathbf{A}$ is label constraint matrix which forces data samples with identical class label to have the same representation so that the samples are more discriminative. Li et al. [59] proposed a Locally Constrained A-optimal nonnegative projection method which not only preserves the locally geometrical structure of the data but also incorporates label information as constraints to enhance the discriminating power. Under the assumption that data samples from different domains have different distributions, but share same feature and class label spaces, Wang et al. [101] proposed a novel NMF-based approach for multiple-domain learning.

Generally, all the existing MMFs which either incorporate regularization terms or prior information for more accurate learning, all tend to obtain a single representation. However, it is well recognized that real data are usually complex and contain various components. For example, face images have expressions and genders. Each component mainly reflects one aspect of data only and provides information others do not have. Therefore, exploring the semantic information of multiple components as well as the diversity among them is of great benefit to understand data comprehensively and in-depth. Besides, real data such as video sequences contain ordered structure, i.e., consecutive neighbouring data samples are very likely share similar features unless a sudden change occurs. Since this ordered structure provides valuable information about the relationship between data, exploiting the ordered structure with NMF holds a great potential for seeking for optimal representations.

### 2.3.2 Multi-view NMF

All the NMF methods mentioned above are developed to handle a single view (feature) for finding explicit data representations. In fact, data collected from various sources or represented by different feature extractors are available in many real-world applications [109, 4, 28, 9, 97, 107]. For example, a web page that shown in the figure 2.1, may be represented by multiple contents and hyperlinks; one document may be translated into different languages; an image or video can be represented by different visual descriptors, such as SIFT [71], HOG [18] and GIST [79]; research communities are formed according to research topics as well as co-authorship links and so on. These heterogeneous features that are represented by different perspectives of data are referred as multiple views [74, 110]. Taken alone, each of these views will often be deficient or incomplete because different views describe distinct perspectives of data. Therefore, a key problem for data analysis is how to integrate the multiple views and discover the underlying structures.

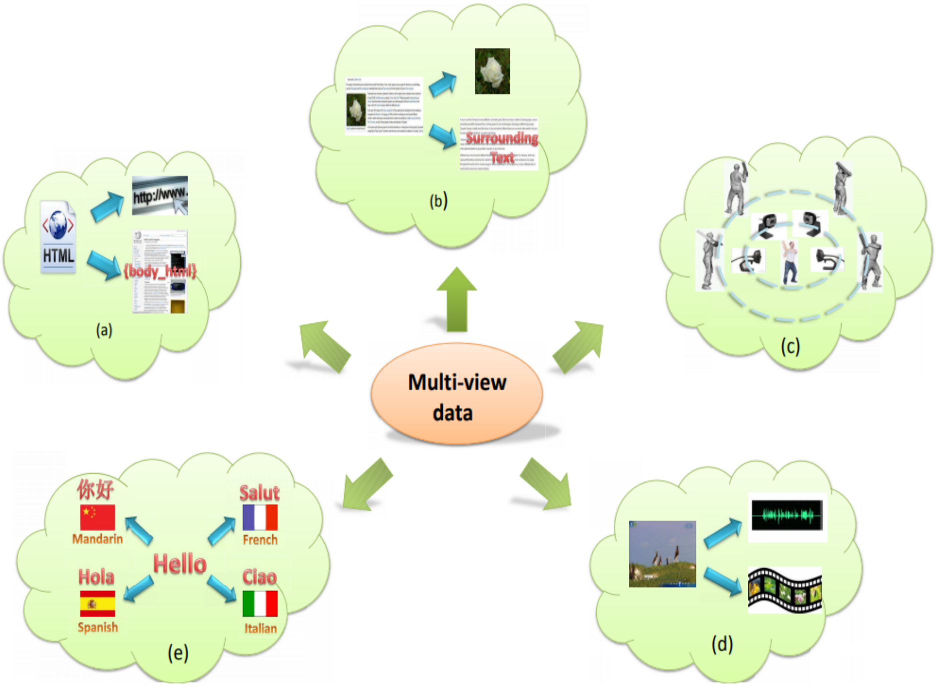


Figure 2.1: Multi-view data: a) a web document represented by its URL and words on the page, b) a web image depicted by its surrounding text separate to the visual information, c) images of a 3D object taken from different viewpoints, d) video clips combined with audio signals and visual frames, e) multilingual documents with one view in each language [109].

Recently, some NMF-based approaches on learning from multi-view data have been proposed. With the increasing amount of multi-view data, the approaches employing NMF-based multi-view learning have attracted attention. Assuming that a dataset comes with $V$ views, the objective function of multi-view NMF can be written as

$$\min_{\mathbf{W}^{(i)},\mathbf{H}^{(i)}\geq 0} \sum_{i=1}^{V} D(\mathbf{X}^{(i)}, \mathbf{W}^{(i)}\mathbf{H}^{(i)}) + \alpha \sum_{i=1}^{V} J_1(\mathbf{W}^{(i)}) + \beta \sum_{i=1}^{V} J_2(\mathbf{H}^{(i)}) \tag{2.11}$$

. For example, MultiNMF [65] formulates a joint multi-view NMF learning process with the constraint that encourages representation of each view towards a common consensus.

$$J_2(\mathbf{H}^{(i)}) = \|\mathbf{H}^{(i)}\mathbf{Q}^{(i)} - \mathbf{H}^{*}\|_F^2, \tag{2.12}$$

where $\mathbf{Q}^i = Diag(\sum_{j=1}^{m}\mathbf{W}_{j,1}^{(i)}, \sum_{j=1}^{m}\mathbf{W}_{j,2}^{(i)}, \ldots, \sum_{j=1}^{m}\mathbf{W}_{j,k}^{(i)})$ to ensure the representations of different views are comparable and $\mathbf{H}^{*}$ is the common consensus matrix.

Subsequently, several approaches [116, 46, 80, 99] were proposed based on MultiNMF. Specifically, Zhang et al. [116] developed a multi-manifold NMF (MMNMF) by incorporating the locally geometrical structure of data across multiple views. It regards each view as one manifold and the intrinsic manifold of a dataset as a mixture of the manifolds. Kalayeh et al. [46] proposed a weighted extension of MultiNMF [65] for image annotation, in which two weight matrices are introduced to alleviate the issue of dataset imbalance in real applications. Ou et al. [80] explored the local geometric structure for each view under the patch alignment framework and adopted correntropy-induced metric to measure the reconstruction error of each view to improve the robustness. Though existing approaches have shown superior results, some limitations remain to be dealt with. Firstly, existing approaches are unable to exploit the distinct information embedded of each view, so that the learned data representations from existing approaches contain mutually redundant information and lack diverse information. Secondly, given that utilizing a small amount of prior information can produce considerable improvements in learning accuracy [2], [119], it is potentially beneficial to incorporate such information to improve the discriminability of representations. This has not yet been attempted in existing approaches. Finally, since each view often contributes to final performance unequally, the selection of a weight for each view could result in a substantial effect on the results. However, the current methods only determine weights empirically using the labeled data or the same weight for all views, which restricts their applications in practice. Therefore, it is necessary and important to address these limitations and explores correlations among different views more effectively.

## 2.4 Evaluation metrics

To evaluate NMF-based approaches for data clustering, the accuracy (AC) [64], the normalized mutual information (NMI) [64] and the purity [21] are three widely used evaluation metrics to assess the quality of the results. For all the three metrics, the higher value indicates better clustering quality. These measurements are widely used by comparing the obtained label of each sample with that provided by the data set in different clustering approaches.

**Clustering accuracy** (**AC**) is used to measure the percentage of correct labels obtained. Given a data set containing $n$ images, let $l_i$ and $r_i$ be the the obtained cluster label and label provided from each sample images, respectively. The AC is defined as follows,

$$AC = \frac{\sum_{i=1}^{n} \delta(r_i, map(l_i))}{n} \tag{2.13}$$

where $\delta(x, y)$ is the delta function that equals one if $x = y$ and equals zero otherwise, and $map(l_i)$ is the permutation mapping function that maps each cluster label $l_i$ to the equivalent label $r_i$ from the data set. The best mapping can be found by using the Kuhn-Munkres algorithm [82].

**Normalized mutual information** (**NMI**) is used to measure the similarity between the cluster assignments and the pre-existing input labeling of the classes. Let $C$ and $C'$ denote the set of clusters obtained from the ground truth and obtained from our algorithm, respectively, their mutual information metric $MI(C, C')$ is defined as follows,

$$MI(C, C') = \sum_{c_i \in C, {c_j}' \in C'} p(c_i, {c_j}') \cdot log \frac{p(c_i, {c_j}')}{p(c_i) \cdot p({c_j}')}, \tag{2.14}$$

where $p(c_i)$, $p({c_j}')$ are the probabilities that an image randomly selected from the data set belongs to the clusters $c_i$ and $c_j$, respectively, and $p(c_i, {c_j}')$ denotes the joint probability that this randomly selected image belongs to the cluster $c_i$ as well as $c_j$ at the same time. In our experiment, we used the normalized metric $NMI(C, C')$ as follows,

$$NMI(C, C') = \frac{MI(C, C')}{max(H(C), H(C'))}, \tag{2.15}$$

where $H(C)$ and $H(C')$ are the entropies of $C$ and $C'$, respectively. It is easy to check that $NMI(C,C')$ ranges from 0 to 1. $NMI = 1$ when the two sets of image clusters are identical, and it becomes zero when the two sets are completely independent.

**Purity** measures the extent to which each cluster contained data points from primarily one class. The purity of a clustering solution is obtained as a weighted sum of individual cluster purity values and is given by

$$Purity = \sum_{i=1}^{K} \frac{n_i}{n} P(S_i), P(S_i) = \frac{1}{n_i} \max_j (n_i^j) \tag{2.16}$$

where $S_i$ is a particular cluster of size $n_i$, $n_i^j$ is the number of documents of the $i$-th input class that were assigned to the $j$-th cluster, $K$ is the number of clusters and $n$ is the total number of points.

# Chapter 3

# Multi-Component Nonnegative Matrix Factorization

## 3.1 Introduction

This chapter mainly explores semantic information of embedded multiple components of data and diverse information among them. In reality, data are usually complex and contain various components. Taken the Yale dataset[1] (Figure 3.1 ) as an example, the face images consist of multiple components including gender, facial expressions, ethnicity, and lighting direction (under which the images were taken), etc. Since each component mainly represents one subset of features and contains the specific information of the data, it is important to explore diverse information from multiple components in order to represent data more comprehensively and accurately. Besides, when clustering the dataset with exploring latent multiple components, multiple clustering solutions can be obtained such as one cluster of images can be faces with glasses and another can be faces with a happy expression. This will also enables us to understand data at a semantic level.

To this end, a novel NMF based approach is proposed for multi-component learning. It captures more comprehensive information and interprets data from different perspectives, by leveraging the multiple components. Specifically, different from existing NMF-based approaches that seek for a single representation

[1] http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html

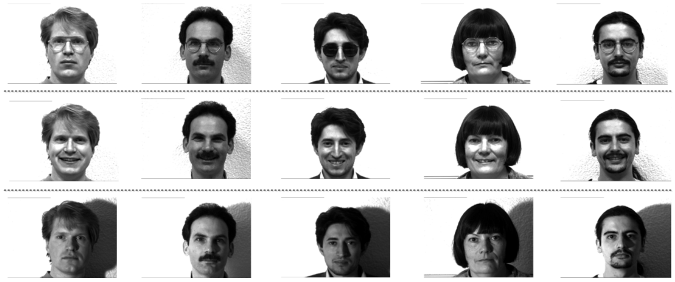


Figure 3.1: Sample images of the Yale dataset. Each column shows one subject's faces. Images in the same rows contain same components, such as faces with glasses and a neutral expression in row 1; faces without glasses and a happy expression in row 2; faces lit from left and with a neutral expression in row 3.

matrix, the proposed approach learns multiple representations simultaneously. By utilizing the Hilbert Schmidt Independence Criterion (HSIC) as a diversity term, the proposed approach explores the diverse information among the representations, where each representation corresponds to a component.

## 3.2 Single-Component NMF

The previous chapter has briefly introduced NMF. That is, given a $n$ data matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n] \in \mathbb{R}_+^{m \times n}$ where each column is a $m$-dimensional data vector, NMF [55] aims to minimize the following nmf22ective function:

$$\min_{\mathbf{W},\mathbf{H}\geq 0} \|\mathbf{X} - \mathbf{W}\mathbf{H}\|_F^2, \tag{3.1}$$

In essence, current single-view NMF-based approaches are all based on this standard NMF with regarding the features of data as a whole and seeking for a single representation matrix. Obviously, they are unable to distinguish these embedded components and can be considered as single-component approaches.

## 3.3 Multi-Component NMF

Real data often contains multiple latent components, since each component provides distinct information to each other, it is of paramount importance to explore diversity from multiple latent components for comprehensive and accurate data representations. Also, it is arguable that the semantic information of data is much richer than what a single component can capture. Hence, we propose a multi-component NMF (MCNMF) approach to seek for more accurate learning and exploit semantic meaning of data simultaneously. Different from current NMF-based approaches that seek for a single representation matrix, MCNMF learns multiple representations simultaneously, where each representation corresponds to each component. Figure 3.2 shows the differences between current NMFs and MCNMF with application on clustering. We can see that current NMFs get only one clustering solution (i.e., all face images of a subject being grouped into one cluster) based on global features of a single representation matrix. However, based on learning representations of multiple components, MCNMF can achieve multiple clustering solutions. For example, one cluster of images can be faces with glasses and another can be faces with a happy expression. In the following subsection, we will introduce our MCNMF model.

### 3.3.1 Objective function

Assuming $\mathbf{X}$ comes with $V$ components, we use $\mathbf{H}^{(i)} \in \mathbb{R}^{k^{(i)} \times n}$ to denote the representation with $k^{(i)}$-dimensional features that corresponds to the $i$-th ($i \in \{1, 2, \ldots, V\}$) components, and $\mathbf{W}^{(i)}$ be the corresponding representation matrix of $\mathbf{H}^{(i)}$. Then the product of each $\mathbf{W}^{(i)}\mathbf{H}^{(i)}$ should well approximate $\mathbf{X}$, i.e., $\mathbf{X} \approx \mathbf{W}^{(i)}\mathbf{H}^{(i)}$, from each perspective. To seek for multiple optimal representations $\{\mathbf{H}^{(i)^*}\}_{i=1}^{V}$, we have the following function:

$$\min_{\mathbf{W}^{(i)} \geq 0, \mathbf{H}^{(i)} \geq 0} \sum_{i=1}^{V} \|\mathbf{X} - \mathbf{W}^{(i)}\mathbf{H}^{(i)}\|_F^2. \tag{3.2}$$

This will allow us to factorize $\mathbf{X}$ straightforwardly. However, it may fail to explore the diverse information of multiple components effectively as each $\mathbf{H}^{(i)}$ could be very close to or even same as each other.

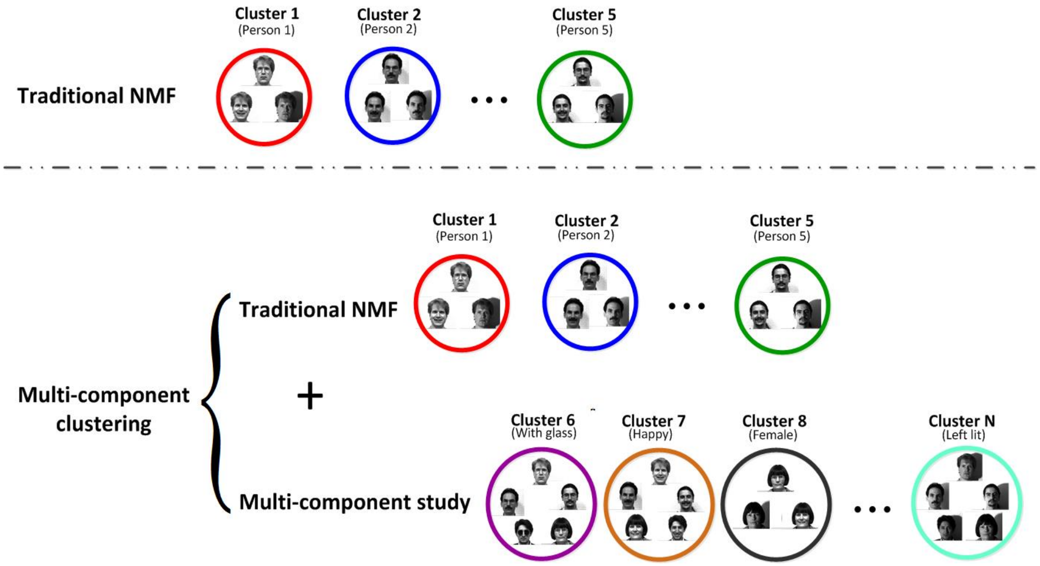


Figure 3.2: The comparison of traditional NMFs and MCNMF, where circles in different colors represent different clusters.

For any data, $\mathbf{x}_f$, it comes with a pair of components, $i$ and $j$. $\mathbf{x}_f$'s latent distinct information of each component cannot be fully explored unless its representations of two components, i.e., $\mathbf{h}_f^{(i)}$ and $\mathbf{h}_f^{(j)}$, are enforced to be independent to each other. Given $n$ data vectors, we assume that each $i$th component is drawn from $\mathcal{X}$ space and the $j$th component from $\mathcal{Y}$ space. Then, in essence, we aim to learn a mapping function $G$ of their representations from $S := \{(\mathbf{h}_1^{(i)}, \mathbf{h}_1^{(j)}), (\mathbf{h}_2^{(i)}, \mathbf{h}_2^{(j)}), \ldots, (\mathbf{h}_n^{(i)}, \mathbf{h}_n^{(j)})\} \subseteq \mathcal{X} \times \mathcal{Y}$, i.e., $G$: $\mathcal{X} \rightarrow \mathcal{Y}$, to minimize the dependence between data representations in the $\mathcal{X}$ and $\mathcal{Y}$.

To do so, we employ the Hilbert-Schmidt Independence Criterion (HSIC)[32] due to its several advantages. First, HSIC measures dependence by mapping variables into a reproducing kernel Hilbert space (RKHS) such that correlations measured in that space correspond to high-order joint moments between the original distributions and more complicated (such as nonlinear) dependence can be addressed. Second, it is able to estimate dependence between variables without explicitly estimating the joint distribution of the random variables. Hence, it is of high computational efficiency. Last but not least, the empirical HSIC turns out to be equal to the trace of product of the data matrix, which makes our problem solvable. HSIC computes the square of the norm of the cross-covariance operator over the domain $\mathcal{X} \times \mathcal{Y}$ in Hilbert Space. As an effective measure of dependence, the HSIC has been applied to several machine learning tasks recently [88, 117, 78]. Mathmatically, an empirical estimate of the HSIC [32] is defined as

$$\text{HSIC}(\mathbf{H}^{(i)}, \mathbf{H}^{(j)}) = (n-1)^{-2} tr(\mathbf{R}\mathbf{K}^{(i)}\mathbf{R}\mathbf{K}^{(j)}), \tag{3.3}$$

where $\mathbf{K}^{(i)}$ and $\mathbf{K}^{(j)}$ are the centered Gram matrices [2] of kernel functions defined over $\mathbf{H}^{(i)}$ and $\mathbf{H}^{(j)}$. $\mathbf{R} = \mathbf{I} - \frac{1}{n}\mathbf{e}\mathbf{e}^T$, where $\mathbf{I}$ is an identity matrix and $\mathbf{e}$ is an all-one column vector.

Thus, to explore the diverse information from more components, we extend (3.3) and combine it with (3.2) to produce the following function:

$$\min_{\mathbf{W}^{(i)} \geq 0, \mathbf{H}^{(i)} \geq 0} \sum_{i=1}^{V} \|\mathbf{X} - \mathbf{W}^{(i)}\mathbf{H}^{(i)}\|_F^2 + \alpha \sum_{j \neq i} \text{HSIC}(\mathbf{H}^{(i)}, \mathbf{H}^{(j)}), \tag{3.4}$$

where $\alpha$ is the parameter of the diversity regularization term. The first term represents the error between $\mathbf{X}$ and the product of the basis and representation matrices in different components. The second term ensures that any two of $V$ representations be diverse to each other.

Here, we use the inner product kernel for HSIC, i.e., $\mathbf{K}^{(i)} = {\mathbf{H}^{(i)}}^T \mathbf{H}^{(i)}$. For notational convenience, we ignore the scaling factor $(n-1)^{-2}$ of HSIC, and rewrite (3.4) to form the final objective function as

$$\min_{\mathbf{W}^{(i)} \geq 0, \mathbf{H}^{(i)} \geq 0} \sum_{i=1}^{V} \|\mathbf{X} - \mathbf{W}^{(i)}\mathbf{H}^{(i)}\|_F^2 + \alpha \sum_{j \neq i} tr(\mathbf{R}\mathbf{K}^{(i)}\mathbf{R}\mathbf{K}^{(j)}). \tag{3.5}$$

After obtaining the optimal representation ${\mathbf{H}^{(i)}}^*$ of each component, the final aggregated representation $\mathbf{H}^*$ can be obtained by combining all ${\mathbf{H}^{(i)}}^*$, i.e., $\mathbf{H}^* = [{\mathbf{H}^{(1)}}^*, {\mathbf{H}^{(2)}}^*, \ldots, {\mathbf{H}^{(V)}}^*] \in \mathbb{R}^{\sum_{i=1}^{V} k^{(i)} \times n}$.

**Remarks**: When $V = 1$, (3.1) is exact that of NMF. Moreover, Our method is not limited to one specific NMF method. Our method is based on standard NMF,

[2] Given a set $V$ of $m$ vectors $\in \mathbb{R}^n$ ), the Gram matrix $G$ is the matrix of all possible inner products of $V$

since it is clearer to demonstrate the effectiveness of latent components without the help of other regularization. Nevertheless, other state-of-the-art NMF-based approaches, such as GNMF [6], RNMF [50] can also be implemented into our method and better results can be expected.

### 3.3.2 Optimization

The optimization problem in (3.5) is not convex in both variables $\mathbf{W}^{(i)}$ and $\mathbf{H}^{(i)}$, so it is infeasible to find the global minimum. In addition, as the matrix $\mathbf{R}$ contains negative values, it is technically challenging to solve (3.5) directly. Here we propose an algorithm that separates the optimization of (3.5) to two subproblems and optimizes them iteratively, which guarantees each subproblem converges to the local minima.

**$\mathbf{W}^{(i)}$-subproblem**: Updating $\mathbf{W}^{(i)}$ with $\mathbf{H}^{(i)}$ fixed in (3.5) leads to a standard NMF formulation [56], so the updating rule for $\mathbf{W}^{(i)}$ is

$$\mathbf{W}^{(i)} \leftarrow \mathbf{W}^{(i)} \odot \frac{(\mathbf{X}\mathbf{H}^{(i)^T})}{(\mathbf{W}^{(i)}\mathbf{H}^{(i)}\mathbf{H}^{(i)^T})}. \tag{3.6}$$

**$\mathbf{H}^{(i)}$-subproblem**: When updating $\mathbf{H}^{(i)}$ with $\mathbf{W}^{(i)}$ in (3.5) fixed, we need to solve the following function:

$$\min_{\mathbf{H}^{(i)}\geq 0} J(\mathbf{H}^{(i)}) = \|\mathbf{X} - \mathbf{W}^{(i)}\mathbf{H}^{(i)}\|_F^2 + \alpha \sum_{j=1, j\neq i}^{V} tr(\mathbf{R}\mathbf{K}^{(i)}\mathbf{R}\mathbf{K}^{(j)}) \tag{3.7}$$

In general, the method of Lagrange Multipliers is used to find the solution for optimization problems constrained to one or more equalities. Since the constraints of (3.7) also have inequalities, we need to extend the method to the KKT conditions.

**Definition 1** The KKT conditions is when given a problem

$$\begin{aligned} x* &= \arg\min_x f(x) \\ s.t. \quad h_i(x) &= 0, \forall i = 1, \ldots, m \\ s.t. \quad g_i(x) &\leq 0 \forall i = 1, \ldots, m \end{aligned}$$

The expression for the optimization problem becomes:

$$x* = \arg\min L(x,\lambda,\mu) = \arg\min_x f(x) + \sum_{i=1}^{m} \lambda_i h_i(x) + \sum_{i=1}^{n} \mu_i g_i(x),$$

here $\arg\min L(x,\lambda,\mu)$ is the Lagrangian and depends also on $\lambda$ and $\mu$, which are vectors of the multipliers.

According to the **Definition 1**, we then introduce a Lagrange multiplier matrix $\boldsymbol{\eta} = [\eta_{pq}] \in \mathbb{R}^{k\times n}$ for the nonnegative constraint on $\mathbf{H}^{(i)}$. Utilizing $\|\mathbf{A}\|_F^2 = tr(\mathbf{A}^T\mathbf{A})$, we obtain the following function:

$$\begin{aligned} \min_{\mathbf{H}^{(i)}\geq 0} J'(\mathbf{H}^{(i)}) &= tr(\mathbf{X}\mathbf{X}^T) - 2tr(\mathbf{X}{\mathbf{H}^{(i)}}^T{\mathbf{W}^{(i)}}^T) \\ &+ tr(\mathbf{W}^{(i)}\mathbf{H}^{(i)}{\mathbf{H}^{(i)}}^T{\mathbf{W}^{(i)}}^T) \\ &+ \alpha \sum_{j=1,j\neq i}^{V} tr(\mathbf{R}{\mathbf{H}^{(i)}}^T\mathbf{H}^{(i)}\mathbf{R}\mathbf{K}^{(j)}) + tr(\boldsymbol{\eta}\mathbf{H}^{(i)}). \end{aligned} \tag{3.8}$$

Setting the derivative of $J'(\mathbf{H}^{(i)})$ to be 0 with respect to $\mathbf{H}^{(i)}$, we have

$$\boldsymbol{\eta} = {\mathbf{W}^{(i)}}^T\mathbf{X} - {\mathbf{W}^{(i)}}^T\mathbf{W}^{(i)}\mathbf{H}^{(i)} - \alpha\mathbf{H}^{(i)}\mathbf{R}\sum_{j=1,j\neq i}^{V}\mathbf{K}^{(j)}\mathbf{R}. \tag{3.9}$$

Following the Karush-Kuhn-Tucker (KKT) condition [5] for the nonnegativity of $\mathbf{H}^{(i)}$, we have the following equation:

$$({\mathbf{W}^{(i)}}^T\mathbf{X} - {\mathbf{W}^{(i)}}^T\mathbf{W}^{(i)}\mathbf{H}^{(i)} - \alpha\mathbf{H}^{(i)}\mathbf{R}\sum_{j=1,j\neq i}^{V}\mathbf{K}^{(j)}\mathbf{R})_{pq}H_{pq}^{(i)} = 0. \tag{3.10}$$

Because $\mathbf{R}$ contains negative values, we decompose $\mathbf{R}$ into two nonnegative parts for ensuring $\mathbf{H}^{(i)} \geq 0$ in each iteration:

$$\mathbf{R} = \mathbf{R}^+ - \mathbf{R}^-, \tag{3.11}$$

where $\mathbf{R}_{pq}^+ = (|\mathbf{R}_{pq}| + \mathbf{R}_{pq})/2$ and $\mathbf{R}_{pq}^- = (|\mathbf{R}_{pq}| - \mathbf{R}_{pq})/2$. Substituting (3.11) into

(3.10), we obtain

$$\begin{aligned}&(\mathbf{W}^{(i)^T}\mathbf{X} - \mathbf{W}^{(i)^T}\mathbf{W}^{(i)}\mathbf{H}^{(i)} + \alpha\mathbf{H}^{(i)}(\mathbf{R}^+ \sum_{j=1,j\neq i}^{V} \mathbf{K}^{(j)}\mathbf{R}^- + \mathbf{R}^- \sum_{j=1,j\neq i}^{V} \mathbf{K}^{(j)}\mathbf{R}^+) \\ &- \alpha\mathbf{H}^{(i)}(\mathbf{R}^- \sum_{j=1,j\neq i}^{V} \mathbf{K}^{(j)}\mathbf{R}^- + \mathbf{R}^+ \sum_{j=1,j\neq i}^{V} \mathbf{K}^{(j)}\mathbf{R}^+))_{pq}H_{pq}^{(i)} = 0.\end{aligned} \tag{3.12}$$

This is the fixed point equation whose solution must satisfy at convergence. Denote $\mathbf{R}_a = \mathbf{R}^+ \sum_{j=1,j\neq i}^{V} \mathbf{K}^{(j)}\mathbf{R}^-$, $\mathbf{R}_b = \mathbf{R}^- \sum_{j=1,j\neq i}^{V} \mathbf{K}^{(j)}\mathbf{R}^+$, $\mathbf{R}_c = \mathbf{R}^-\sum_{j=1,j\neq i}^{V} \mathbf{K}^{(j)}\mathbf{R}^-$, $\mathbf{R}_d = \mathbf{R}^+ \sum_{j=1,j\neq i}^{V} \mathbf{K}^{(j)}\mathbf{R}^+$, then given an initial value of $\mathbf{H}^{(i)}$, the successive update of $\mathbf{H}^{(i)}$ is:

$$\mathbf{H}^{(i)} \leftarrow \mathbf{H}^{(i)} \odot \sqrt{\frac{\mathbf{W}^{(i)^T}\mathbf{X} + \alpha\mathbf{H}^{(i)}(\mathbf{R}_a + \mathbf{R}_b)}{\mathbf{W}^{(i)^T}\mathbf{W}^{(i)}\mathbf{H} + \alpha\mathbf{H}^{(i)}(\mathbf{R}_c + \mathbf{R}_d)}}. \tag{3.13}$$

The correctness of the updating rule (3.13) can be guaranteed by the following theorem.

**Theorem 1**. If the updating rule of $\mathbf{H}^{(i)}$ converges, then the final solution satisfies the KKT optimality condition.

**Proof of Theorem 1**. At convergence, $\mathbf{H}^{\infty} = \mathbf{H}^{t+1} = \mathbf{H}^t = \mathbf{H}$, where $t$ denotes the $t$-th iteration, i.e.,

$$\mathbf{H}^{(i)} = \mathbf{H}^{(i)} \odot \sqrt{\frac{\mathbf{W}^{(i)^T}\mathbf{X} + \alpha\mathbf{H}^{(i)}(\mathbf{R}_a + \mathbf{R}_b)}{\mathbf{W}^{(i)^T}\mathbf{W}^{(i)}\mathbf{H} + \alpha\mathbf{H}^{(i)}(\mathbf{R}_c + \mathbf{R}_d)}} \tag{3.14}$$

Then for each $H_{pq}^{(i)}$ , we have

$$\begin{aligned}&(\mathbf{W}^{(i)^T}\mathbf{X} - \mathbf{W}^{(i)^T}\mathbf{W}^{(i)}\mathbf{H}^{(i)} + \alpha\mathbf{H}^{(i)}(\mathbf{R}_a + \mathbf{R}_b) \\ &- \alpha\mathbf{H}^{(i)}(\mathbf{R}_c + \mathbf{R}_d))_{pq}(\mathbf{H}^{(i)})_{pq}^2 = 0.\end{aligned} \tag{3.15}$$

which is equivalent to (3.12).

We can now prove the convergence of the updating rule, by making use of an auxiliary function as in [56]. The definition of the auxiliary function is as follows:

**Definition 2**. A function $G(\mathbf{Q}, \mathbf{Q}')$ is an auxiliary function of the function $J(\mathbf{Q})$ if $G(\mathbf{Q}, \mathbf{Q}') \geq J(\mathbf{Q})$ and $G(\mathbf{Q}, \mathbf{Q}) = J(\mathbf{Q})$ for any $\mathbf{Q}$, $\mathbf{Q}'$.

The auxiliary function gives rise to the following lemma [56]:

**Lemma 1**. If $G$ is an auxiliary function of $J$, then $J$ is non-increasing under the update rule $\mathbf{Q}^{t+1} = \arg\min_{\mathbf{Q}} G(\mathbf{Q}, \mathbf{Q}^t)$.

Under the constraint in (3.11), we now have the specific form of the auxiliary function $G(\mathbf{H}^{(i)}, \mathbf{H}^{(i)'})$ for the objective function $J(\mathbf{H}^{(i)})$ in (3.7) based on Lemma 2.

**Lemma 2**. The function

$$\begin{aligned} G(\mathbf{H}^{(i)}, \mathbf{H}^{(i)'}) &= -2\sum_{pq}(\mathbf{W}^{(i)^T}\mathbf{X})_{pq}\mathbf{H}^{(i)'}{}_{pq}(1 + \log\frac{\mathbf{H}_{pq}^{(i)}{}'}{\mathbf{H}_{pq}^{(i)}}) \\ &+ \sum_{pq}\frac{(\mathbf{W}^{(i)^T}\mathbf{W}^{(i)}\mathbf{H}^{(i)'})_{pq}\mathbf{H}_{pq}^{(i)^2}}{\mathbf{H}_{pq}^{(i)'}} \\ &- \sum_{pqk}(\mathbf{R}_a + \mathbf{R}_b)_{jk}\mathbf{H}^{(i)'}{}_{pq}\mathbf{H}^{(i)'}{}_{pk}(1 + \log\frac{\mathbf{H}_{pq}^{(i)}\mathbf{H}^{(i)}{}_{pk}}{\mathbf{H}^{(i)'}{}_{pq}\mathbf{H}^{(i)'}{}_{pk}}) \\ &+ \sum_{pq}\frac{(\mathbf{H}^{(i)'}(\mathbf{R}_c + \mathbf{R}_d))_{pq}\mathbf{H}_{pq}^{(i)^2}}{\mathbf{H}_{pq}^{(i)'}} \end{aligned} \tag{3.16}$$

is an auxiliary function for $J(\mathbf{H}^{(i)})$ in (3.7).

**Proof of Lemma 2**. We find upper bounds for each of the two positive terms by the following lemma [20],

**Lemma 3**. For any nonnegative matrices $\mathbf{S} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{g\times g}$, $\mathbf{F} \in \mathbb{R}^{n\times g}$ and $\mathbf{F}' \in \mathbb{R}^{n\times g}$, with $\mathbf{S}$ and $\mathbf{B}$ being symmetric, then the following inequality holds

$$tr(\mathbf{F}^T\mathbf{S}\mathbf{F}\mathbf{B}) \leq \sum_{i=1}^{n}\sum_{p=1}^{g}(\mathbf{S}\mathbf{F}'\mathbf{B})\frac{\mathbf{F}^2{}_{ip}}{\mathbf{F}'{}_{ip}}. \tag{3.17}$$

Then, we have following inequations:

$$tr(\mathbf{W}^{(i)^T}\mathbf{W}^{(i)}\mathbf{H}^{(i)}\mathbf{H}^{(i)^T}) \leq \sum_{pq}\frac{(\mathbf{W}^{(i)^T}\mathbf{W}^{(i)}\mathbf{H}'_i)_{pq}(\mathbf{H}^{(i)})^2_{pq}}{(\mathbf{H}'_i)_{pq}}, \tag{3.18}$$

$$tr(\mathbf{H}(\mathbf{R}_c + \mathbf{R}_d)\mathbf{H}^T) \leq \sum_{pq}\frac{(\mathbf{H}'(\mathbf{R}_c + \mathbf{R}_d))_{pq}\mathbf{H}^2_{pq}}{\mathbf{H}'_{pq}}. \tag{3.19}$$

To obtain lower bounds for the remaining terms, we use the inequality $z >$

$1 + \log z, \forall z > 0$ [20] and have

$$tr(\mathbf{W}^T\mathbf{X}\mathbf{H}^T) \geq \sum_{pq}(\mathbf{W}^T\mathbf{X})_{pq}\mathbf{H}'_{pq}(1 + \log\frac{\mathbf{H}_{pq}}{\mathbf{H}'_{pq}}), \tag{3.20}$$

$$tr(\mathbf{H}(\mathbf{R}_a + \mathbf{R}_b)\mathbf{H}^T) \geq \sum_{pqk}(\mathbf{R}_a + \mathbf{R}_b)_{jk}\mathbf{H}'_{pq}\mathbf{H}'_{pk}(1 + \log\frac{\mathbf{H}_{pq}\mathbf{H}_{pk}}{\mathbf{H}'_{pq}\mathbf{H}'_{pk}}). \tag{3.21}$$

Collecting all bounds, we have the final auxiliary function in Lemma 2.

Based on the lemmas 1 and 2, we can prove the convergence of the updating rule (3.13).

**Theorem 2**. The optimization problem (3.7) is non-increasing under the iterative updating rule (3.13).

**Proof of Theorem 2**. Lemma 2 provides a specific form $G(\mathbf{H}, \mathbf{H}')$ of the auxiliary function for $J(\mathbf{H})$ in the problem (3.7). We can have the solution for $\min_{\mathbf{H}} G(\mathbf{H}, \mathbf{H}')$ by the following KKT condition

$$\begin{aligned}\frac{\partial G(\mathbf{H}, \mathbf{H}')}{\partial \mathbf{H}_{pq}} =& -2(\mathbf{W}^T\mathbf{X})_{pq}\frac{\mathbf{H}'_{pq}}{\mathbf{H}_{pq}} + 2\frac{(\mathbf{W}^T\mathbf{W}\mathbf{H}')_{pq}\mathbf{H}_{pq}}{\mathbf{H}'_{pq}} \\ & - 2\frac{(\mathbf{H}'(\mathbf{R}_a + \mathbf{R}_b))_{pq}\mathbf{H}'_{pq}}{\mathbf{H}_{pq}} + 2\frac{(\mathbf{H}'(\mathbf{R}_c + \mathbf{R}_d))_{pq}\mathbf{H}_{pq}}{\mathbf{H}'_{pq}} = 0,\end{aligned} \tag{3.22}$$

which gives rise to the updating rule in (3.13). Following Lemma 1, under this updating rule the objective function values of $J(\mathbf{H})$ in (3.7) will be non-increasing.

### 3.3.3 Complexity analysis

Based on (3.13) and (3.6), we estimate the number of operations for each iteration as above. The complexity of updating $\mathbf{W}^{(i)}$ is $\mathcal{O}(mnk^{(i)})$. When update $\mathbf{H}^{(i)}$, the cost of multiplications for $\mathbf{W}^{(i)T}\mathbf{X}$, $\mathbf{H}^{(i)}(\mathbf{R}_a + \mathbf{R}_b)$, $\mathbf{W}^{(i)T}\mathbf{W}^{(i)}\mathbf{H}^{(i)}$ and $\mathbf{H}^{(i)}(\mathbf{R}_c+\mathbf{R}_d)$ are $\mathcal{O}(k^{(i)}mn)$, $\mathcal{O}(\sum_{j=1,j\neq i}^{V}(k^{(j)}n^2+nk^{(j)^2}))$, $\mathcal{O}(mk^{(i)^2}+nk^{(i)^2})$ and $\mathcal{O}(\sum_{j=1,j\neq i}^{V}(k^{(j)}n^2+nk^{(j)^2}))$, respectively. Since usually $\{k^{(i)}, k^{(j)}\} \ll \min(m, n)$, the overall computation of MCNMF is $\mathcal{O}(\sum_{i=1}^{V}(\sum_{j=1,j\neq i}^{V} k^{(j)}n^2 + k^{(i)}mn))$. Since the updating rules for each element of both $\mathbf{W}^{(i)}$ and $\mathbf{H}^{(i)}$ at each iteration are independent, the computational cost can be significantly reduced if all elements are updated in parallel, such as through CUDA [39].

## 3.4 Experiments

### 3.4.1 Description of datasets

We carried several experiments on the following benchmark datasets to show the effectiveness of MCNMF.

- **Yale**: It contains 11 facial images for each of 15 subjects. Sample images are shown in Figure 3.1. For each subject, its face images are either in different facial expressions (such as happy or sad), or configurations (such as with or without glasses).

- **ORL**[3]: This dataset consists of 400 facial images belonging to 40 different subjects. Similar to the Yale dataset, the images were taken with various lighting and facial expressions.

- **Notting − Hill**[9] This is a video face dataset, which is derived from the movie "Notting Hill". The faces of 5 main casts were used, including 4660 faces in 76 tracks.

- **COIL20**[4]: It is composed of 1440 images for 20 nmf22ects. The 72 images of each nmf22ect were captured by a fixed camera at a pose intervals of 5 degree. For this dataset, we regard the different poses and shapes as components.

### 3.4.2 Experimental setup

We first compared MCNMF against the standard NMF [56] to verify the effectiveness of exploring diverse information from multi-components, and then with the state-of-the-arts: RNMF [50], GNMF [6], Cauchy NMF [69] and LANMF [66]. For each compared method, the parameters were set according to the parameter settings in original papers. For MCNMF, we varied the regularization parameter $\alpha$ within [0.01, 0.05] with 0.01 interval and fixed the number of components $V = 3$ . In addition, we set each $k^{(i)}$ equals to number of clusters according to the groundtruth of each dataset. The dimensions of obtained optimal representations $\mathbf{H}^*$ for all the compared methods were all set to be $k = \sum_{i=1}^{V} k^{(i)}$ for fair comparison.

[3] http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html

[4] http://www1.cs.columbia.edu/CAVE/software/softlib/coil-20.php

Table 3.1: Clustering results ((mean ± standard deviation)%) on the four datasets (bold numbers represent the best results)

| | **Metric** | **NMF** | **RNMF** | **GNMF** | **Cauchy NMF** | **LANMF** | **MCNMF** |
|---|---|---|---|---|---|---|---|
| **Yale** | AC | 40.48 ±3.25 | 38.55 ±2.76 | 41.58±2.54 | 41.45 ±4.26 | 39.76 ±2.70 | **46.42**±**1.95** |
| | NMI | 46.35 ±2.15 | 43.98 ±2.46 | 46.30 ±1.66 | 49.57 ±2.88 | 45.52 ±**1.25** | **49.65**±1.66 |
| | purity | 42.91 ±2.16 | 41.33 ±3.36 | 42.67±2.73 | 43.39 ±3.02 | 42.18 ±1.64 | **47.15**±**1.45** |
| **ORL** | AC | 54.90±3.44 | 54.20 ±2.11 | 59.60±2.50 | 56.45±2.86 | 52.40 ±2.31 | **62.95**±**1.20** |
| | NMI | 76.22±1.34 | 75.33±1.04 | 77.80±1.12 | 74.80±1.23 | 73.11±1.79 | **79.39**±**1.10** |
| | purity | 62.20±2.08 | 59.75±1.51 | 64.55±1.59 | 60.45±1.85 | 57.80±1.55 | **66.20**±**1.47** |
| **COIL20** | AC | 62.49±1.56 | 59.57±4.83 | 68.39±2.62 | 63.49±4.34 | 63.17 ±3.98 | **69.61**±**1.30** |
| | NMI | 74.35±1.49 | 73.24±2.21 | 77.30±1.54 | 76.34±2.08 | 76.63±2.51 | **78.84**±**0.81** |
| | purity | 66.40±1.37 | 64.29±3.75 | 69.83±2.47 | 67.28±2.06 | 67.68±3.51 | **70.06**±**1.25** |
| **Notting-Hill** | AC | 68.04±3.68 | 74.08 ±2.90 | 75.88 ±3.40 | 64.65±4.93 | 72.64 ±4.17 | **77.54**±**1.69** |
| | NMI | 60.27±3.50 | 64.74±**2.55** | 62.97±2.93 | 56.29±2.32 | 64.94±3.56 | **66.63** ±3.33 |
| | purity | 72.93±5.38 | 78.39±**2.25** | 77.19±2.85 | 70.25±3.93 | 78.67±3.67 | **79.49**±2.79 |

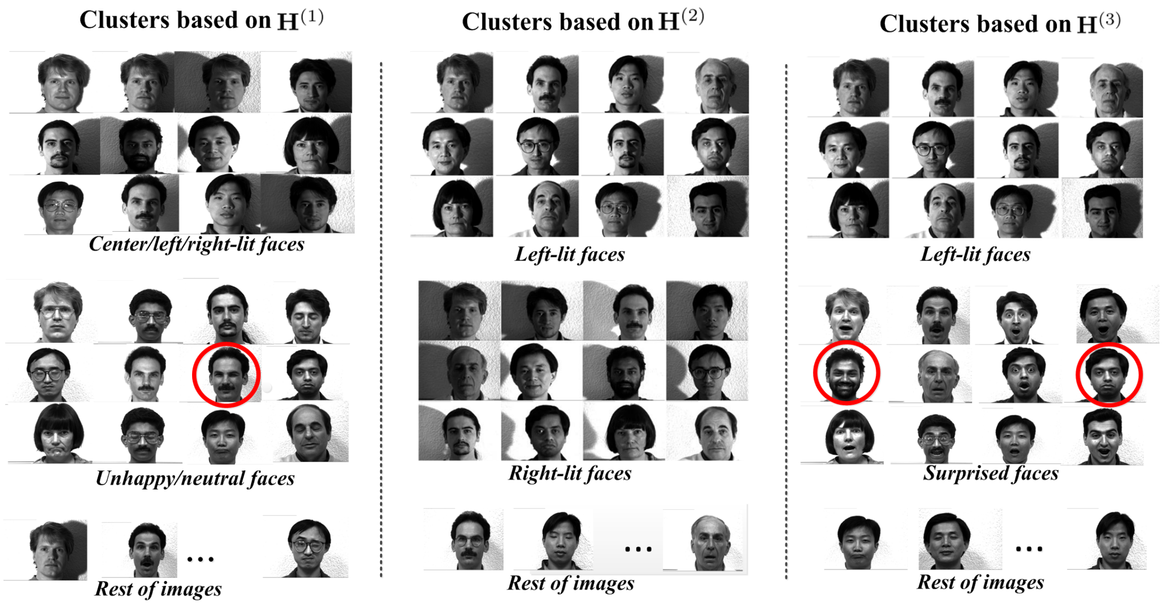

Clusters based on $\mathbf{H}^{(1)}$
Clusters based on $\mathbf{H}^{(2)}$
Clusters based on $\mathbf{H}^{(3)}$
Center/left/right-lit faces
Left-lit faces
Left-lit faces
Unhappy/neutral faces
Right-lit faces
Surprised faces
...
Rest of images
...
Rest of images
...
Rest of images

Figure 3.3: Sample clustering results of the Yale dataset based on each representation $\mathbf{H}^{(i)}$. Images circled in red are outliers.

### 3.4.3 Performance analysis

**Clustering result**. We applied $k$-means to the obtained representations $\mathbf{H}^*$ for clustering. Since $k$-means is sensitive to the initial values, we repeated the clustering process 50 times to give the average performance. Moreover, since all the compared methods converge to local minimum, we ran each method 10 times to avoid randomness. The final average clustering results along with standard deviations are reported in Table 3.1. As we can see, MCNMF outperforms the other methods against all metrics and gets the lowest standard deviations on 9 of 12 results, which demonstrate the robustness of MCNMF. Besides, it can be noticed that GNMF performs the second best in terms of AC, but not for other metrics. Especially, MCNMF outperforms GNMF with a large margin: 4.84% and 3.35% on the Yale and ORL, respectively. This is probably because that the images in both of the two datasets have more components, such as different lighting and expressions. Obviously, richer information has been explored and obtained for comprehensive representations, which brings significant improvements.

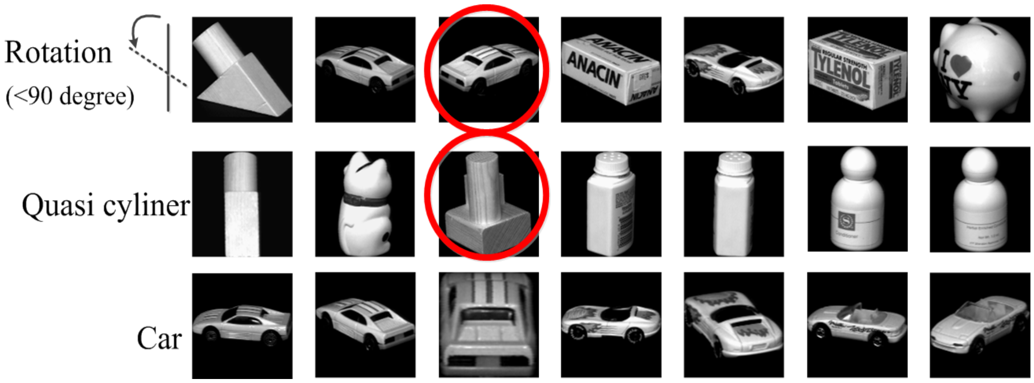


Figure 3.4: Sample clustering results of the COIL20 dataset. Each row, from top to bottom, represents a cluster based on the representations $\mathbf{H}^{(1)}$ , $\mathbf{H}^{(2)}$ and $\mathbf{H}^{(3)}$ , respectively. Images circled in red are outliers.

**Component study**. We closely examined the learned representations for each component to analyze their latent semantics. In particular, we took the Yale dataset as an example. Like the previous experiment setting, we fixed the number of components $V$ to 3, and applied $k$-means on the representation $\mathbf{H}^{(i)}$ of each component to cluster the data into 3 clusters. The results are shown in Figure 3.3. We can see that the each representation has effectively captured some distinct information (such as unhappy or surprised expressions) which is reflected by a corresponding cluster. This result enables the understanding of the data from various perspectives in a semantic level, which would be hardly achievable by current NMF-based methods as they cannot identify components. Also, note that from $\mathbf{H}^{(2)}$ and $\mathbf{H}^{(3)}$, there is a common cluster: the right-lit faces. This is reasonable that although multiple representations usually describe data from different perspectives, they are not completely exclusive to each other mutually. We further tested MCNMF on a larger dataset COIL20, example results are shown in Figure 3.4. Again, the results are quite good and promising, with multiple clusters being obtained through different components (right rotation, pottery, etc).

**Parameter analysis**. We tested the effect of parameter $\alpha$ of MCNMF on the datasets. $\alpha$ varies from 0.01 to 0.05 with an increment of 0.01. Here we presented the accuracy of MCNMF with respect to $\alpha$ on Yale and ORL as examples. Seen from Figure 3.5, the accuracy varies slightly showing a relatively stable performance. Also, for all values of $\alpha$, the performance of MCNMF is consistently better than NMF ( Table 6.1). For example, for Yale, the worst result of MCNMF is about 0.4182, while NMF only gets 0.4048.

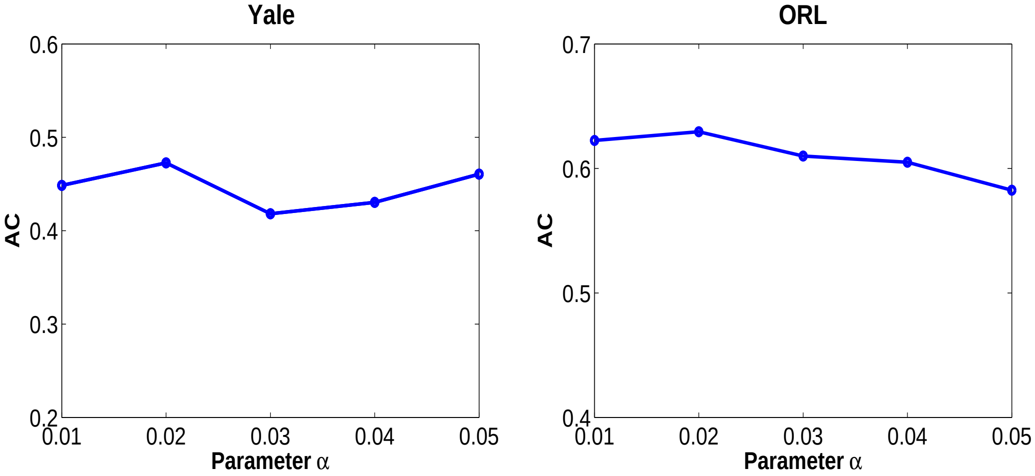


Figure 3.5: The effect of the parameter $\alpha$.

We also tested the effect of the number of components $V$. Here we fixed $\alpha = 0.01$ and varied $V$ from 1 to 7 with an increment of 1. Seen from Figure 3.6, for both Yale and ORL, the accuracy with multiple components ($V \geq 2$) is always better than MCNMF with $V = 1$ (NMF). Specifically, the accuracy increases sharply when $V$ is tuned from 1 to 3, which indicates the effectiveness of MCNMF by exploring multiple components. Then the accuracy fluctuates slightly when $V$ increases from 3 to 7. The fluctuation could be due to a compromise between the amount of features for each representation and the diverse information among them. When $V$ increases, more diverse information can be utilized. However, given a fixed $k = \sum_{i=1}^{3} k^{(i)}$, the increase of $V$ will result in reduction of the feature dimension $k^{(i)}$ for each representation.

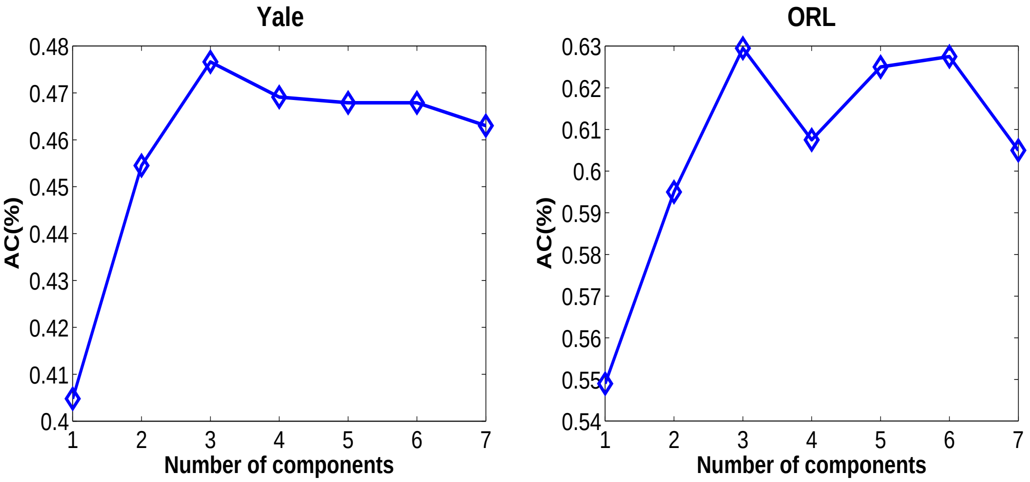


Figure 3.6: The effect of the number of components $V$.

**Convergence analysis**. Having proven the convergence of our update rules of MCNMF in previous sections, here we experimentally demonstrate its convergence in Figure 3.7, where the horizontal axis is the number of iterations and the vertical axis is the value of nmf22ective function. It can be seen that the nmf22ective function values are non-increasing and drop sharply within 5 iterations on both datasets.

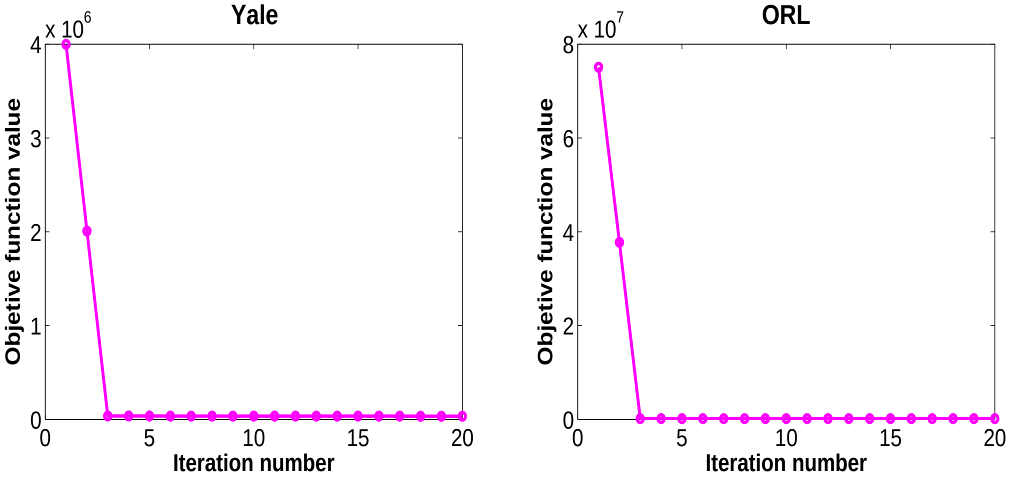


Figure 3.7: Convergence curves.

## 3.5 Conclusion

A Multi-Component Nonnegative Matrix Factorization (MCNMF) approach has been proposed to find multi-representation of data by exploring embedded latent components. Different from existing NMF-based approaches that seek for a single representation matrix, MCNMF learns multiple representations simultaneously. Utilizing Hilbert Schmidt Independence Criterion (HSIC) as a penalty term, MCNMF explicitly enforces the diversity of different data representations. A novel updating rule to optimize the nmf22ective function has been derived, together with correctness and convergence being proven. Extensive experiments have demonstrated that MCNMF can not only obtain multiple representations with each one reflecting one property of data, but also increases the accuracy by aggregating multiple representations. In fact, often real data such as motion sequences and video clips, are with ordered structure, i.e., consecutive neigh- bouring data samples are very likely share similar features unless a sudden change occurs. MCNMF deal with features of each data points independently, making it not proper for the analysis of such data. In the next chapter, a novel approach is proposed to capture the embedded ordered structure of data to enhance performance.

# Chapter 4

# Ordered Structured Preserved Nonnegative Matrix Factorization

## 4.1 Introduction

It is well recognized that NMF based approaches have been widely used in the fields of machine learning and computer vision such as motion segmentation [12, 72], human activity recognition [33], face recognition [70, 64, 49], etc. Often the data which these applications process are sequential, such as a video clip, a sequence of a subject's images taken under changing illuminations, etc. These data can be sampled such that consecutive samples are similar to each other unless a big or sudden change occurs. This sequential nature or ordered structure provides valuable information about the relationship between data [75, 91, 106, 36]. For example, to cluster frames of a video clip into scenes they belong to, the representations of the frames in the same scene based on the existing approaches could be quite different, due to the fact that the only the frame's characteristic features such as illumination or perspective are utilized. Instead, if the ordered structure is incorporated as a constraint, these differences are reduced because the representations of every two neighbouring frames are enforced to be similar which will improve the clustering accuracy. Thus, exploiting the ordered structure with NMF holds a great potential for seeking for optimal representations.

However, the approaches introduced previously are specific to data that data samples and features are independently distributed. In other words, they are not able to exploit the sequential relationship and extremely challenging to find optimal representations of sequential data. Therefore in this chapter we propose and develop a novel method, named as ordered robust nonnegative matrix factorization (ORNMF), which takes full advantage of the relationship and sequential correlation. A novel neighbour penalty term is constructed to enforce the similarity of the consecutive data representations. A $L_{2,1}$-norm loss function is used to improve the robustness so that ORNMF is insensitive to the data outliers and applicable to applications with noisy data. An efficient and elegant iterative updating rule is derived and analyzed theoretically to demonstrate their correctness and convergence. The experiments on one synthetic and two real datasets, in comparison with both baselines and state-of-the-art methods, have demonstrated the superiority of ORNMF in terms of accuracy and normalized mutual information.

## 4.2 Ordered Robust NMF (ORNMF)

ORNMF is proposed in this study to enforce the similarity between representations of neighbouring data. The inspiration behind ORNMF is that the changes between neighbouring data are usually very subtle, so the representations of these data should be similar to each other. Taken a video sequence for an example, since the scenes in the sequence normally change much less frequently than the frame rate, it is safe to assume that a high similarity exists among consecutive frames, except when two neighbouring frames are from different scenes.

To achieve the optimal data representations by incorporating this ordered structure, a novel regularization term is incorporated to the conventional NMF objective function in two steps. First, we construct the following matrix $\mathbf{R} \in \mathbb{R}^{n\times(n-1)}$, which is a lower triangular matrix with $-1$ on the diagonal and 1 on the second diagonal:

$$\mathbf{R} = \begin{bmatrix} -1 & 0 & 0 & \dots & 0 \\ 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & \ddots & -1 \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}.$$

Multiplying $\mathbf{H}$ by $\mathbf{R}$ gives $\mathbf{HR} = [\mathbf{H}_2 - \mathbf{H}_1, \mathbf{H}_3 - \mathbf{H}_2, \mathbf{H}_4 - \mathbf{H}_3 \dots \mathbf{H}_n - \mathbf{H}_{n-1}]$. If the columns of $\mathbf{HR}$ are or nearly equal to zero vectors, i.e. $\mathbf{H}_i - \mathbf{H}_{i-1} \approx 0$, data must be from the same subject/scene because they are similar, or a boundary or sudden change exists inbetween. Given $k$ subjects, ideally, only $k-1$ non-zero columns should $\mathbf{HR}$ have. To guarantee $k-1$ non-zeros columns, we introduce a $L_{2,0}$-norm, $\|\cdot\|_{2,0}$, to penalise each column directly and maintain the sparsity of $\mathbf{HR}$. The quasi-norm $L_{2,0}$-norm is defined as the number of non-zero columns. We thereby propose an objective function as

$$\min_{\mathbf{W}\geq 0, \mathbf{H}\geq 0} J = \|\mathbf{X} - \mathbf{WH}\|_F^2 + \alpha\|\mathbf{HR}\|_{2,0}, \tag{4.1}$$

where $\alpha$ is a trade-off parameter that controls the weight of the regularization term.

However, solving the problem (4.1) is NP-hard because of the $L_{2,0}$-norm [76]. According to [76], the $L_{2,1}$-norm of a given matrix $\mathbf{X}$, i.e., $\|\mathbf{X}\|_{2,1}$, is the minimum convex hull of $\|\mathbf{X}\|_{2,0}$. When $\mathbf{X}$ is column-sparse enough, namely, many zero columns are involved, minimize $\|\mathbf{X}\|_{2,1}$ is always equivalent to minimize $\|\mathbf{X}\|_{2,0}$. Therefore, we can relax the objective function (4.1) as:

$$\min_{\mathbf{W}\geq 0, \mathbf{H}\geq 0} J = \|\mathbf{X} - \mathbf{WH}\|_F^2 + \alpha\|\mathbf{HR}\|_{2,1}. \tag{4.2}$$

Since the error, i.e. the first term of (4.2) is squared, a few big ones due to outliers or noises may dominate the objective function. As in [50], we then propose a more robust function as the following:

$$\min_{\mathbf{W}\geq 0,\mathbf{H}\geq 0} J = \|\mathbf{V}-\mathbf{WH}\|_{2,1} + \alpha\|\mathbf{HR}\|_{2,1}, \tag{4.3}$$

where the $L_{2,1}$-norm is applied to the loss function and defined as $\|\mathbf{X}-\mathbf{WH}\|_{2,1} = \sum_{i=1}^{n}\|\mathbf{X}_i - \mathbf{WH}_i\|$. With the error for each data not being squared, the impact of large errors is reduced significantly.

### 4.2.1 Optimization

Since the optimization problem in (4.3) is not convex in both variables $\mathbf{W}$ and $\mathbf{H}$, it is infeasible to find the global minimum. In addition, as the matrix $\mathbf{R}$ contains negative values, it is technically challenging to solve (4.3) directly. Following [55], here we propose an algorithm that iteratively updates $\mathbf{H}$ with $\mathbf{W}$ fixed and then $\mathbf{W}$ with $\mathbf{H}$ fixed, which guarantees the objective function values do not increase with iterations.

**Update for H**: To update $\mathbf{H}$ with $\mathbf{W}$ fixed, we need to solve the following problem:

$$\min_{\mathbf{H}\geq 0} J(\mathbf{H}) = \|\mathbf{X}-\mathbf{WH}\|_{2,1} + \alpha\|\mathbf{HR}\|_{2,1}. \tag{4.4}$$

We introduce a Lagrange multiplier matrix $\boldsymbol{\eta} = [\eta_{ij}] \in \mathbb{R}^{k\times n}$ for the constraint $\mathbf{H} \geq 0$, then we have the following equivalent objective function:

$$\begin{aligned} J(\mathbf{H}) = {} & tr(\mathbf{XD}_1\mathbf{X}^T - 2\mathbf{XD}_1\mathbf{H}^T\mathbf{W}^T + \mathbf{WHD}_1\mathbf{H}^T\mathbf{W}^T) \\ & + \alpha tr(\mathbf{HRD}_2\mathbf{R}^T\mathbf{H}^T) + tr(\boldsymbol{\eta}\mathbf{H}). \end{aligned} \tag{4.5}$$

where $\mathbf{D}_1$ and $\mathbf{D}_2$ are diagonal matrices with the diagonal elements being

$$(\mathbf{D_1})_{ii} = \frac{1}{\|\mathbf{X}_i - \mathbf{WH}_i\|}, i = 1, 2..., n. \tag{4.6}$$

$$(\mathbf{D_2})_{ii} = \frac{1}{\|(\mathbf{HR})_i\|}, i = 1, 2..., n-1. \tag{4.7}$$

Setting the derivative of $J(\mathbf{H})$ to be 0 with respect to $\mathbf{H}$, we have

$$\boldsymbol{\eta} = 2\mathbf{W}^T\mathbf{XD}_1 - 2\mathbf{W}^T\mathbf{WHD}_1 - 2\alpha\mathbf{HRD_2R}^T, \tag{4.8}$$

Following the Karush-Kuhn-Tucker (KKT) condition [5] $\eta_{ij}\mathbf{H}_{ij} = 0$, we have

$$(\mathbf{W}^T\mathbf{X}\mathbf{D}_1 - \mathbf{W}^T\mathbf{W}\mathbf{H}\mathbf{D}_1 - \alpha\mathbf{H}\mathbf{R}\mathbf{D}_2\mathbf{R}^T)_{ij}\mathbf{H}_{ij} = 0. \tag{4.9}$$

Because $\mathbf{R}$ contains negative values, we decompose $\mathbf{R}$ into two nonnegative parts for ensuring $\mathbf{H} \geq 0$ in each iteration:

$$\mathbf{R} = \mathbf{R}^+ - \mathbf{R}^-, \tag{4.10}$$

where $\mathbf{R}^+_{ij} = (|\mathbf{R}_{ij}| + \mathbf{R}_{ij})/2$ and $\mathbf{R}^-_{ij} = (|\mathbf{R}_{ij}| - \mathbf{R}_{ij})/2$. Substituting (4.10) into (4.9), we obtain

$$\begin{aligned}&(\mathbf{W}^T\mathbf{X}\mathbf{D}_1 - \mathbf{W}^T\mathbf{W}\mathbf{H}\mathbf{D}_1 + \alpha\mathbf{H}(\mathbf{R}^+\mathbf{D}_2\mathbf{R}^{-T} + \mathbf{R}^-\mathbf{D}_2\mathbf{R}^{+T})\\ &- \alpha\mathbf{H}(\mathbf{R}^+\mathbf{D}_2\mathbf{R}^{+T} + \mathbf{R}^-\mathbf{D}_2\mathbf{R}^{-T}))_{ij}\mathbf{H}_{ij} = 0.\end{aligned} \tag{4.11}$$

Denoting $\mathbf{R}_a = \mathbf{R}^+\mathbf{D}_2(\mathbf{R}^-)^T$, $\mathbf{R}_b = \mathbf{R}^-\mathbf{D}_2(\mathbf{R}^+)^T$, $\mathbf{R}_c = \mathbf{R}^+\mathbf{D}_2\mathbf{R}^{+T}$, $\mathbf{R}_d = \mathbf{R}^-\mathbf{D}_2\mathbf{R}^{-T}$, we then have the following successive update of $\mathbf{H}$ with an initial value of $\mathbf{H}$.

$$\mathbf{H}_{ij} \leftarrow \mathbf{H}_{ij}\sqrt{\frac{(\mathbf{W}^T\mathbf{X}\mathbf{D}_1 + \alpha\mathbf{H}(\mathbf{R}_a + \mathbf{R}_b))_{ij}}{(\mathbf{W}^T\mathbf{W}\mathbf{H}\mathbf{D}_1 + \alpha\mathbf{H}(\mathbf{R}_c + \mathbf{R}_d))_{ij}}}. \tag{4.12}$$

When (4.12) converges, its solution satisfies (4.11).

This updating rule of $\mathbf{H}$ satisfies the following theorem, which guarantees the correctness of the rule.

**Theorem 1**. If the updating rule of $\mathbf{H}$ converges, then the final solution satisfies the KKT optimality condition.

*Proof of Theorem 1.* At convergence, $\mathbf{H}^\infty = \mathbf{H}^{t+1} = \mathbf{H}^t = \mathbf{H}$, where t denotes the $t$-th iteration, i.e.,

$$\mathbf{H}_{ij} = \mathbf{H}_{ij}\sqrt{\frac{(\mathbf{W}^T\mathbf{X}\mathbf{D}_1 + \alpha\mathbf{H}(\mathbf{R}_a + \mathbf{R}_b))_{ij}}{(\mathbf{W}^T\mathbf{W}\mathbf{H}\mathbf{D}_1 + \alpha\mathbf{H}(\mathbf{R}_c + \mathbf{R}_d))_{ij}}}. \tag{4.13}$$

This is the same as

$$\begin{aligned}&(\mathbf{W}^T\mathbf{X}\mathbf{D}_1 - \mathbf{W}^T\mathbf{W}\mathbf{H}\mathbf{D}_1 + \alpha\mathbf{H}(\mathbf{R}^+\mathbf{D}_2\mathbf{R}^{-T} + \mathbf{R}^-\mathbf{D}_2\mathbf{R}^{+T})\\ &- \alpha\mathbf{H}(\mathbf{R}^+\mathbf{D}_2\mathbf{R}^{+T} + \mathbf{R}^-\mathbf{D}_2\mathbf{R}^{-T}))_{ij}\mathbf{H}^2_{ij} = 0.\end{aligned} \tag{4.14}$$

which is equivalent to (4.11). We now prove the convergence of the updating rule. To achieve this goal, following [56], we use an auxiliary function as following .

**Definition 1** [56] A function $G(\mathbf{H}, \mathbf{H}')$ is an auxiliary function of the function $J(\mathbf{H})$ if $G(\mathbf{H}, \mathbf{H}') \geq J(\mathbf{H})$ and $G(\mathbf{H}, \mathbf{H}) = J(\mathbf{H})$ for any $\mathbf{H}$ and a constant matrix $\mathbf{H}'$.

The auxiliary function helps because of the following lemma:

**Lemma 1** [56] If $G$ is an auxiliary function of $J$, then $J$ is non-increasing under the updating rule $\mathbf{H}^{t+1} = \arg\min_{\mathbf{H}} G(\mathbf{H}, \mathbf{H}^t)$.

*Proof.* $J(\mathbf{H}^{t+1}) \leq G(\mathbf{H}^{t+1}, \mathbf{H}^t) \leq G(\mathbf{H}^t, \mathbf{H}^t) = J(\mathbf{H}^t)$

Now we have the specific form of the auxiliary function $G(\mathbf{H}, \mathbf{H}')$ for the objective function $J(\mathbf{H})$ in the problem (4.4), based on the following lemma.

**Lemma 2** The function

$$\begin{aligned} G(\mathbf{H}, \mathbf{H}') = &-2\sum_{ij}(\mathbf{W}^T\mathbf{X}\mathbf{D}_1)_{ij}\mathbf{H}'_{ij}(1 + \log\frac{\mathbf{H}_{ij}}{\mathbf{H}'_{ij}}) \\ &+ \sum_{ij}\frac{(\mathbf{W}^T\mathbf{W}\mathbf{H}'\mathbf{D}_1)_{ij}\mathbf{H}^2_{ij}}{\mathbf{H}'_{ij}} \\ &- \sum_{ijk}((\mathbf{R}_a + \mathbf{R}_b)_{jk})\mathbf{H}'_{ij}\mathbf{H}'_{ik}(1 + \log\frac{\mathbf{H}_{ij}\mathbf{H}_{ik}}{\mathbf{H}'_{ij}\mathbf{H}'_{ik}}) \\ &+ \sum_{ij}\frac{(\mathbf{H}'(\mathbf{R}_c + \mathbf{R}_d))_{ij}\mathbf{H}^2_{ij}}{\mathbf{H}'_{ij}} \end{aligned} \tag{4.15}$$

is an auxiliary function for $J(\mathbf{H})$ in problem (4.4).

*Proof of Lemma 2.* We find upper bounds for each of the two positive terms by the following lemma,

**Lemma 3** [20]. For any nonnegative matrices $\mathbf{S} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{g\times g}$, $\mathbf{F} \in \mathbb{R}^{n\times g}$ and $\mathbf{F}' \in \mathbb{R}^{n\times g}$, with $\mathbf{S}$ and $\mathbf{B}$ are symmetric, then the following inequality holds

$$tr(\mathbf{F}^T\mathbf{S}\mathbf{F}\mathbf{B}) \leq \sum_{i=1}^{n}\sum_{p=1}^{g}(\mathbf{S}\mathbf{F}'\mathbf{B})\frac{\mathbf{F}^2{}_{ip}}{\mathbf{F}'{}_{ip}}. \tag{4.16}$$

Then, we have following inequations:

$$tr(\mathbf{W}^T\mathbf{W}\mathbf{H}\mathbf{D}_1\mathbf{H}^T) \leq \sum_{ij}\frac{(\mathbf{W}^T\mathbf{W}\mathbf{H}'\mathbf{D}_1)_{ij}\mathbf{H}^2_{ij}}{\mathbf{H}'_{ij}}, \tag{4.17}$$

$$tr(\mathbf{H}(\mathbf{R}_c+\mathbf{R}_d)\mathbf{H}^T) \le \sum_{ij} \frac{(\mathbf{H}'(\mathbf{R}_c+\mathbf{R}_d))_{ij}\mathbf{H}^2_{ij}}{\mathbf{H}'_{ij}}. \tag{4.18}$$

To obtain lower bounds for the remaining terms, we use the inequality $z > 1+\log z, \forall z > 0$ [20] and have

$$\begin{aligned} & tr(\mathbf{W}^T\mathbf{X}\mathbf{D}_1\mathbf{H}^T) \\ & \ge \sum_{ij}(\mathbf{W}^T\mathbf{X}\mathbf{D}_1)_{ij}\mathbf{H}'_{ij}(1+\log\frac{\mathbf{H}_{ij}}{\mathbf{H}'_{ij}}), \end{aligned} \tag{4.19}$$

$$\begin{aligned} & tr(\mathbf{H}(\mathbf{R}_a+\mathbf{R}_b)\mathbf{H}^T) \\ & \ge \sum_{ijk}(\mathbf{R}_a+\mathbf{R}_b)_{jk}\mathbf{H}'_{ij}\mathbf{H}'_{ik}(1+\log\frac{\mathbf{H}_{ij}\mathbf{H}_{ik}}{\mathbf{H}'_{ij}\mathbf{H}'_{ik}}). \end{aligned} \tag{4.20}$$

Collecting all bounds, we have the final auxiliary function in **Lemma 2**.

Based on Lemmas 1 and 2, we can show the convergence of the updating rule (4.12).

**Theorem 2**. The problem (4.4) is non-increasing under the iterative updating rule (4.12).

*Proof of Theorem 2.* **Lemma 2** provides a specific form $G(\mathbf{H},\mathbf{H}')$ of the auxiliary function for $J(\mathbf{H})$ in problem (4.4). We can have the solution for $\min_{\mathbf{H}} G(\mathbf{H},\mathbf{H}')$ by the following KKT condition

$$\begin{aligned} \frac{\partial G(\mathbf{H},\mathbf{H}')}{\partial \mathbf{H}_{ij}} & = -2(\mathbf{W}^T\mathbf{X}\mathbf{D}_1)_{ij}\frac{\mathbf{H}'_{ij}}{\mathbf{H}_{ij}} + 2\frac{(\mathbf{W}^T\mathbf{W}\mathbf{H}'\mathbf{D}_1)_{ij}\mathbf{H}_{ij}}{\mathbf{H}'_{ij}} \\ & - 2\frac{(\mathbf{H}'(\mathbf{R}_a+\mathbf{R}_b))_{ij}\mathbf{H}'_{ij}}{\mathbf{H}_{ij}} + 2\frac{(\mathbf{H}'(\mathbf{R}_c+\mathbf{R}_d))_{ij}\mathbf{H}_{ij}}{\mathbf{H}'_{ij}} = 0, \end{aligned} \tag{4.21}$$

which gives rise to the updating rule in (4.12). Following **Lemma 1**, under this updating rule the objective function values of $J(\mathbf{H})$ in (4.4) will be non-increasing.

**Update for W**: To update **W** with **H** fixed, we need to solve the following problem:

$$\min_{\mathbf{W}\ge 0} J(\mathbf{W}) = \|\mathbf{X}-\mathbf{W}\mathbf{H}\|_{2,1} \tag{4.22}$$

This is exactly same as that in [50]. So we have the following updating rule for (4.22).

$$\mathbf{W}_{di} \leftarrow \mathbf{W}_{di}\frac{(\mathbf{X}\mathbf{D}_1\mathbf{H}^T)_{di}}{(\mathbf{W}\mathbf{H}\mathbf{D}_1\mathbf{H}^T)_{di}}. \tag{4.23}$$

Table 4.1: Comparison of Clustering Results (%) on Synthetic Data

| | Noises | Ncut | NMF | RNMF | GNMF | GNMF$_{KL}$ | OM-RPCA | OM-CRPCA | NMFi | ORNMF |
|---|---|---|---|---|---|---|---|---|---|---|
| AC | 0% | 100 | 84.38 | 86.25 | 100 | 100 | 100 | 100 | 96.25 | **100** |
| | 20% | 100 | 100 | 83.57 | 83.13 | 85.00 | 93.50 | 100 | 94.38 | **100** |
| | 50% | 96.06 | 96.88 | 96.25 | 100 | 81.25 | 96.25 | 96.37 | 98.13 | **100** |
| NMI | 0% | 100 | 91.67 | 91.67 | 100 | 100 | 100 | 100 | 92.75 | **100** |
| | 20% | 100 | 100 | 91.67 | 91.67 | 91.67 | 96.67 | 100 | 89.96 | **100** |
| | 50% | 95.65 | 95.66 | 95.18 | 100 | 91.67 | 98.33 | 98.67 | 96.36 | **100** |

More details on the correctness analysis and convergence proof of (4.23) can be found in [50].

The details of the algorithm is described in Algorithm 4.1.

**Algorithm 4.1** The algorithm of ORNMF

```
Input:
    The sequential data matrix $\mathbf{X}$
    The constructed matrix $\mathbf{R}$
    The parameter $\alpha$
Output:
    The data representation matrix $\mathbf{H}$
Initialize $\mathbf{W}$ and $\mathbf{H}$
while not converges do
    Decompose $\mathbf{R}$ into two nonnegative parts by (4.10)
    Calculate the diagonal matrices $\mathbf{D}_1$ and $\mathbf{D}_2$ by (4.6) and (4.7)
    Fixing $\mathbf{W}$, update $\mathbf{H}$ by (4.12)
    Fixing $\mathbf{H}$, update $\mathbf{W}$ by (4.23)
end while
```

### 4.2.2 Complexity analysis

Based on (4.12) and (4.23), we estimate the number of operations for each iteration. When we update $\mathbf{H}$, the cost of multiplications for $\mathbf{W}^T\mathbf{X}\mathbf{D}_1$, $\mathbf{H}(\mathbf{R}_a + \mathbf{R}_b)$, $\mathbf{W}^T\mathbf{W}\mathbf{H}\mathbf{D}_1$ and $\mathbf{H}(\mathbf{R}_c + \mathbf{R}_d)$ are $\mathcal{O}(kmn + kn^2)$, $\mathcal{O}(kn^2)$, $\mathcal{O}(mk^2 + nk^2 + kn^2)$ and $\mathcal{O}(kn^2)$, respectively. And $\mathbf{R}_a$, $\mathbf{R}_b$, $\mathbf{R}_c$ and $\mathbf{R}_d$ have computational complexity of $\mathcal{O}(n^3)$ each. So the overall cost for $\mathbf{H}$ is $\mathcal{O}(n^3 + kmn)$ as we usually set $k \ll \min(m, n)$; similarly, the cost for $\mathbf{W}$ is $\mathcal{O}(kn^2 + mnk)$. Nevertheless, $\mathbf{D}_1$,

$\mathbf{D}_2$, $\mathbf{R}_a$, $\mathbf{R}_b$, $\mathbf{R}_c$ and $\mathbf{R}_d$ are sparse matrices. The overall complexity for $\mathbf{H}$ and $\mathbf{W}$ can be greatly reduced with sparse matrices multiplication. Besides, many optimized libraries for matrix multiplication[1], such as OpenBLAS[2], are currently available to further speed up the computation.

## 4.3 Experiments

We conduct experiments on four datasets including one synthetic dataset and three real-world datasets to demonstrate ORNMF's performance and compare it with a few state-of-the-art approaches. The synthetic data is used to present and validate the ordered data representations with ORNMF. The Yale dataset[3] is to test ORNMF's performances against benchmark data with quasi sequential nature. The video sequence dataset [106] that consists of two short videos is to evaluate ORNMF's effectiveness on handling the sequential data. For each experiment, the parameter $\alpha$ of ORNMF in (4.3) is tuned within [0.1, 0.7]. The corresponding parameters of all competing methods (as listed below) are tuned for their best performances. $k$-means is applied on the obtained new data representation matrix $\mathbf{H}$ and repeated 20 times to produce the average performances.

### 4.3.1 Methods to compare

1. Standard normalized cut (Ncut) in [86].

2. Nonnegative Matrix Factorization minimizing F-norm cost [55].

3. Robust Nonnegative Matrix Factorization (RNMF) [50]: This is a robust formulation of NMF which adopts $L_{2,1}$-norm loss function to alleviate the noise problem.

4. Graph Regularized Nonnegative Matrix Factorization (GNMF) [6] which encodes the geometrical information of the data space into matrix factorization. It has two versions: GNMF minimizing F-norm cost and GNMF$_{KL}$ minimizing KL-divergence cost.

[1] https://github.com/attractivechaos/matmul

[2] http://www.openblas.net/

[3] http://cvc.yale.edu/projects/yalefaces/yalefaces.html.

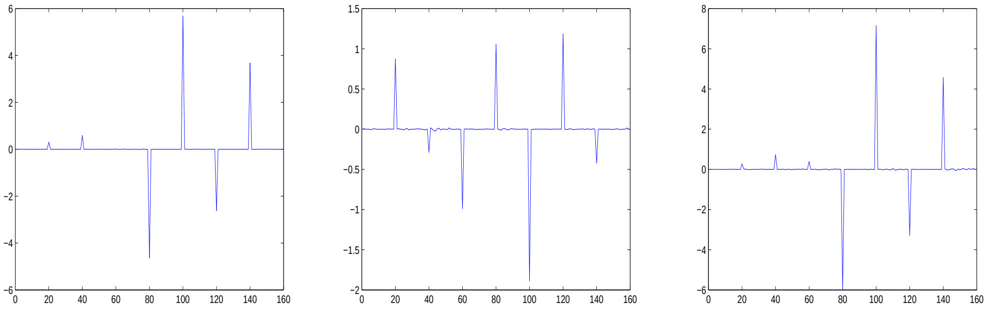
5. Optimal Mean Robust Principal Component Analysis (OMPCA) [77] which can correctly calculate the euclidean distance based mean of robust PCA. It has two implementations: OMPCA and OMCPCA.

6. Nonnegative Matrix Factorization with Interpolated Coefficients (NMFi) [13] which incorporates temporal constraint by adding a simple smoothness on the update rules of NMF.

7. Our proposed Ordered Robust Nonnegative Matrix Factorization(ORNMF).

### 4.3.2 Experiment on synthetic dataset

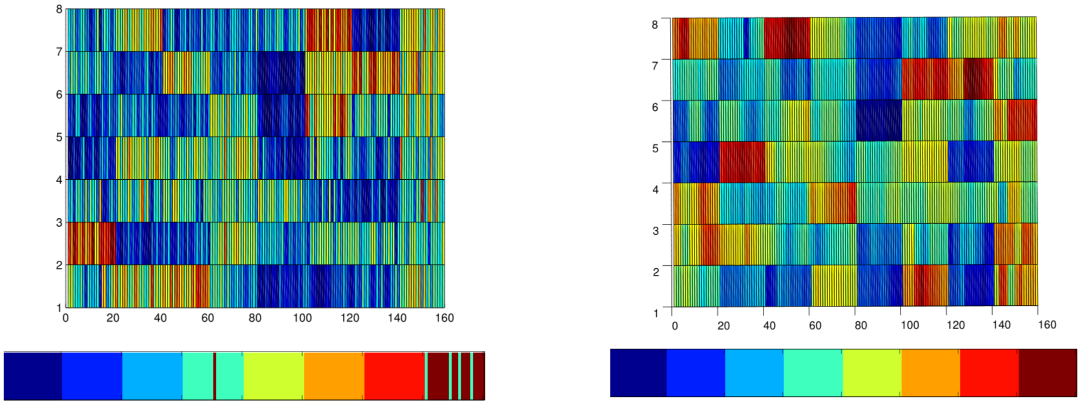
To build the dataset we first construct a data matrix $\mathbf{A} = [\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_8] \in \mathbb{R}^{400\times 8}$, in which each element of the data vector $\mathbf{A}_i, i \in \{1, 2, \ldots 8\}$ is a random number between 0 and 1, i.e., $\mathbf{A}_{ji} = [0, 1], j \in \{1, 2, \ldots 400\}$. Multiplying $\mathbf{A}$ with a uniform random weights $s_i \in \mathbb{R}^8$ forms a single synthetic data vector $\mathbf{X}_i$ $(=\mathbf{A}s_i)$. We then duplicate $\mathbf{X}_i$ 20 times to construct $\mathbf{X}^i = [\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_{20}] \in \mathbb{R}^{400\times 20}$. Repeating the progress for $\mathbf{X}^i$ 8 times with $\mathbf{A}$ being an invariant and combining all $\mathbf{X}^i$, we finally build our artificial data matrix $\mathbf{X} = [\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^8] \in \mathbb{R}^{400\times 160}$. The experiment is expected to group $\mathbf{X}$ into 8 clusters.



(a) NMF ($k$=8) (b) ORNMF ($k$=8) (c) ORNMF ($k$=50)

Figure 4.1: Comparison on inferring the number of clusters.

When data are clean, ORNMF is able to detect the cluster boundaries and infer the number of clusters, which can not be achieved by most NMF based methods. To demonstrate this, we calculate $\mathbf{HR} = [\mathbf{H}_2 - \mathbf{H}_1, \mathbf{H}_3 - \mathbf{H}_2, \ldots, \mathbf{H}_{160} - \mathbf{H}_{159}]$ after obtaining $\mathbf{H}$, and sum the values of each column of $\mathbf{HR}$ to find the peak values. The visualization results of NMF and ORNMF with clean data are shown in Figure 4.1. It can be seen that NMF in (a) achieves 6 peak values indicating 7 clusters, which is incorrect as the predefined number of clusters is 8. On the contrary, ORNMF finds 8 clusters according to the number of significant peak values as shown in (b), since all the columns in **HR** are nearly zeros but the boundaries. To demonstrate the robustness of ORNMF to $k$, we then randomly chose $k = 50$ and reported result in (c). As we can see, ORNMF can also find 8 clusters. As a result, ORNMF can correctly find the cluster boundaries and get the number of clusters regardless of the value of $k$. Nevertheless, in case the number of clusters is known beforehand or data is noisy, $k$-means is still a good option to cluster the data.



(a) NMF

(b) ORNMF

Figure 4.2: Top figures in (a) and (b) represent the data representation matrix **H**. The horizontal is the number of data and the vertical represents the reduced dimensionality of each data, $k$. Every consecutive 20 data belong to one subject. Each bottom figure displays the clustering results, where different colors represent different clusters.

According to [91], to further test the robustness of ORNMF, we add 20% and 50% level of Gaussian noise with zero mean and unit variance onto **X** and then normalize the corresponding contaminated **X** between 0 and 1 to evaluate the performances. As shown in Table 4.1, although all methods have obtained promising results, only ORNMF achieves the perfect performances in all three cases.

In order to present the performances visually, Figure 4.2 illustrates the data representation matrix **H** and the corresponding clustering results of NMF and ORNMF when data come with 50% level of Gaussian noise. The data representations within each cluster of **H** in ORNMF are smooth, which implies that they are of high similarity despite of being contaminated by noises. This is inline with the expectation behind our proposed ORNMF. Hence **H** in ORNMF captures the ordered structure effectively, leading to the perfect segmentation result which NMF fails to achieve as shown in the bottom figures.

### 4.3.3 Face clustering

This experiment is to group a set of face images in the Yale dataset into different clusters. The dataset consists of 11 facial images of 15 subjects/clusters - total 165 grayscale images. Each image comes with different facial expression or configuration: center-light, w/glasses, happy, left-light, w/no glasses, normal, rightlight, sad, sleepy, surprised, and wink. Before clustering, images are preprocessed. First, we normalize the images in scale and orientation such that eyes are all aligned at the same position horizontally. Then, the facial areas were cropped into the final images for clustering.

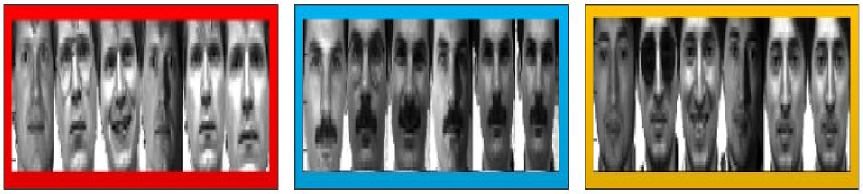


Figure 4.3: Samples of Yale Dataset. Different color indicates different clusters.

To reduce the computational cost and the memory requirements, all face images are downsized to 32 × 32 pixels with 256 gray levels per pixel as shown in Figure 4.3 for example. Thus, each image is represented by a data vector $\mathbf{X}_i \in \mathbb{R}^{1024}$ and we concatenate all these data vectors in order. Strictly speaking, these data are not sequential. However, since the similarities among images of the

Table 4.2: Comparison of Clustering Results (%) on Yale Dataset

| | $k$ | Ncut | NMF | RNMF | GNMF | GNMF$_{KL}$ | OM-RPCA | OM-CRPCA | NMFi | ORNMF |
|---|---|---|---|---|---|---|---|---|---|---|
| AC | 2 | 71.82 | 78.64 | 90.91 | 86.36 | 86.36 | 86.36 | 90.91 | 86.36 | **91.59** |
| | 3 | 57.27 | 66.36 | 66.97 | 60.61 | 60.61 | 75.76 | **75.76** | 63.64 | 69.70 |
| | 4 | 52.73 | 63.18 | 68.18 | 65.91 | 65.91 | 68.18 | 63.64 | 63.64 | **70.45** |
| | 5 | 51.27 | 58.19 | 68.00 | 65.45 | 67.27 | 61.82 | 67.27 | 69.09 | **74.55** |
| | 6 | 49.84 | 49.09 | 57.12 | 57.58 | 53.03 | 53.03 | 54.55 | 59.09 | **63.64** |
| | 7 | 39.39 | 44.25 | 52.07 | 50.89 | 50.05 | 44.27 | 57.65 | 50.65 | **52.92** |
| | 8 | 44.66 | 45.45 | 54.55 | 61.36 | 46.59 | 54.55 | 56.82 | 52.27 | **64.77** |
| | 9 | 43.78 | 43.34 | 49.44 | **57.11** | 48.28 | 35.54 | 44.29 | 54.55 | 51.16 |
| | 10 | 36.91 | 48.36 | 48.18 | 48.18 | 44.55 | 39.09 | 41.82 | 50.91 | **52.59** |
| | Avg. | 49.37 | 56.30 | 63.05 | 60.35 | 59.41 | 59.19 | 63.46 | 61.13 | **66.65** |
| NMI | 2 | 33.97 | 40.76 | 56.05 | 41.27 | 43.23 | 43.23 | 56.05 | 52.30 | **68.65** |
| | 3 | 32.75 | 37.69 | 40.32 | 37.76 | 37.76 | 43.30 | 43.30 | 41.25 | **52.60** |
| | 4 | 41.23 | 43.50 | 57.51 | 49.14 | 47.47 | 43.75 | 43.28 | 51.04 | **66.38** |
| | 5 | 43.74 | 42.39 | 52.89 | 39.14 | 49.36 | 44.25 | 52.08 | 62.66 | **62.94** |
| | 6 | 44.93 | 36.07 | 43.95 | 40.96 | 39.31 | 39.54 | 48.46 | 47.74 | **52.49** |
| | 7 | 39.39 | 44.25 | 52.07 | 50.89 | 50.05 | 44.27 | 57.65 | 44.55 | **52.92** |
| | 8 | 45.51 | 40.59 | 46.38 | 38.14 | 42.72 | 46.91 | 54.16 | 48.47 | **62.64** |
| | 9 | 43.78 | 43.34 | 49.44 | **57.11** | 48.28 | 35.54 | 44.29 | 53.07 | 51.16 |
| | 10 | 43.04 | 46.22 | 50.69 | 41.91 | 46.39 | 37.72 | 40.26 | 53.93 | **52.31** |
| | Avg. | 40.93 | 41.64 | 49.92 | 44.04 | 44.95 | 42.06 | 48.84 | 50.56 | **58.01** |

same subject are much stronger than those from different subjects, the dataset can be regarded as exhibiting a quasi sequential nature.

Similar to the experimental setting in [64], we conduct the experiments for each method on the different number of clusters from 2 to 10 to make a thorough comparison. For a fixed cluster number $k$, we randomly choose $k$ categories from the dataset, and mix the images of these $k$ categories as the collection $\mathbf{X}$ for clustering.

The clustering results of each $k$ and the overall average performances on all cases are reported in Table 4.2, in which it can be clearly seen that ORNMF significantly outperforms other methods in most cases. Specifically, for average results, compared to the second best method, ORNMF achieves 3.19% improvements in AC and a bigger margin of 7.45% in NMI.

We also test the effect of the parameter $\alpha$, which is first selected in a wide range and then changes within a relative robust range, i.e, from 0.1 to 0.7 with an increment of 0.1. For a clear presentation, Figure 4.4 illustrates the performances with even $k$ numbers only. It is easy to see that ORNMF produces excellent and relatively stable results, which demonstrates ORNMF is insensitive to $\alpha$.

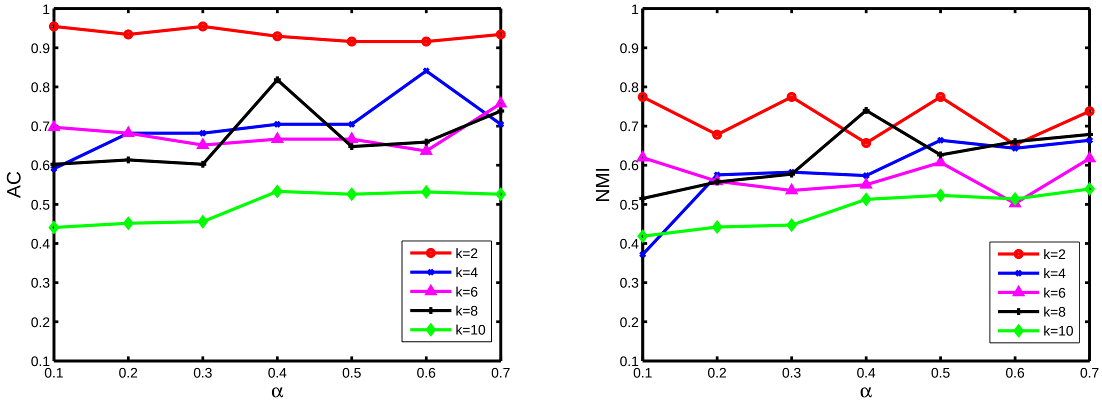


Figure 4.4: **Left**: Comparison of AC w.r.t $\alpha$. **Right**: Comparison of NMI w.r.t $\alpha$.

### 4.3.4 Video scene segmentation

We extract video sequences from two short animations available free from Internet, same as that in [106]. The videos 1 and 2 contain 19 and 24 sequences, respectively. Each sequence is about 10 s (approximately 300 frames), containing three scenes (that to be segmented). Those frames in which the scene changes are annotated manually and used as our ground truth data. Each sequence is then converted from color to grayscale and resized to a resolution of 129 × 96. The frames are vectorized to $\mathbf{X}_i \in \mathbb{R}^{12384}$ and concatenated in order to form $\mathbf{X}$ for segmentation. Figure 4.5 is an example of sequences. This experiment aims to cluster frames into the scene they belong to.

The experimental results on the two videos are shown in Table 4.3. ORNMF outperforms other methods consistently in both videos 1 and 2. For example, the improvements against RNMF are 1.51% and 4.1% in terms of AC and NMI

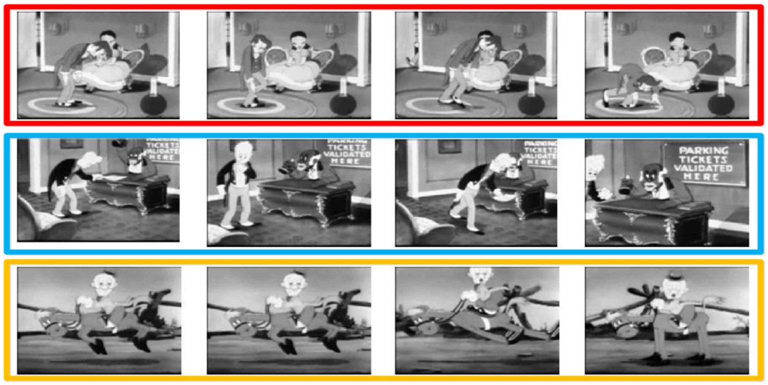
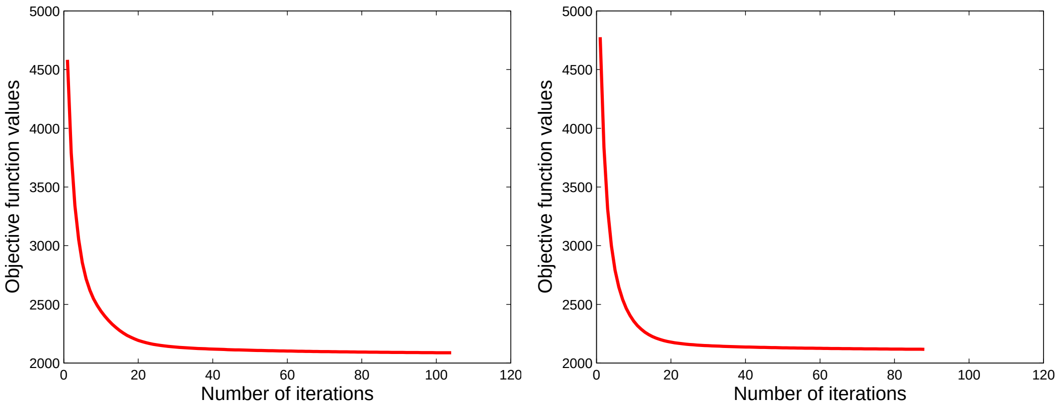


Figure 4.5: A sequence with three scenes from the video 1 marked by coloured borders.

in video 1; 6.68% AC and 5.93% NMI in video 2. This is due to the effectiveness of ORNMF in utilizing the ordered structure of video sequences. Because we use multiplicative updating rules to obtain the local optimum, it is important to analyze the convergence. Here we choose a sequence from the video 2 and compare the convergence speed of ORNMF and RNMF. The convergence criteria is $\frac{J_{t+1}-J_t}{J_t} < 10^{-4}$, where $J_t$ is the objective function value in $t$th iteration. The comparison in Figure 4.6 shows that the objective function values of ORNMF drop sharply in about 20 iterations and are non-increasing in the whole iterative procedure. And ORNMF takes about 90 iterations to finish the computation, which is 20 iterations less than RNMF. This demonstrates ORNMF converges effectively.

Table 4.3: Comparison of Clustering Results (%) on Video Sequences Dataset

| | | Ncut | NMF | RNMF | GNMF | GNMF$_{KL}$ | OM-RPCA | OM-CRPCA | NMFi | ORNMF |
|---|---|---|---|---|---|---|---|---|---|---|
| Video 1 | AC | 73.37 | 77.78 | 77.57 | 74.46 | 77.49 | 77.72 | 75.97 | 78.29 | **79.08** |
| | NMI | 60.96 | 66.65 | 65.33 | 63.48 | 67.60 | 69.40 | 66.98 | 66.29 | **69.43** |
| Video 2 | AC | 79.86 | 84.41 | 85.16 | 78.69 | 82.12 | 80.53 | 82.45 | 86.51 | **91.84** |
| | NMI | 70.31 | 76.76 | 77.95 | 63.21 | 76.12 | 73.44 | 72.68 | 76.83 | **83.88** |



(a) RNMF

(b) ORNMF

Figure 4.6: Comparison on convergence speed.

### 4.3.5 Human activity segmentation

The aim of this experiment is to segment activities in a sequence from the HDM05 Motion Capture Database [73]. The motion sequences were performed by five actors according to the guidelines specified in a script. The script consists of five parts, where each part is subdivided into several scenes. For this experiment we choose the scene 1-1 which contains 9842 frames and 14 activities. However, there is no frame by frame ground truth provided. We assembled the ground truth by watching the replay of the activities and manually labelling the activities using the activity list provided by [73]. We report clustering performances for this experiment in Table 4.4. It is clear to see that Ncut performs worst with 42.13% accuracy only, and all the other existing approaches achieve around 60% accuracies, while ORNMF gets more than 70% rate which outperforms other methods with a large margin. This well demonstrates the effectiveness of ORNMF.

Table 4.4: Comparison of Clustering Results (%) on HDM05 dataset

| | Ncut | NMF | RNMF | GNMF | GNMF$_{KL}$ | OM-RPCA | OM-CRPCA | NMFi | ORNMF |
|---|---|---|---|---|---|---|---|---|---|
| AC | 42.13 | 60.72 | 58.21 | 61.14 | 61.84 | 58.86 | 58.86 | 60.92 | **71.00** |
| NMI | 51.14 | 68.78 | 65.16 | 71.93 | 71.03 | 72.16 | 69.89 | 71.62 | **74.15** |

## 4.4 Conclusion

We have presented a novel approach, called ordered robust nonnegative matrix factorization (ORNMF) to exploit the ordered nature of sequential data. With a neighbour penalty term to enforce the similarity of data presentations, ORNMF has achieved more discriminative and explicit data representations. Using $L_{2,1}$-norm based loss function, ORNMF has effectively dealt with noisy data. A new iterative updating optimization scheme has been derived to solve ORNMF's objective function. In comparison to baselines (NMF, Ncut) and state-of-art approaches (RNMF, GNMF, OM-PCA), ORNMF has achieved the superior performances on both synthetic data, the benchmark dataset (Yale), video sequences and human activities (HDM05) in accuracy and normalized mutual information. ORNMF is a single-view approach which can only deal with a type of feature, such as pixels in images in our experiments. In reality, data are often collected from various sources or represented by different feature extractors, such as an image can be represented by different visual descriptors, such as SIFT [71], HOG [18] and GIST [79]. Therefore, how to apply NMF model in such situation is needed to be considered. In the following chapter, we focus on exploiting information of different aspects of data to enhance clustering under NMF framework.

# Chapter 5

# Diverse Multi-View Nonnegative Matrix Factorization

## 5.1 Introduction

This chapter studies NMF in multi-view setting by exploring diversity among different views. The diversity means that each view of data contains some distinct information that other views do not have. A main limitation of existing multi-view NMF-based approaches is that they all tend to exploit common information shared by multiple views but neglect the diversity among views, so that the learned data representations from multiple views contain mutually redundant information and lack diverse information. On the contrary, by taking the diversity into account, we can capture more information of data and achieve more comprehensive and accurate learning, because different views usually describe data from different aspects. Some researches [4, 11, 103] have also shown that the diversity is of importance to multi-view learning. Therefore, it is beneficial to integrate diversity properties of views into NMF learning.

To achieve this goal, we propose a novel Diverse Nonnegative Matrix Factorization (DiNMF) method. With a novel regularization term, DiNMF encourages the representations from multiple views to be diverse enough to capture comprehensive information, so that a diverse and more accurate data representation is eventually achieved. As illustrated in Figure 5.1, existing approaches (the upper figure) learn the data representations jointly to capture the underlying common

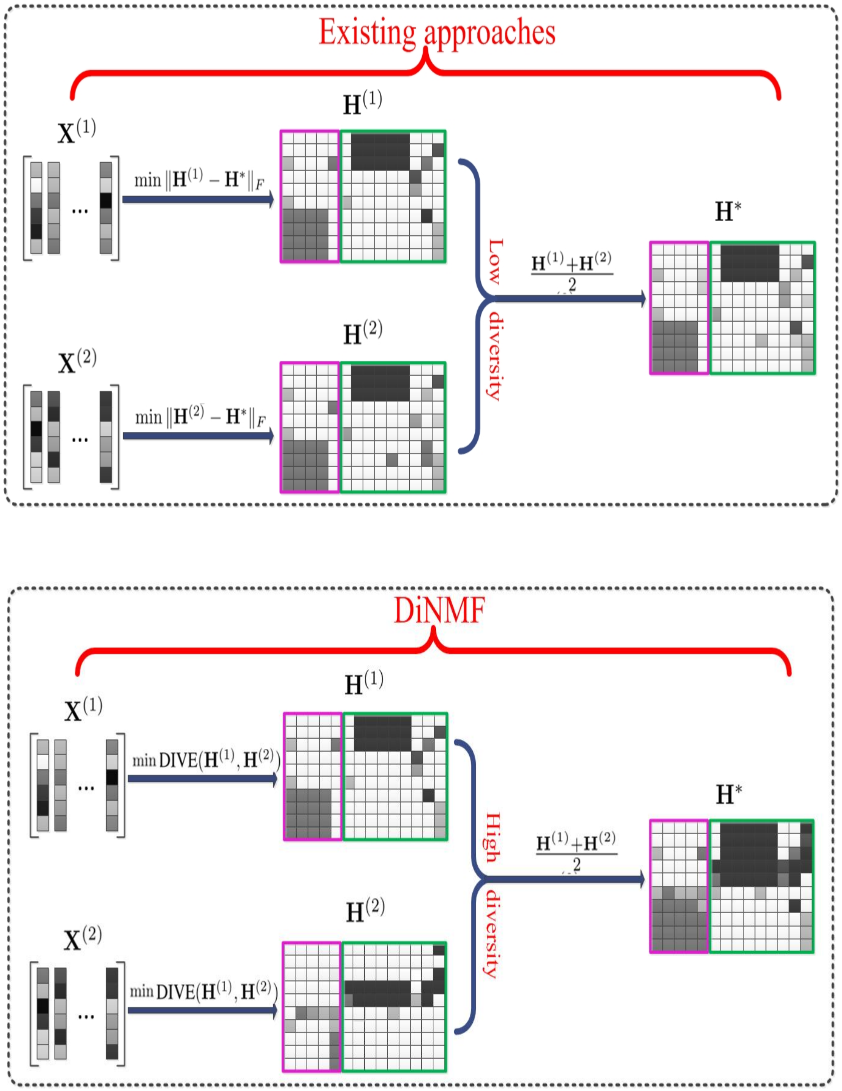

Existing approaches
$\mathbf{X}^{(1)}$
$\min \|\mathbf{H}^{(1)} - \mathbf{H}^{*}\|_F$
$\mathbf{H}^{(1)}$
Low diversity
$\frac{\mathbf{H}^{(1)}+\mathbf{H}^{(2)}}{2}$
$\mathbf{H}^{*}$
$\mathbf{X}^{(2)}$
$\min \|\mathbf{H}^{(2)} - \mathbf{H}^{*}\|_F$
$\mathbf{H}^{(2)}$
DiNMF
$\mathbf{X}^{(1)}$
$\min \mathrm{DIVE}(\mathbf{H}^{(1)}, \mathbf{H}^{(2)})$
$\mathbf{H}^{(1)}$
High diversity
$\frac{\mathbf{H}^{(1)}+\mathbf{H}^{(2)}}{2}$
$\mathbf{H}^{*}$
$\mathbf{X}^{(2)}$
$\min \mathrm{DIVE}(\mathbf{H}^{(1)}, \mathbf{H}^{(2)})$
$\mathbf{H}^{(2)}$

Figure 5.1: Comparison of existing NMF-based Multi-view approaches and the proposed DiNMF. A multi-view dataset $\mathbf{X}$ contains two equally important views, i.e., $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$. $\mathbf{H}^{(1)}$ and $\mathbf{H}^{(2)}$ are the corresponding learned representation matrices. $\mathbf{H}^{*}$ is the final representation. For all matrices, the data vectors are column-wise and the features are row-wise. The ground-truth is shown as group-1 in purple and group-2 in green. By enforcing $\mathbf{H}^{(1)}$ and $\mathbf{H}^{(2)}$ to be close to $\mathbf{H}^{*}$, the existing approaches learn the data representations of two views jointly to capture the shared underlying common information but cannot ensure their diversity. In contrast, DiNMF is based on a diversity term (DIVE), which captures diverse information among data representations. This ensures that $\mathbf{H}^{*}$ not only contains common information captured by existing approaches but also preserves some distinct information from each view, thus more comprehensive and accurate.

structure shared by two views. They enforce the feature distribution of $\mathbf{H}^{(1)}$ and $\mathbf{H}^{(2)}$ to be similar but fail to take advantage of distinct information of each view, which may lead to unsatisfactory results. It can be seen from the last columns of $\mathbf{H}^{(1)}$ and $\mathbf{H}^{(2)}$ that the feature distributions are nearly same and happen to be similar to columns in the group-1 (purple). Through linear computations, the corresponding column of $\mathbf{H}^{*}$ will be categorized into a wrong group, i.e., group-1, due to the similarity of feature distribution. On the contrary, DiNMF is based on a novel diversity constraint, i.e., DIVE, which enforces $\mathbf{H}^{(1)}$ and $\mathbf{H}^{(2)}$ to be as diverse as possible. As a result, $\mathbf{H}^{*}$ contains diverse information for comprehensive learning, since $\mathbf{H}^{(2)}$ captures some distinct information that $\mathbf{H}^{(1)}$ lacks. Moreover, the feature distributions of the two groups are more distinct in-between and this is in line with the ground truth, leading to more accurate learning. The main contributions of our work are as follows:

1. We propose a DiNMF approach which not only ensures the diversity to exploit comprehensive information but also reduces mutually redundancy across multiple representations for more accurate learning. Furthermore, DiNMF is also computationally linear thus has good scalability to large-scale datasets.

2. We further develop Locality Preserved DiNMF (LP-DiNMF) to preserve the locally geometrical structure of the manifolds for multi-view setting, by taking into account the manifold structures in data spaces. This leads to improved clustering accuracy compared with DiNMF.

3. We derive novel and efficient algorithms for both DiNMF and LP-DiNMF to optimize objective functions. The convergence of both algorithms are proved.

4. Experiments on both synthetic and real-world datasets from different domains demonstrate that the proposed methods are not only faster but also achieve more accurate clustering than other state-of-the-art methods.

## 5.2 Diverse NMF (DiNMF)

In this section, we first introduce a straightforward approach to extend the single-view NMF to multi-view setting. After that, we present DiNMF and propose an efficient optimization algorithm for solving the objective function.

It is well-known that traditional NMF aims to minimize the following objective function:

$$\|\mathbf{X} - \mathbf{W}\mathbf{H}\|_F^2, \quad s.t. \quad \mathbf{W}, \mathbf{H} \geq 0 \tag{5.1}$$

For the multi-view setting, we assume that $\mathbf{X}^{(v)} \in \mathbb{R}^{m^{(v)} \times n}$ be the feature matrix corresponding to the $v$th view. Similarly, $\mathbf{W}^{(v)}$ and $\mathbf{H}^{(v)}$ are the corresponding basis matrix and representation matrix, respectively. Thus, given $V$ heterogeneous features, we directly integrate all these features together so the objective function (5.1) becomes

$$\sum_{v=1}^{V} \|\mathbf{X}^{(v)} - \mathbf{W}^{(v)}\mathbf{H}^{(v)}\|_F^2. \quad s.t. \quad \mathbf{W}^{(v)}, \mathbf{H}^{(v)} \geq 0 \tag{5.2}$$

Obviously, this approach learns each data representation independently and cannot ensure the diversity of different views. To facilitate the subsequent discussion, we call this approach Non-diverse Multi-view Nonnegative Matrix Factorization (NdNMF).

### 5.2.1 Objective function

A desirable multi-view NMF approach for data analysis needs to satisfy two requirements. First, it should exploit diverse information across multi-view data representations for more comprehensive and accurate learning. Second, it is scalable since the number of data $n$ and dimension of features $m$ could be quite large. In the following, we describe how DiNMF satisfies these two requirements.

Diversity requires that two data vectors be as orthogonal to each other as possible, so that more comprehensive information can be exploited. Let $\mathbf{h}_i^{(v)}$ and $\mathbf{h}_i^{(w)}$ be the $i$th data representation vectors in two views, i.e, the $v$-th and $w$-th views. To ensure the diversity between the two vectors, their product should be 0, approximately. To achieve this, we can minimize the following function [35]

$$\|\mathbf{h}_i^{(v)} \circ \mathbf{h}_i^{(w)}\|_0, \tag{5.3}$$

where $\circ$ designates the element-wise product, and $\|\cdot\|_0$ is the $l^0$ norm which indicates the number of non-zero elements. Due to the non-convexity and discontinuity of $l^0$ norm, (5.3) can be relaxed by using $l^1$ norm as follows,

$$\|\mathbf{h}_i^{(v)} \circ \mathbf{h}_i^{(w)}\|_1 = \sum_{j=1}^{k} |h_{ji}^{(v)}| \cdot |h_{ji}^{(w)}|, \tag{5.4}$$

where $|\cdot|$ is the absolute value. Since the representations obtained by NMF are non-negative, we can further reformulate (5.4) as

$$\|\mathbf{h}_i^{(v)} \circ \mathbf{h}_i^{(w)}\|_1 = \sum_{j=1}^{k} h_{ji}^{(v)} \cdot h_{ji}^{(w)}. \tag{5.5}$$

By extending the calculation of single data vector in (5.5) to $n$ data vectors setting, we propose the following term to guarantee the diversity among all $n$ data vectors in two views,

$$\mathrm{DIVE}(\mathbf{H}^{(v)}, \mathbf{H}^{(w)}) = \sum_{i=1}^{n}\sum_{j=1}^{k} h_{ji}^{(v)} \cdot h_{ji}^{(w)} = tr(\mathbf{H}^{(v)}\mathbf{H}^{(w)^T}), \tag{5.6}$$

where $tr(\cdot)$ is the trace function. Therefore, minimizing (5.6) will encourage $\mathbf{H}^{(v)}$ and $\mathbf{H}^{(w)}$ to be orthogonal to each other. In other words, the diversity of the representation matrices in two views is guaranteed.

Given a dataset with more views, we incorporate the DIVE into NdNMF to guarantee that data representations in any two views be diverse. Then, the minimization objective function is produced as follows:

$$\begin{gathered}\sum_{v=1}^{V} \|\mathbf{X}^{(v)} - \mathbf{W}^{(v)}\mathbf{H}^{(v)}\|_F^2 + \alpha \sum_{v \neq w} \mathrm{DIVE}(\mathbf{H}^{(v)}, \mathbf{H}^{(w)}) \\ s.t. \quad 1 \leq v, w \leq V, \mathbf{W}^{(v)}, \mathbf{H}^{(v)}, \mathbf{H}^{(w)}, \alpha \geq 0,\end{gathered} \tag{5.7}$$

where $\alpha$ is a trade-off parameter which controls the weight of DIVE. A smooth regularization term $\|\mathbf{H}^{(v)}\|_F^2$ is added to avoid over-fitting of a view, which leads to the overall objective function as follows:

$$\sum_{v=1}^{V} \|\mathbf{X}^{(v)} - \mathbf{W}^{(v)}\mathbf{H}^{(v)}\|_F^2 + \alpha \sum_{v \neq w} \text{DIVE}(\mathbf{H}^{(v)}, \mathbf{H}^{(w)}) + \beta \sum_{v=1}^{V} \|\mathbf{H}^{(v)}\|_F^2 \tag{5.8}$$
$$s.t. \quad 1 \leq v, w \leq V, \mathbf{W}^{(v)}, \mathbf{H}^{(v)}, \mathbf{H}^{(w)}, \alpha, \beta \geq 0.$$

Here $\beta$ is the weight factor of the smoothness term.

To solve the objective function (5.8), we develop an efficient optimization algorithm to find the optimal solution of $\mathbf{H}^{(v)}$. After that, we calculate the average value of $\mathbf{H}^{(v)}$ in all views for the final multi-view data representation $\mathbf{H}^*$, i.e., $\mathbf{H}^* = \frac{\sum_{v=1}^{V} \mathbf{H}^{(v)}}{V}$. Following are the details.

### 5.2.2 Optimization

Since the objective function (5.8) is not convex with both variables $\mathbf{W}^{(v)}$ and $\mathbf{H}^{(v)}$, it is infeasible to find the global minimum. Instead, we propose an algorithm to find a local minima by iteratively updating $\mathbf{W}^{(v)}$ with $\mathbf{H}^{(v)}$ fixed and then updating $\mathbf{H}^{(v)}$ with $\mathbf{W}^{(v)}$ fixed.

For each view, the computations of $\mathbf{W}^{(v)}$ and $\mathbf{H}^{(v)}$ are not dependent on other views, so minimizing (5.8) gives us

$$\begin{aligned} &\|\mathbf{X}^{(v)} - \mathbf{W}^{(v)}\mathbf{H}^{(v)}\|_F^2 + \alpha \sum_{w=1;w\neq v}^{V} tr(\mathbf{H}^{(v)}{\mathbf{H}^{(w)}}^T) + \beta\|\mathbf{H}^{(v)}\|_F^2 \\ &= tr(\mathbf{X}^{(v)}{\mathbf{X}^{(v)}}^T - 2\mathbf{X}^{(v)}{\mathbf{H}^{(v)}}^T{\mathbf{W}^{(v)}}^T + \mathbf{W}^{(v)}\mathbf{H}^{(v)}{\mathbf{H}^{(v)}}^T{\mathbf{W}^{(v)}}^T) \\ &+ \alpha \sum_{w=1,w\neq v}^{V} tr(\mathbf{H}^{(v)}{\mathbf{H}^{(w)}}^T) + \beta tr(\mathbf{H}^{(v)}{\mathbf{H}^{(v)}}^T). \end{aligned} \tag{5.9}$$

Let $\eta_{ij}^{(v)}$ and $\xi_{ij}^{(v)}$ be the Lagrange multipliers for the constraint $w_{ij}^{(v)} \geq 0$ and $h_{ij}^{(v)} \geq 0$, respectively, and $\boldsymbol{\eta}^{(v)} = [\eta_{ij}^{(v)}]$, $\boldsymbol{\xi}^{(v)} = [\xi_{ij}^{(v)}]$, then the Lagrange function

$L$ of (5.9) is

$$\begin{aligned} L = tr(&\mathbf{X}^{(v)}{\mathbf{X}^{(v)}}^T - 2\mathbf{X}^{(v)}{\mathbf{H}^{(v)}}^T{\mathbf{W}^{(v)}}^T \\ &+ \mathbf{W}^{(v)}\mathbf{H}^{(v)}{\mathbf{H}^{(v)}}^T{\mathbf{W}^{(v)}}^T) + \alpha \sum_{w=1,w\neq v}^{V} tr(\mathbf{H}^{(v)}{\mathbf{H}^{(w)}}^T) \\ &+ \beta tr(\mathbf{H}^{(v)}{\mathbf{H}^{(v)}}^T) + tr(\boldsymbol{\eta}^{(v)}\mathbf{W}^{(v)}) + tr(\boldsymbol{\xi}^{(v)}\mathbf{H}^{(v)}). \end{aligned} \tag{5.10}$$

Setting the derivative of $L$ to be 0 with respect to $\mathbf{W}^{(v)}$ and $\mathbf{H}^{(v)}$, we have

$$\boldsymbol{\xi} = 2{\mathbf{W}^{(v)}}^T\mathbf{X}^{(v)} - 2{\mathbf{W}^{(v)}}^T\mathbf{W}^{(v)}\mathbf{H}^{(v)} - \alpha \sum_{w=1,w\neq v}^{V} \mathbf{H}^{(w)} - 2\beta\mathbf{H}^{(v)}, \tag{5.11}$$

and

$$\boldsymbol{\eta} = 2{\mathbf{W}^{(v)}}^T\mathbf{X}^{(v)} - 2{\mathbf{W}^{(v)}}^T\mathbf{W}^{(v)}\mathbf{H}^{(v)}. \tag{5.12}$$

Following the Karush-Kuhn-Tucker (KKT) condition [5] $\eta_{ij}^{(v)}w_{ij}^{(v)} = 0$ and $\xi_{ij}^{(v)}h_{ij}^{(v)} = 0$, we get the equations for $w_{ij}^{(v)}$ and $h_{ij}^{(v)}$:

$$(2{\mathbf{W}^{(v)}}^T\mathbf{X}^{(v)} - 2{\mathbf{W}^{(v)}}^T\mathbf{W}^{(v)}\mathbf{H}^{(v)} - \alpha \sum_{w=1,w\neq v}^{V} \mathbf{H}^{(w)} - 2\beta\mathbf{H}^{(v)})h_{ij}^{(v)} = 0, \tag{5.13}$$

$$(2\mathbf{X}^{(v)}{\mathbf{H}^{(v)}}^T - 2\mathbf{W}^{(v)}\mathbf{H}^{(v)}{\mathbf{H}^{(v)}}^T)w_{ij}^{(v)} = 0. \tag{5.14}$$

These equations lead to the following updating rules:

$$h_{ij}^{(v)} \leftarrow h_{ij}^{(v)} \frac{(2{\mathbf{W}^{(v)}}^T\mathbf{X}^{(v)})_{ij}}{(2{\mathbf{W}^{(v)}}^T\mathbf{W}^{(v)}\mathbf{H}^{(v)} + \alpha \sum_{w=1,w\neq v}^{V} \mathbf{H}^{(w)} + 2\beta\mathbf{H}^{(v)})_{ij}}, \tag{5.15}$$

$$w_{di}^{(v)} \leftarrow w_{di}^{(v)} \frac{(\mathbf{X}^{(v)}{\mathbf{H}^{(v)}}^T)_{di}}{(\mathbf{W}^{(v)}\mathbf{H}^{(v)}{\mathbf{H}^{(v)}}^T)_{di}}. \tag{5.16}$$

The procedure to solve (5.8) is summarized in the Algorithm 5.1.

**Algorithm 5.1** The algorithm of DiNMF

```
Input:
  Data for $V$ views $\{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, ..., \mathbf{X}^{(V)}\}$.
  Parameter $\alpha$ and $\beta$.
  for $v = 1$ to $V$ do
    Normalizing $\mathbf{X}^{(v)}$
    Initializing $\mathbf{W}^{(v)}, \mathbf{H}^{(v)}$
  end for
  for $v = 1$ to $V$ do
    while not converging do
      Fixing $\mathbf{W}^{(v)}$, updating $\mathbf{H}^{(v)}$ by (5.15)
      Fixing $\mathbf{H}^{(v)}$, updating $\mathbf{W}^{(v)}$ by (5.16)
    end while
  end for
  Calculate the average value of all data representations of each view by $\mathbf{H}^* = \frac{\sum_{v=1}^{V} \mathbf{H}^{(v)}}{V}$.
Output: The final representation matrix $\mathbf{H}^*$.
```

### 5.2.3 Convergence of DiNMF

In this section, we prove the convergence of the updating rules (5.15) and (5.16). Algorithm 5.1 is guaranteed to converge to a local minima by the following theorem:

**Theorem 1**. The objective function (5.8) is non-increasing under the update rules (5.15) and (5.16).

To prove Theorem 1, we need to show that (5.9) for each view is non-increasing under (5.15) and (5.16). Since the second term and the third term of (5.9) are only related to $\mathbf{H}$, we have exactly the same update formula for $\mathbf{W}$ in DiNMF as in [56]. Here, we only prove (5.9) is non-increasing under (5.15). Following [56], we will apply an auxiliary function, which is defined as follows:

**Definition 1** A function $G(h, h')$ is an auxiliary function of the function $J(h)$ if $G(h, h') \geq J(h)$ and $G(h, h) = J(h)$ for any $h$, $h'$.

The auxiliary function helps because of the following lemma [56],

**Lemma 1**. If $G$ is an auxiliary function of the objective function $J$, then $J$ is non-increasing under the update rule

$$h^{t+1} = \arg\min_{h} G(h, h^t). \tag{5.17}$$

Now, we will show that the update for **H** (5.15) is exactly same as the update (5.17) with a proper auxiliary function. We rewrite (5.9) as follows:

$$\begin{aligned}
O_1 &= \|\mathbf{X}^{(v)} - \mathbf{W}^{(v)}\mathbf{H}^{(v)}\|_F^2 \\
&\quad + \alpha \sum_{w=1,w\neq v}^{V} \mathrm{DIVE}(\mathbf{H}^{(v)}, \mathbf{H}^{(w)}) + \beta\|\mathbf{H}^{(v)}\|_F^2 \\
&= \sum_{i=1}^{m^{(v)}}\sum_{j=1}^{n}(x_{ij}^{(v)} - \sum_{k=1}^{K} w_{ik}^{(v)} h_{kj}^{(v)})^2 \\
&\quad + \alpha \sum_{w=1,w\neq v}^{V} \sum_{k=1}^{K}\sum_{j=1}^{n} h_{kj}^{(v)} h_{jk}^{(w)} + \beta \sum_{k=1}^{K}\sum_{j=1}^{n} h_{kj}^{(v)} h_{jk}^{(v)}.
\end{aligned} \tag{5.18}$$

Given an element $h_{ab}^{(v)}$ in $\mathbf{H}^{(v)}$, we use $F_{ab}^{(v)}$ to denote the part of $O_1$ which is only relevant to $h_{ab}^{(v)}$. It is easy to check that

$$\begin{aligned}
F_{ab}^{'} = (\frac{\partial O_1}{\partial \mathbf{H}})_{ab} =& (-2{\mathbf{W}^{(v)}}^T\mathbf{X}^{(v)} + 2{\mathbf{W}^{(v)}}^T\mathbf{W}^{(v)}\mathbf{H}^{(v)})_{ab} \\
&+ (\alpha \sum_{w=1,w\neq v}^{V} \mathbf{H}^{(w)} + 2\beta\mathbf{H}^{(v)})_{ab},
\end{aligned} \tag{5.19}$$

$$F_{ab}^{''} = (2{\mathbf{W}^{(v)}}^T\mathbf{W}^{(v)})_{aa} + 2\beta\mathbf{I}_{bb}. \tag{5.20}$$

Since our update is essentially element wise, it is sufficient to show that each $F_{ab}$ is non-increasing under the update rule (5.15). We prove this by defining the auxiliary function regarding $h_{ab}^{(v)}$ as follows:

**Lemma 2**. The function

$$\begin{aligned}
&G(h_{ab}^{(v)}, {h_{ab}^{(v)}}^t) = F_{ab}({h_{ab}^{(v)}}^t) + F_{ab}^{'}({h_{ab}^{(v)}}^t)(h^{(v)} - {h_{ab}^{(v)}}^t) \\
&+ \frac{2({\mathbf{W}^{(v)}}^T\mathbf{W}^{(v)}\mathbf{H}^{(v)})_{ab} + \alpha\sum_{w=1,w\neq v}^{V}\mathbf{H}_{ab}^{(w)} + 2\beta\mathbf{H}_{ab}^{(v)}}{{h_{ab}^{(v)}}^t}(h^{(v)} - {h_{ab}^{(v)}}^t)^2
\end{aligned} \tag{5.21}$$

is an auxiliary function for $F_{ab}$, which is the part of $O_1$ and only relevant to $h_{ab}^{(v)}$.

**Proof**. Since $G(h^{(v)}, h^{(v)}) = F_{ab}(h^{(v)})$ is obvious, we need only show that $G(h^{(v)}, {h_{ab}^{(v)}}^t) \geq F_{ab}(h^{(v)})$. To do this, we compare the Taylor series expansion of

$F_{ab}(h^{(v)})$:

$$F_{ab}(h^{(v)}) = F_{ab}(h_{ab}^{(v)^t}) + F_{ab}^{'}(h^{(v)} - h_{ab}^{(v)^t}) + F_{ab}^{''}(h^{(v)} - h_{ab}^{(v)^t})^2. \tag{5.22}$$

Introducing (5.19) and (5.20) into (5.22) and comparing with (5.21), we can see that, instead of proving that $G(h^{(v)}, h_{ab}^{(v)^t}) \geq F_{ab}(h^{(v)})$, it is equivalent to prove

$$\frac{(\mathbf{W}^{(v)^T}\mathbf{W}^{(v)}\mathbf{H}^{(v)})_{ab} + \beta \mathbf{H}_{ab}^{(v)}}{h_{ab}^{(v)^t}} \geq (\mathbf{W}^{(v)^T}\mathbf{W}^{(v)})_{aa} + \beta \mathbf{I}_{bb}. \tag{5.23}$$

Since we have

$$\begin{aligned} (\mathbf{W}^{(v)^T}\mathbf{W}^{(v)}\mathbf{H}^{(v)})_{ab} &= \sum_{k=1}^{K} (\mathbf{W}^{(v)^T}\mathbf{W}^{(v)})_{al} h_{lb}^{(v)^t} \\ &\geq (\mathbf{W}^{(v)^T}\mathbf{W}^{(v)})_{aa} h_{ab}^{(v)^t} \end{aligned} \tag{5.24}$$

and

$$\beta \mathbf{H}_{ab}^{(v)} = \beta \sum_{j=1}^{n} h_{aj}^{(v)^t} \mathbf{I}_{jb} \geq \beta h_{ab}^{(v)^t} \mathbf{I}_{bb}, \tag{5.25}$$

(5.23) holds and $G(h^{(v)}, h_{ab}^{(v)^t}) \geq F_{ab}(h^{(v)})$.

We can now demonstrate the convergence of Theorem 1.

**Proof of Theorem 1**. Replacing $G(h^{(v)}, h_{ab}^{(v)^t})$ in (5.17) by (5.21) results in the update rule

$$\begin{aligned} h_{ab}^{(v)^{t+1}} &= h_{ab}^{(v)^t} - h_{ab}^{(v)^t} \frac{F_{ab}'(h_{ab}^{(v)^t})}{(2\mathbf{W}^{(v)^T}\mathbf{W}^{(v)}\mathbf{H}^{(v)} + \alpha \sum_{w=1, w\neq v}^{V} \mathbf{H}^{(w)} + 2\beta \mathbf{H}^{(v)})_{ab}} \\ &= h_{ab}^{(v)^t} \frac{(2\mathbf{W}^{(v)^T}\mathbf{X}^{(v)})_{ab}}{(2\mathbf{W}^{(v)^T}\mathbf{W}^{(v)}\mathbf{H}^{(v)} + \alpha \sum_{w=1, w\neq v}^{V} \mathbf{H}^{(w)} + 2\beta \mathbf{H}^{(v)})_{ab}}. \end{aligned} \tag{5.26}$$

This is exactly the same as (5.15). Since (5.21) is an auxiliary function for $F_{ab}$, $F_{ab}$ is non-increasing under (5.15) according to Lemma 1.

## 5.3 Locality Preserved DiNMF (LP-DiNMF)

Recent research has shown that data are found to lie on a nonlinear low dimensional manifold embedded in a high dimensional ambient space [2, 85, 60]. However, the standard NMF fails to discover such intrinsic geometrical structure of the data space [6]. To find a compact representation which uncovers the hidden semantics and simultaneously respects the intrinsic geometrical structure, we further extend DiNMF to LP-DiNMF so that local geometrical structure could be captured in each view.

### 5.3.1 Objective function

Cai et al. [6] imposed graph regularization on NMF. The method is based on the manifold assumption which means that, if two data points $\mathbf{x}_i$ and $\mathbf{x}_j$ are close in the original feature space, the representations of these two data points should be also close to each other. Mathematically, this can be represented by the following form: $\|\mathbf{x}_i - \mathbf{x}_j\| \to 0 \Rightarrow \|\mathbf{h}_i - \mathbf{h}_j\| \to 0$. With multi-view setting, a locality preserved term corresponding to the $v$th view is defined as:

$$\frac{1}{2}\sum_{i,j=1}^{n}(a_{ij}^{(v)}\|\mathbf{h}_i^{(v)} - \mathbf{h}_j^{(v)}\|^2)) = tr(\mathbf{H}^{(v)}\mathbf{L}^{(v)}\mathbf{H}^{(v)^T}), \tag{5.27}$$

where $\mathbf{L}^{(v)}$ is the Lagrange matrix $\mathbf{L}^{(v)} = \mathbf{D}^{(v)} - \mathbf{A}^{(v)}$, $\mathbf{A}^{(v)} = (a_{ij}^{(v)})$ is the weight matrix measuring the spatial closeness of data points and $\mathbf{D}^{(v)}$ is a diagonal matrix with $d_{ii}^{(v)} = \sum_j a_{ij}^{(v)}$. One of the most commonly used approaches to define the weight matrix $\mathbf{A}^{(v)}$ on the graph is $0 - 1$ weighting [6], since it is simple to implement and it performs well in practice. If $x_i^{(v)}$ and $x_j^{(v)}$ are one of the nearest neighbors to each other, $a_{ij}^{(v)} = 1$ otherwise $a_{ij}^{(v)} = 0$. Same as [116], we adopt this approach for it is simple to implement and performs well in practice. Combining this locality preserved regularizer with the objective function of DiNMF (5.8) gives rise to our LP-DiNMF, which minimizes the objective function as follows:

$$\begin{aligned}
&\sum_{v=1}^{V}\|\mathbf{X}^{(v)}-\mathbf{W}^{(v)}\mathbf{H}^{(v)}\|_F^2+\alpha\sum_{v\neq w}\mathrm{DIVE}(\mathbf{H}^{(v)},\mathbf{H}^{(w)})\\
&+\beta\sum_{v=1}^{V}\|\mathbf{H}^{(v)}\|_F^2+\gamma\sum_{v=1}^{V}tr(\mathbf{H}^{(v)}\mathbf{L}^{(v)}\mathbf{H}^{(v)^T})\\
&s.t.\quad 1\leq v,w\leq V,\mathbf{W}^{(v)},\mathbf{H}^{(v)},\mathbf{H}^{(w)},\alpha,\beta,\gamma\geq 0.
\end{aligned} \tag{5.28}$$

Please note that if we set $\alpha=\beta$, the objective function (5.28) becomes simpler as

$$\begin{aligned}
&\sum_{v=1}^{V}\|\mathbf{X}^{(v)}-\mathbf{W}^{(v)}\mathbf{H}^{(v)}\|_F^2+\alpha\sum_{v=1}^{V}\sum_{w=1,w\neq v}^{V}\mathrm{DIVE}(\mathbf{H}^{(v)},\mathbf{H}^{(w)})\\
&+\gamma\sum_{v=1}^{V}tr(\mathbf{H}^{(v)}\mathbf{L}^{(v)}\mathbf{H}^{(v)^T})\\
&s.t.\quad 1\leq v,w\leq V,\mathbf{W}^{(v)},\mathbf{H}^{(v)},\mathbf{H}^{(w)},\alpha,\gamma\geq 0.
\end{aligned} \tag{5.29}$$

The DIVE term in (5.29) not only works on multi-view setting, but also on the single view. In detail, given different views ($v\neq w$), DIVE enforces the diversity among them. For the single view ($v=w$), DIVE plays an important role to avoid over-fitting. This demonstrates the full compatibility of our objective function.

### 5.3.2 Optimization

Note that comparing with (5.8), the last term of (5.29) is related to $\mathbf{H}^{(v)}$ only, so we provide the optimization solution for updating $\mathbf{H}^{(v)}$ with $\mathbf{W}^{(v)}$ fixed.

Since updating $\mathbf{W}^{(v)}$ and $\mathbf{H}^{(v)}$ in each view is independent, (5.29) reduces to minimize the following formulation

$$\begin{aligned}
&\|\mathbf{X}^{(v)}-\mathbf{W}^{(v)}\mathbf{H}^{(v)}\|_F^2+\alpha\sum_{w=1,w\neq v}^{V}\mathrm{DIVE}(\mathbf{H}^{(v)},\mathbf{H}^{(w)})\\
&+\gamma tr(\mathbf{H}^{(v)}\mathbf{L}^{(v)}\mathbf{H}^{(v)^T}).
\end{aligned} \tag{5.30}$$

Let $\varphi_{ij}^{(v)}$ be the Lagrange multipliers for the constraint $h_{ij}^{(v)}\geq 0$ and $\boldsymbol{\varphi}^{(v)}=[\varphi_{ij}^{(v)}]$, the Lagrange function $L$ for each view can be written as

$$\begin{aligned} L = {} & tr(\mathbf{X}^{(v)}{\mathbf{X}^{(v)}}^T - 2\mathbf{X}^{(v)}{\mathbf{H}^{(v)}}^T{\mathbf{W}^{(v)}}^T \\ & + \mathbf{W}^{(v)}\mathbf{H}^{(v)}{\mathbf{H}^{(v)}}^T{\mathbf{W}^{(v)}}^T) + \alpha \sum_{w=1,w\neq v}^{V} tr(\mathbf{H}^{(v)}{\mathbf{H}^{(w)}}^T) \\ & + \alpha tr(\mathbf{H}^{(v)}{\mathbf{H}^{(v)}}^T) + \gamma tr(\mathbf{H}^{(v)}\mathbf{L}^{(v)}{\mathbf{H}^{(v)}}^T) + tr(\boldsymbol{\varphi}^{(v)}\mathbf{H}^{(v)}). \end{aligned} \tag{5.31}$$

Requiring that the derivative of $L$ with respect to $\mathbf{H}^{(v)}$ equals to 0 and using the Karush-Kuhn-Tucker (KKT) condition [5] $\varphi_{ij}^{(v)} h_{ij}^{(v)} = 0$, we have

$$h_{ij}^{(v)} \leftarrow h_{ij}^{(v)} \frac{(2{\mathbf{W}^{(v)}}^T\mathbf{X}^{(v)} + 2\gamma\mathbf{H}^{(v)}\mathbf{A}^{(v)})_{ij}}{(2{\mathbf{W}^{(v)}}^T\mathbf{W}^{(v)}\mathbf{H}^{(v)} + \alpha\mathbf{Q}^{(v)} + 2\gamma\mathbf{H}^{(v)}\mathbf{D}^{(v)})_{ij}}, \tag{5.32}$$

where $\mathbf{Q}^{(v)} = \sum_{w=1,w\neq v}^{V} \mathbf{H}^{(w)} + 2\mathbf{H}^{(v)}$.

The whole procedure for solving (5.29) are summarized in the Algorithm 5.2.

**Algorithm 5.2** The algorithm of LP-DiNMF

```
Input:
  Data for $V$ views $\{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, ..., \mathbf{X}^{(V)}\}$.
  Parameter $\alpha$ and $\beta$.
  Calculate weighting matrix of each view, $\mathbf{A}^{(v)}$
  Calculate diagonal matrix and Lagrange matrix of each view, $\mathbf{D}^{(v)}$ and $\mathbf{L}^{(v)}$, respectively
  for $v = 1$ to $V$ do
    Normalizing $\mathbf{X}^{(v)}$
    Initializing $\mathbf{W}^{(v)}, \mathbf{H}^{(v)}$
    while not converging do
      Fixing $\mathbf{W}^{(v)}$, updating $\mathbf{H}^{(v)}$ by (5.32)
      Fixing $\mathbf{H}^{(v)}$, updating $\mathbf{W}^{(v)}$ by (5.16)
    end while
  end for
  Calculate the average value of all data representations of each view by $\mathbf{H}^* = \frac{\sum_{v=1}^{V} \mathbf{H}^{(v)}}{V}$.
Output: The final representation matrix $\mathbf{H}^*$.
```

### 5.3.3 Convergence of LP-DiNMF

The Algorithm 2 above is guaranteed to converge to a local minima with the following theorem.

**Theorem 2**. The objective function in (5.29) is non-increasing under the update rules in (5.32) and (5.16).

Same as DiNMF, we omit the proof of (5.16) here. To prove (5.29) is non-increasing under (5.32), we first rewrite (5.30) as:

$$
\begin{aligned}
O_2 &= \|\mathbf{X}^{(v)} - \mathbf{W}^{(v)}\mathbf{H}^{(v)}\|_F^2 + \alpha \sum_{w=1, w\neq v}^{V} \mathrm{DIVE}(\mathbf{H}^{(v)}, \mathbf{H}^{(w)}) \\
&\quad + \alpha\|\mathbf{H}^{(v)}\|_F^2 + \gamma tr(\mathbf{H}^{(v)}\mathbf{L}^{(v)}{\mathbf{H}^{(v)}}^T) \\
&= \sum_{i=1}^{m^{(v)}}\sum_{j=1}^{n}(x_{ij}^{(v)} - \sum_{k=1}^{K} w_{ik}^{(v)} h_{kj}^{(v)})^2 + \alpha \sum_{w=1, w\neq v}^{V}\sum_{k=1}^{K}\sum_{j=1}^{n} h_{kj}^{(v)} h_{jk}^{(w)} \\
&\quad + \alpha \sum_{k=1}^{K}\sum_{j=1}^{n} h_{kj}^{(v)} h_{jk}^{(v)} + \gamma \sum_{k=1}^{K}\sum_{j=1}^{n}\sum_{l=1}^{n} h_{kj}^{(v)} L_{jl}^{(v)} h_{lk}^{(v)}.
\end{aligned}
\tag{5.33}
$$

It is easy to check that

$$
\begin{aligned}
F'_{ab} = (\frac{\partial O_2}{\partial \mathbf{H}})_{ab} &= (-2{\mathbf{W}^{(v)}}^T\mathbf{X}^{(v)} + 2{\mathbf{W}^{(v)}}^T\mathbf{W}^{(v)}\mathbf{H}^{(v)})_{ab} \\
&\quad + (\alpha \sum_{w=1, w\neq v}^{V} \mathbf{H}^{(w)} + 2\alpha\mathbf{H}^{(v)} + 2\gamma\mathbf{H}^{(v)}\mathbf{L}^{(v)})_{ab}
\end{aligned}
\tag{5.34}
$$

$$
F''_{ab} = (2{\mathbf{W}^{(v)}}^T\mathbf{W}^{(v)})_{aa} + 2\alpha\mathbf{I}_{bb} + 2\gamma\mathbf{L}_{bb}^{(v)}. \tag{5.35}
$$

Again, we prove each $F_{ab}$ is non-increasing under the update rule (5.32) based on an auxiliary function as following.

**Lemma 3**. Let $\mathbf{Q}_{ab}$=$\mathbf{H}_{ab}^{(w)}$+$2\mathbf{H}_{ab}^{(v)}$, the function

$$
\begin{aligned}
&G(h_{ab}^{(v)}, {h_{ab}^{(v)}}^t) = F_{ab}({h_{ab}^{(v)}}^t) + F'_{ab}({h_{ab}^{(v)}}^t)(h^{(v)} - {h_{ab}^{(v)}}^t) \\
&+ \frac{2({\mathbf{W}^{(v)}}^T\mathbf{W}^{(v)}\mathbf{H}^{(v)})_{ab} + \alpha\mathbf{Q}_{ab} + 2\gamma(\mathbf{H}^{(v)}\mathbf{D}^{(v)})_{ab}}{{h_{ab}^{(v)}}^t}(h^{(v)} - {h_{ab}^{(v)}}^t)^2
\end{aligned}
\tag{5.36}
$$

is an auxiliary function for $F_{ab}$ which is the part of $O_2$ and only relevant to $h_{ab}^{(v)}$.

**Proof**. In fact, we can see that Lemma 2 is a part of Lemma 3. Similar to the proof of Lemma 2, we incorporate (5.34) and (5.35) to the Taylor series expansion of $F_{ab}^{(h(v))}$ (5.22) and compare it with (5.36). Since Lemma 2 has been proved with (5.24) and (5.25), here we only need to show

$$\frac{2\gamma(\mathbf{H}^{(v)}\mathbf{D}^{(v)})_{ab}}{h_{ab}^{(v)^t}} \geq 2\gamma\mathbf{L}_{bb}^{(v)}. \tag{5.37}$$

Since we have

$$\begin{aligned}(\mathbf{H}^{(v)}\mathbf{D}^{(v)})_{ab} &= h_{aj}^{(v)^t}\sum_{j=1}^{n}\mathbf{D}_{jb}^{(v)} \geq h_{ab}^{(v)^t}\mathbf{D}_{bb}^{(v)} \\ &\geq h_{ab}^{(v)^t}(\mathbf{D}^{(v)} - \mathbf{W}^{(v)})_{bb} = h_{ab}^{(v)^t}\mathbf{L}_{bb}^{(v)},\end{aligned} \tag{5.38}$$

(5.36) holds and $G(h^{(v)}, h_{ab}^{(v)^t}) \geq F_{ab}(h^{(v)})$.

We can now demonstrate the convergence of Theorem 2.

**Proof of Theorem 2**. Putting $G(h^{(v)}, h_{ab}^{(v)^t})$ of (5.36) into (5.17), we get

$$\begin{aligned}h_{ab}^{(v)^{t+1}} &= h_{ab}^{(v)^t} - h_{ab}^{(v)^t}\frac{F'_{ab}(h_{ab}^{(v)^t})}{(2\mathbf{W}^{(v)^T}\mathbf{W}^{(v)}\mathbf{H}^{(v)} + \alpha\mathbf{Q} + 2\gamma\mathbf{H}^{(v)}\mathbf{D}^{(v)})_{ab}} \\ &= h_{ab}^{(v)^t}\frac{(2\mathbf{W}^{(v)^T}\mathbf{X}^{(v)} + 2\gamma\mathbf{H}^{(v)}\mathbf{A}^{(v)})_{ab}}{(2\mathbf{W}^{(v)^T}\mathbf{W}^{(v)}\mathbf{H}^{(v)} + \alpha\mathbf{Q} + 2\gamma\mathbf{H}^{(v)}\mathbf{D}^{(v)})_{ab}}.\end{aligned} \tag{5.39}$$

This is in line with (5.32). Since (5.36) is an auxiliary function for $F_{ab}$, $F_{ab}$ is non-increasing under (5.32).

## 5.4 Complexity Analysis

In DiNMF, for each data matrix $\mathbf{X}^{(v)} \in R^{m^{(v)}\times n}$, the complexity of updating $\mathbf{W}^{(v)}$ in (5.16) is $O(m^{(v)}nk)$. This is same as that of NMF [56]. The cost of updating $\mathbf{H}^{(v)}$ in (5.15) is $O(m^{(v)}nk + knV)$. Since usually $V \ll m^{(v)}$, assuming the iterative update stops after $t$ iterations, consequently, the overall computation of DiNMF is $O(\sum_{v=1}^{V}(t(m^{(v)}nk)))$. Clearly, its complexity is linear with respect to the number of data points ($n$) and it can scale well to large datasets. For LP-DiNMF, the overall cost of updating $\mathbf{W}^{(v)}$ and $\mathbf{H}^{(v)}$ is $O(\sum_{v=1}^{V}(tm^{(v)}nk + m^{(v)}n^2)$

because it requires additional $O(m^{(v)}n^2)$ to construct the nearest neighbor graph. The experimental analysis for both complexity is given in the subsection 5.5.7.

# 5.5 Experiments

In this section, we carry out extensive experiments on clustering to demonstrate the effectiveness of DiNMF and LP-DiNMF in exploiting the underlying diverse information across multiple views of data.

## 5.5.1 Description of datasets

We conduct experiments on one synthetic and several real world datasets, which are chosen from different domains, including documents, images and networks. The descriptions of these datasets are summarized in Table 5.1.

Table 5.1: Descriptions of the datasets

| **Datasets** | | **Size** | **view** | **Cluster** |
|---|---|---|---|---|
| Synthetic | | 5000 | 2 | 2 |
| Reuters | Reuters-1 | 600 | 3 | 6 |
| | Reuters-2 | 18578 | 5 | 6 |
| Digit | | 2000 | 2 | 10 |
| WebKB | Cornell | 195 | 2 | 5 |
| | Texas | 187 | 2 | 5 |
| | Washington | 230 | 2 | 5 |
| | Winsconsin | 265 | 2 | 5 |
| Caltech 101 Silhouettes | | 8641 | 2 | 101 |

- **Synthetic**: We first randomly generate basis matrices $\{\mathbf{W}^{(i)}\}_{i=1}^{2}$ of two views. The dimensions of two matrices are 250 and 800, respectively. The representation matrices $\{\mathbf{H}^{(i)}\}_{i=1}^{2} \in \mathbb{R}^{20\times5000}$ are generated with the constraint that the corresponding vectors of these two matrices are orthonormal to each other. To ensure that the two data representations not only contain respective distinct information but also share common information, we sample 30% vectors from one representation matrix by adding Gaussian noise with $\mathcal{N}(0,1)$ and keep these corresponding vectors exactly same in the second view. Thus, we have a dataset that consists of two views, i.e., $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$, where $\mathbf{X}^{(i)} = \mathbf{W}^{(i)}\mathbf{H}^{(i)}$. This dataset is constructed to demonstrate the correctness of the proposed diversity term and also for the computational speed analysis.

- **Reuters**[1]: As in [65], we randomly sample 100 documents each for 6 clusters, and choose English, French and German as three views to form a dataset. We call it **Reuters-1**. Besides, to demonstrate the performance of the proposed methods on large-scale dataset, we also use the original dataset and we call it **Reuters-2**. It contains feature characteristics of documents that are translated into five languages over 6 categories. In our experiments, we choose one language, English (EN), as the original language source and take the translated documents in the other four languages as the other four sources.

- **UCI Handwritten Digit**[2] : The dataset is composed of 2000 examples from 0 to 9 ten-digit classes. Each example is represented by two kinds of features, pixel averages in $2 \times 3$ windows and Zernike moment (Zernike moment represents properties of an image with no redundancy or overlap of information between the moments.).

- **WebKB**[3]: It is composed of web pages collected from computer science department websites of four universities: Cornell, Texas, Washington and Wisconsin. The webpages are classified into 7 categories. Here, we choose four most populous categories (course, faculty, project, student) for clustering. A webpage is made of two views: the text on it and the anchor text on the hyperlinks pointing to it.

- **Caltech 101 Silhouettes**[4]: This dataset is based on the Caltech 101 image annotations [25]. It centers and scales each polygon outline of the primary object in the Caltech 101 and render it on a $16 \times 16$ pixel image-plane. The outline is rendered as a filled, black polygon on a white background. Since this dataset contains one type of feature only, following [7], we extracted HOG [18] as the second view.

[1] http://multilingreuters.iit.nrc.ca

[2] http://archive.ics.uci.edu/ml/datasets/Multiple+Features

[3] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/

[4] https://people.cs.umass.edu/ marlin/data.shtml

### 5.5.2 Methods to compare

We compare the proposed approaches with several representative multi-view clustering methods and their variations.

• **Best Single View-NMF** (**BSV**): We run each view of datasets with NMF [56] and the best single view result is reported.

• **Best Single View-GNMF** (**BSVG**): Similar to **BSV**, we run each view of datasets with GNMF [6] and report the best single view results.

• **Feature Concatenation** (**FeatConcate**): It concatenates the features of all views and applies NMF to extract the low dimensional subspace representation.

• **ColNMF** [87]: It simultaneously factors data matrices of multiple views to different basis matrices with the shared consensus coefficient matrix.

• **MultiNMF** [65]: It searches for a compatible clustering solutions across multiple views by minimizing the differences between data representation matrices of each view and the consensus matrix.

• **MMNMF** [116]: It preserves the locally geometrical structure of the manifolds for multi-view clustering with regarding that the intrinsic manifold of the dataset is embedded in a convex hull of all the views' manifolds, and incorporates such an intrinsic manifold and an intrinsic coefficient matrix with a multi-manifold regularizer.

• **RMKMC** [8]: This multi-view $k$-means approach integrates heterogeneous features of data and utilizes the common cluster indicator to do clustering across multiple views. $l_{2,1}$-norm is employed to improve the robustness.

• **CoRegSPC** [53]: This pairwise multi-view spectral clustering method coregularizes the clustering hypotheses to enforce corresponding data points in each view to have the same cluster membership.

• **RMSC** [107]: This is a multi-view spectral clustering method based on low rank and sparse decomposition of the transition matrix.

• **NdNMF**: It conducts each view independently using standard NMF [55], and then applies $k$-means on the combination of new representations of each view.

Table 5.2: Comparison of clustering results ((mean ± standard deviation)%) on the datasets

| Metrics | Methods | Synthetic | Reuters-1 | Reuters-2 | Digit | WebKB | | | | Caltech101 Silhouettes |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | Cornell | Texas | Washington | Winsconsin | |
| AC | NMF | 74.45±0.24 | 45.58±2.66 | 45.29±1.57 | 64.30±3.55 | 42.21±2.19 | 53.90±2.14 | 55.30±2.02 | 48.00±1.89 | 24.61±0.64 |
| | BSVG | 74.11±0.10 | 46.23±1.88 | 45.19±0.96 | 61.15±2.63 | 43.15±1.23 | 54.62±1.95 | 55.74±0.97 | 49.75±0.75 | 25.03±0.35 |
| | FeatConcate | 78.30±0.12 | 49.37±2.61 | 43.27±2.41 | 48.66±1.27 | 40.64±1.61 | 53.32±2.76 | 54.87±2.82 | 45.28±1.68 | 26.06±2.28 |
| | ColNMF | 78.33±0.08 | 41.53±1.68 | 48.64±1.47 | 66.90±3.72 | 40.26±3.79 | 47.60±2.71 | 51.17±0.76 | 41.30±1.49 | 25.29±0.58 |
| | MultiNMF | 71.08±1.22 | 51.45±1.55 | 44.37±0.91 | 61.70±1.80 | 44.51±2.55 | 55.29±2.84 | 50.17±1.50 | 47.40±0.93 | 25.89±0.97 |
| | MMNMF | 88.00±0.17 | 52.23±2.01 | 43.76±0.47 | 70.65±3.22 | 46.77±1.22 | 55.87±2.69 | 50.09±1.36 | 49.87±1.12 | 26.66±0.89 |
| | RMKMC | 90.36±0.10 | 39.48±3.08 | 40.28±1.19 | 58.97±1.06 | 39.99±0.66 | 54.48±0.92 | 59.18±0.64 | 54.21±0.87 | 25.31±0.51 |
| | CoRegSPC | 79.06±1.34 | 54.13±1.66 | N/A | 58.41±3.89 | 42.26±0.50 | 57.54±0.52 | 59.74±0.42 | 48.91±0.26 | 24.64±0.68 |
| | RMSC | 70.26±1.14 | 48.46±2.32 | N/A | 67.50±0.00 | 38.56±0.96 | 42.67±0.61 | 41.39±0.91 | 36.26±1.61 | 26.51±0.41 |
| | NdNMF | 76.20±1.45 | 45.38±3.25 | 43.75±0.51 | 62.16±0.63 | 47.90±1.10 | 57.54±0.84 | 54.74±1.04 | 56.30±0.91 | 25.68±0.21 |
| | DiNMF | *98.88±0.46* | *54.17±2.44* | *50.43±1.27* | *72.35±2.05* | *54.46±1.75* | *57.77±1.07* | *59.30±1.15* | *62.03±2.62* | *28.60±0.31* |
| | LP-DiNMF | **99.96±0.48** | **55.08±2.12** | **51.00±1.83** | **95.20±1.73** | **60.10±2.92** | **59.25**±3.09 | **65.39±1.74** | **67.02±1.90** | **29.13±0.26** |
| NMI | BSV | 60.95±0.14 | 33.75±2.00 | 29.04±1.14 | 59.84±0.98 | 13.59±1.82 | 24.32±1.95 | 28.59±1.34 | 25.10±1.94 | 50.72±0.31 |
| | BSVG | 53.16±0.47 | 34.06±2.65 | 28.44±0.53 | 60.34±2.39 | 21.73±2.45 | 24.46±1.73 | 29.27±1.70 | 25.49±1.27 | 51.71±0.08 |
| | FeatConcate | 52.51±0.00 | 31.39±2.25 | 28.25±2.91 | 43.30±2.82 | 16.57±1.52 | 22.60±2.23 | 30.49±1.28 | 21.62±1.85 | 53.03±0.41 |
| | ColNMF | 54.65±0.68 | 31.55±2.45 | 32.09±2.21 | 62.95±1.96 | 17.42±1.73 | 20.56±2.69 | 20.36±2.56 | 30.17±1.24 | 52.44±0.82 |
| | MultiNMF | 53.29±0.45 | 36.38±0.81 | 29.02±1.39 | 55.44±0.79 | 12.41±0.36 | 20.28±1.28 | 21.57±0.75 | 13.58±0.90 | 50.00±0.38 |
| | MMNMF | 67.83±0.79 | 38.71±3.01 | 29.24±1.23 | *73.95±1.83* | 26.38±2.90 | 28.15±2.83 | 33.52±1.35 | 29.98±2.82 | 50.16±0.37 |
| | RMKMC | 55.72±0.68 | 22.03±3.47 | 22.18±0.14 | 58.79±3.41 | 11.79±2.97 | 17.24±2.34 | 30.27±1.17 | 28.42±0.77 | 52.33±0.26 |
| | CoRegSPC | 52.34±0.94 | 35.77±1.77 | N/A | 50.68±2.69 | 14.34±0.00 | 21.42±0.28 | 23.54±0.09 | 15.32±0.00 | 52.31±0.35 |
| | RMSC | 41.89±0.55 | 29.75±1.61 | N/A | 62.00±0.00 | 14.52 ±1.15 | 17.95 ±0.89 | 18.85±0.95 | 14.48±0.24 | 53.83±0.32 |
| | NdNMF | 44.08±0.72 | 21.95±4.09 | 22.95±1.61 | 56.65±2.83 | 22.81±1.57 | 20.66±3.14 | 29.75±3.02 | 33.18±2.71 | 52.55±0.84 |
| | DiNMF | *80.23±0.79* | *21.50±2.31* | *29.27±1.50* | *66.60±1.63* | *33.31±1.78* | *37.84±2.51* | *38.20±1.43* | *43.00±1.79* | *54.60±0.02* |
| | LP-DiNMF | **83.12**±0.46 | **38.52±1.49** | **32.14±1.22** | **90.45±1.27** | **32.65±3.04** | **40.57±2.51** | **42.39±1.39** | **43.03±2.80** | **55.41±0.34** |
| Purity | BSV | 79.45±0.24 | 47.41±2.41 | 48.35±1.01 | 67.63±2.97 | 45.77±1.44 | 64.55±1.09 | 67.20±1.13 | 51.70±1.96 | 43.58±0.22 |
| | BSVG | 80.11±0.10 | 48.48±2.32 | 51.73±0.70 | 69.19±3.83 | 53.41±2.51 | 66.88±0.70 | 67.87 ±1.40 | 62.55±1.43 | 44.37±0.13 |
| | FeatConcate | 79.30±0.15 | 50.20±2.80 | 48.33±3.53 | 51.55±2.73 | 49.54±2.81 | 64.18±1.19 | 66.35±2.62 | 73.21±2.91 | 45.89±0.31 |
| | ColNMF | 78.33±0.05 | 44.14±2.14 | 44.57±0.46 | 71.11±1.07 | 49.44±1.16 | 62.74±1.13 | 68.43±1.44 | 61.00±1.25 | 45.26±0.35 |
| | MultiNMF | 70.08±1.22 | 53.60±1.31 | 54.50±1.14 | 63.21±1.22 | 45.18±1.09 | 64.01±1.07 | 67.83±1.28 | 59.55±1.21 | 44.89±0.48 |
| | MMNMF | 88.00±0.17 | 53.70±2.39 | 53.43±0.75 | 74.15±2.73 | 57.95±2.80 | 67.43±3.04 | 70.43±0.98 | 67.66±2.09 | 41.94±0.29 |
| | RMKMC | 90.16±0.62 | 40.63±3.28 | 40.22±0.32 | 63.87±3.43 | 46.59±2.70 | 60.99±1.66 | 68.91±1.45 | 65.81±1.64 | 45.15±0.24 |
| | CoRegSPC | 77.54±1.32 | **55.47±1.70** | N/A | 58.66±1.43 | 44.97±0.25 | 64.54±0.52 | 65.48±0.22 | 52.72±0.36 | 45.35±0.58 |
| | RMSC | 69.36±0.37 | 49.98±2.25 | N/A | 68.60±0.78 | 49.28±1.65 | 60.67±2.61 | 63.26±1.11 | 56.23±0.53 | 47.30±0.41 |
| | NdNMF | 76.06±0.45 | 46.57±3.35 | 47.25±1.57 | 64.23±2.02 | 57.36±2.01 | 63.03±1.66 | 67.57±2.78 | 68.40±2.91 | 44.89±0.23 |
| | DiNMF | *98.36±0.52* | 54.65±2.04 | *54.55±1.27* | *74.45±0.49* | *63.82±2.37* | *69.51±1.39* | *70.96±2.03* | *73.48±2.11* | *47.69±0.52* |
| | LP-DiNMF | **98.96±0.42** | *55.15±1.51* | **56.39±1.79** | **95.20±1.27** | **67.13±2.29** | **72.62±3.39** | **72.78±1.01** | **74.19±2.18** | **48.37±0.30** |

### 5.5.3 Experimental setup

For each compared method, we set the parameters according to original papers where the approaches were first proposed. As BSVG, MMNMF and LP-DiNMF require construction of the nearest neighbor graph, we set the number of nearest neighbor equal to the number of classes of the data $k$, as suggested in [116]. For DiNMF and LP-DiNMF, we normalize the data first and then initialize both $\mathbf{W}^{(v)}$ and $\mathbf{H}^{(v)}$ for each view in the range [0,1]. Similar to [97, 42], the regularization parameters ($\alpha$, $\beta$ in (5.8) and $\alpha$, $\gamma$ in (5.29)) are chosen from $\{0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000\}$. To avoid randomness, we run each method 10 times with different initializations and report the average results and their standard deviations. The clustering results are evaluated by three widely adopted evaluation metrics, including accuracy (AC) [64], normalized mutual information (NMI) [64] and Purity [21](More details of the metrics are already given in the previous chapter.). Each metric favors different properties in clustering, and hence we report results on these measures to perform a more comprehensive evaluation. For all these metrics, the higher value indicates better clustering quality.

### 5.5.4 Clustering results

Table 5.2 demonstrates the average results and standard deviations for each method on the datasets. Note that, the results of CoRegSPC and RMSC on Reuters-2 are not available (N/A) since they demand huge memory. In each row of the table, the best result is highlighted in **boldface** and the second best result in *italic*. It is clear to see that both DiNMF and LP-DiNMF consistently outperform the other methods, sometimes even very significantly, which demonstrates the advantage of our approaches in terms of clustering performance. Compared with NdNMF, DiNMF improves performances more than 5% on all datasets in terms of AC, NMI and Purity, which proves the effectiveness of the proposed diversity constraints. We also notice that directly concatenating all the features (i.e., FeatConcate) is not an ideal approach since it always performs worse than the best single view (BSV). Moreover, LP-DiNMF performs better than DiNMF on all the datasets. This indicates that exploiting the geometric structures in data spaces indeed can improve the cluster performance, also verifies the manifold assumption and confirms the correctness of our approaches.

### 5.5.5 Analysis of redundancy rate

To verify that DIVE reduces the redundancy information among multiple representations, we propose a redundancy rate (RED) metric as follows:

$$\begin{aligned} \mathrm{RED}(\mathbf{H}^{(1)}, ..., \mathbf{H}^{(V)}) &= \frac{\sum_{i=1}^{n} \sum_{v=1, v\neq w}^{V} cos^2(\mathbf{h}_i^{(v)}, \mathbf{h}_i^{(w)})}{V(V-1)n}. \\ s.t. \quad cos^2(\mathbf{h}_i^{(v)}, \mathbf{h}_i^{(w)}) &= \frac{\mathbf{h}_i^{(v)} \odot \mathbf{h}_i^{(w)}}{|\mathbf{h}_i^{(v)}| \odot |\mathbf{h}_i^{(w)}|} \end{aligned} \tag{5.40}$$

It assesses the average sum of similarity of all $n$ data vectors in all pairs of views and ranges from 0 to 1, where 0 means a completely complementary result, and 1 vice versa.

We compare the redundancy rate of the proposed approaches against MultiNMF, MMNMF and NdNMF, which are all under the framework of NMF and then take the same approach to obtain the final multi-view representation matrix $\mathbf{H}^*(=\frac{\sum_{v=1}^{V}\mathbf{H}^{(v)}}{V})$. The results of comparison are reported in Table 5.3. It can be seen that MultiNMF always gets the highest rate followed by MMNMF and NdNMF, while it is less than 20% for DiNMF in all cases. This demonstrates the effectiveness of the proposed DIVE that enforces the complementarity across multiple views. However, LP-DiNMF does not always achieve stable and low redundancy rate. For example, it gets the lowest redundancy rate in Texas with 0.1222 compared with other approaches, but a higher rate (0.1852) than DiNMF in Winsconsin. This is because the representations of multiple views in LP-DiNMF are co-regularized by both the manifold structure and the diversity

Table 5.3: Comparison of redundancy rate

| Methods | Synthetic | Reuters-1 | Digit | Cornell | Texas | Washington | Winsconsin |
|---|---|---|---|---|---|---|---|
| MultiNMF | 0.9986 | 0.9970 | 0.5826 | 0.8503 | 0.8472 | 0.8229 | 0.8521 |
| MMNMF | 0.5998 | 0.4800 | 0.4437 | 0.3440 | 0.4318 | 0.3598 | 0.3698 |
| NdNMF | 0.4637 | 0.2658 | 0. 2755 | 0.2395 | 0.2077 | 0.2683 | 0.1122 |
| DiNMF | **0.1838** | **0.1087** | **0.1931** | **0.0651** | 0.1873 | **0.0609** | **0.0783** |
| LP-DiNMF | 0.3509 | 0.1266 | 0.2663 | 0.0894 | **0.1222** | 0.1013 | 0.1852 |

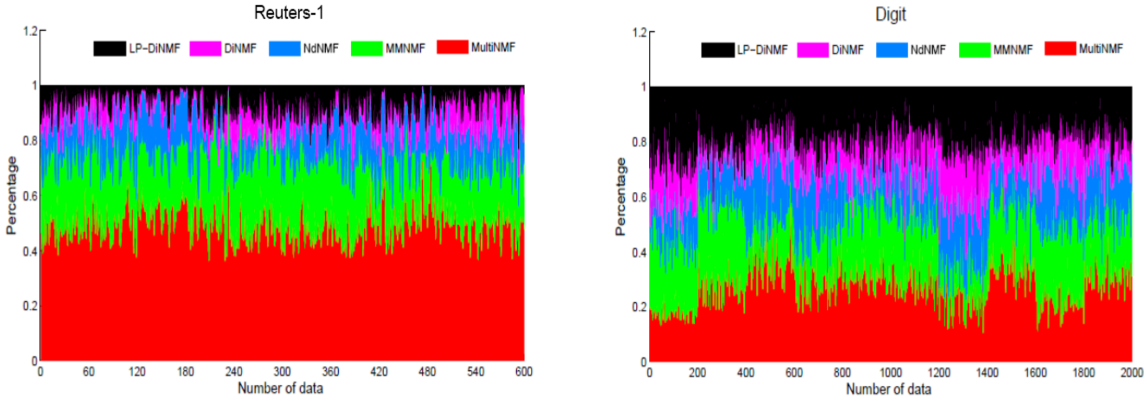


Figure 5.2: Comparison of redundancy rate on Reuters-1 and Digit dataset

term. There is a tradeoff between the two regularization terms. Thus, different from DiNMF which is only regularized by the diversity term, LP-DiNMF is less likely to get the lowest rate.

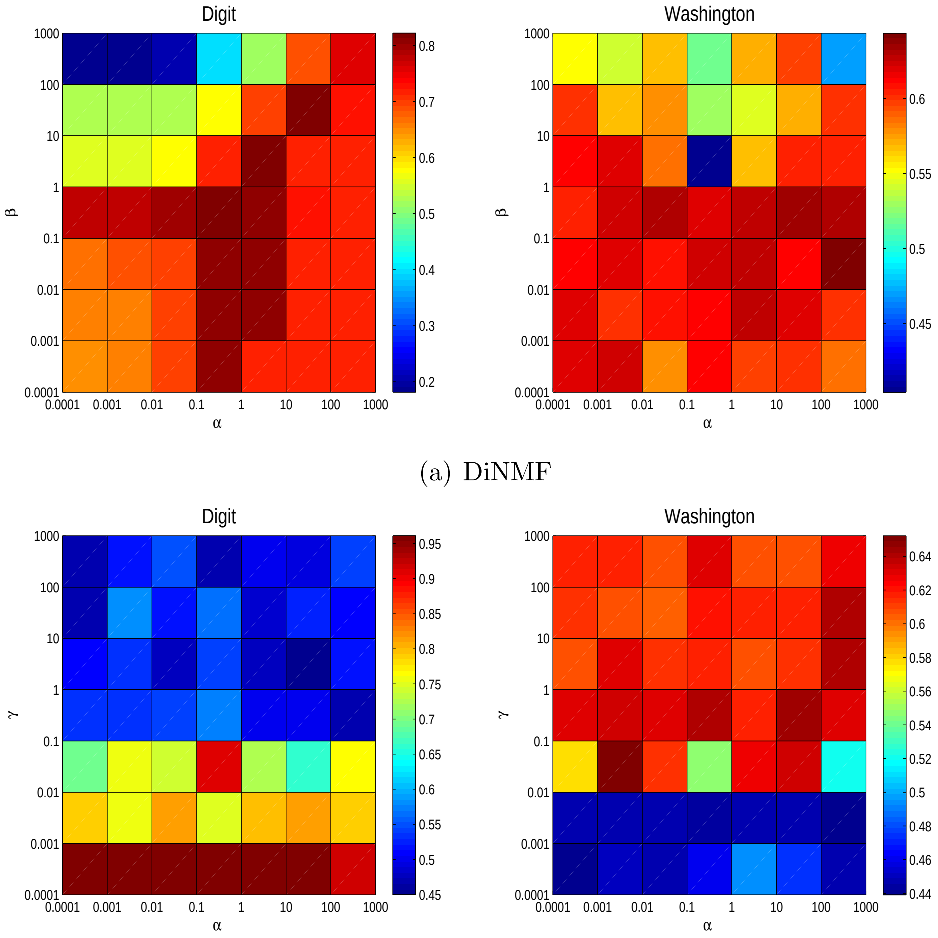
To have a visual perception of redundancy, we take the Digit (2 views) and Reuters-1 (3 views) as examples and demonstrate the redundancy rate of each data vector in details, as shown in Figure 5.2. The horizontal axis represents the number of data points and the vertical axis means the scaled redundancy rate. For each approach, the scaled redundancy rate is the percentage of its true redundancy rates over that of all five approaches. Each method is represented by one color. The wider area a color occupies, the more redundant information an approach has. Figure 5.2 shows that DiNMF (marked in purple) occupies the narrowest area, while MultiNMF occupies the widest area in both Digit and Reuters datasets.

The results of Figure 5.2 is inline with Table 5.3, which proves that DiNMF effectively exploits the diverse information across multiple views.

### 5.5.6 Parameter study

We tested the effect of the parameters $\alpha$ and $\beta$ of DiNMF, as well as $\alpha$ and $\gamma$ of LP-DiNMF. In DiNMF $\alpha$ and $\beta$ affect the diversity and smoothness, while in LP-DiNMF, $\alpha$ and $\gamma$ adjust the effects of the diversity and graph regularization term. For both methods, we picked the value of each parameter from

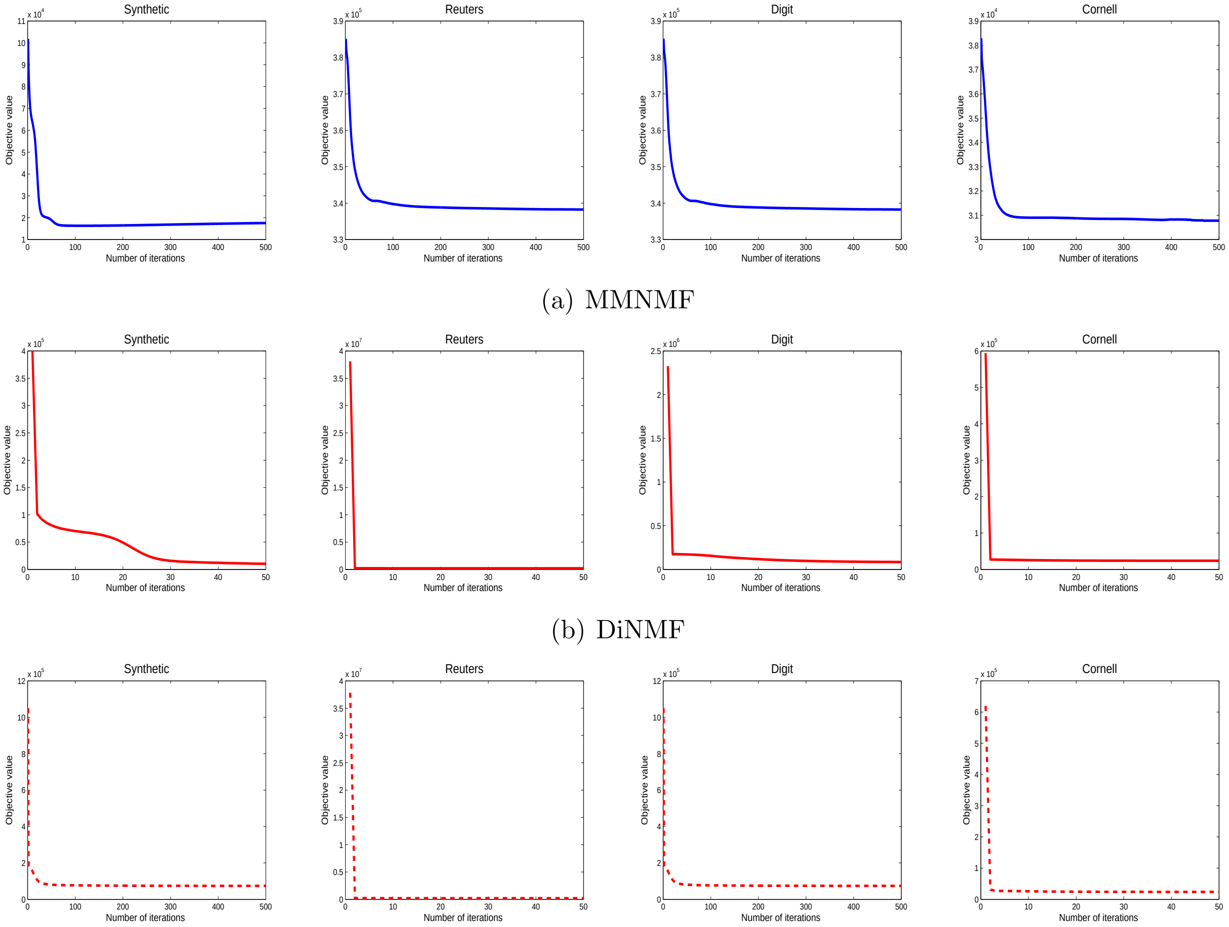
$\{0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000\}$. Taking the Digit and Washington as examples, we can find that DiNMF in Figure 5.3(a) achieves more than 70% accuracy in Digit and 60% in Washington for $\alpha$ and $\beta$ in most cases, demonstrating that the performance of DiNMF is relatively robust to parameter tuning. Figure 5.3(b) shows that LP-DiNMF is relatively stable with varying $\alpha$, but significantly affected by $\gamma$. This further verified the importance of preserving manifold structure.



(a) DiNMF

(b) LP-DiNMF

Figure 5.3: The effect of parameter $\alpha$ and $\beta$ in DiNMF and $\alpha$ and $\gamma$ in LP-DiNMF. Different colors means different accuracies and the color close to red indicates high accuracy.



(a) MMNMF

(b) DiNMF

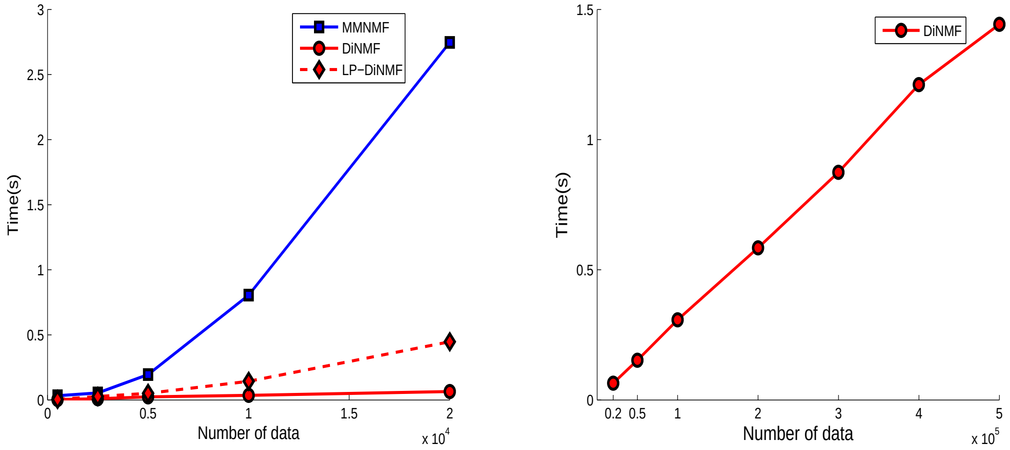
(c) LP-DiNMF

Figure 5.4: Comparison of convergence speed (Note that different scales of axes are used for clearer illustration)

### 5.5.7 Study of computational speed

We have proven the convergence of our update rules and analyzed the computational complexity of DiNMF and LP-DiNMF against MMNMF in previous sections. Here our experiments demonstrate their convergence curves in Figure 5.4 and computational time in Figure 5.5. All our experiments are conducted on a PC with two octa-core Intel Xeon CPU processors at 2.5 GHz and 256G bytes memory.

Because the results of different networks datasets (Cornell, Texas, Washington and Winsconsin) have similar convergency, here we just took one network (Cornell) as an example. Figure 5.4 shows the convergence curve of the three methods



(a) (b)

Figure 5.5: Running time of DiNMF v.s. MMNMF on Synthetic dataset.

on Synthetic, Reuters, Digit and Cornell. For each figure, the horizontal axis is the number of iterations and the vertical axis is the value of objective function. We can see that MMNMF (Figure 5.4(a)) needs around 100 iterations for each dataset, while DiNMF (Figure 5.4(b)) is the most efficient, since the objective function values are non-increasing and drop sharply within a small number of iterations (10 iterations) in all cases. Although LP-DiNMF (Figure 5.4(c)) requires nearly 100 iterations for the Synthetic and Digit database, its objective values drop faster than that of MMNMF. This empirically proves our convergence theory.

As discussed in section 5.4, DiNMF has linear time complexity with the number of data points. Here, we verify this claim on the Synthetic dataset. Figure 5.5 reports the average running time of each iteration of three methods on the Synthetic dataset. The default setting is 5000 data points, 2 clusters, and 2 views. During the experiment, we fixed the number of clusters and views but changed the number of data. Figure 5.5 (a) shows the running time of three methods in terms of varying data points within $\{0.05, 0.25, 0.5, 1, 1.5, 2\} \times 10^4$. Clearly, DiNMF is linear in execution time, and MMNMF costs significantly more time than DiNMF and LP-DiNMF. To better demonstrate DiNMF's linearity and good scalability to large datasets, we increased the amount of data to a large scale, i.e., $\{0.2, 0.5, 1, 2, 3, 4, 5\} \times 10^5$ and reported corresponding running time each in

Figure 5.5 (b). Clearly, the results are in line with the analysis in subsection 5.4.

## 5.6 Conclusion

In this chapter, we have advanced the frontier of NMF by proposing a novel idea that explores diverse information among multi-view representations. To achieve this, we have proposed a Diverse Nonnegative Matrix Factorization (DiNMF) approach for more comprehensive and accurate multi-view learning. With a novel diversity regularization term, DiNMF explicitly enforces the orthogonality of different data representations. Importantly, DiNMF converges linearly and scales well with large-scale data. Taking a step further, we have extended DiNMF by incorporating manifold information and proposed Locality Preserved DiNMF (LP-DiNMF) method. Extensive experiments conducted on both synthetic and benchmark datasets have demonstrated promising results of our methods, which conform to our theoretical analysis. DiNMF is inapplicable to many real-world problems where limited knowledge from domain experts is available such as label information. In reality, the cost associated with the labeling process may render a fully labeled training set infeasible, whereas acquisition of a small set of labeled data is relatively inexpensive. The labeled data, which if utilized, could be benefit for more accurate learning. In the next chapter, we tend to extend multi-view NMF to a semi-supervised setting by taking integration of multi-view information and label information into consideration.

# Chapter 6

# Adaptive Multi-View Nonnegative Matrix Factorization

## 6.1 Introduction

In reality, often some supervised information, e.g., labels of data or the pairwise information (such as must-link constraints) between data, are often available. Such information indicate whether the data must be or cannot be in the same cluster, therefore stronger discriminant information can be delivered for clustering performances. Although, semi-supervised learning approaches [98, 29, 68, 114, 64] have received a great attention recently, few have utilized semi-supervised multi-view learning methods. For multi-view learning, the supervised information usually has consistency across multiple views. If we can guarantee data with same label but come from various views are still grouped into the same cluster, this will improve the clustering accuracy [2], [119]. Therefore, how to utilize this discriminative information for guiding the multi-view learning is of great value. Besides, existing approaches are difficult to determine the weight of each view and treat them equally. This oversimplified way are not always satisfied in the real-world application, since each view may have different contributions. Taking faces and cars as example, although they can be represented by a combination of multiple viewpoints, some of the views are more informative hence are better representations than others. For instance, a frontal or a three-quarter view is a better representation for faces than a profile view [3], [63]. Similarly, the side or frontal view of a car is more informative than the top view of it. Finally, outliers or noisy data are ubiquitous, and thus, a robust multi-view learning approach is required for practical applications.

To address these challenges altogether, we propose a new multi-view NMF approaches, called Adaptive Multi-View Semi-Supervised Nonnegative Matrix Factorization (AMVNMF), which not only considers the consistency of multi-view and supervised information, but also can adjust the weight of each view automatically. The overall advantages of this approach are as follows:

1. By taking the label information as hard constraints, AMVNMF guarantees that data sharing the same label will have the same new representation and be mapped into the same class in the low-dimensional space regardless whether they come from the same view.

2. To our best knowledge, this is the first attempt to introduce a single parameter to control the distribution of weighting factors for NMF-based multi-view clustering. Consequently, the weight factor of each view can be assigned automatically depending on the dissimilarity between each new representation matrix and the consensus matrix.

3. Using the structured sparsity-inducing, $L_{2,1}$-norm, AMVNMF is robust against noises and hence can achieve more stable clustering results.

## 6.2 Review of Constrained NMF

Given $n$ data $\mathbf{X} = [x_1, x_2, ..., x_n] \in \mathbb{R}^{m \times n}$, traditional NMF is to measure the dissimilarity between $\mathbf{X}$ and $\mathbf{W}\mathbf{H}^T$ as

$$\|\mathbf{X} - \mathbf{W}\mathbf{H}^T\|_F^2, \tag{6.1}$$

where $\mathbf{W} \in \mathbb{R}^{m \times k}$ and $\mathbf{H} \in \mathbb{R}^{n \times k}$.

NMF is an unsupervised learning algorithm. That is, NMF is inapplicable to many real-world problems where limited knowledge from domain experts is available. However, some supervision information such as labels or instance-level constraints including must-link constraints and cannot-link constraints [95, 96] is often available, which can be a valuable guidance for finding a more discriminative representation. To this end, Liu et al. [64] proposed a constraint NMF (CNMF) to extend the traditional unsupervised NMF to a semi-supervised learning approach. It builds a label constraint matrix which incorporates the label information as hard constraints so that the data sharing the same label have the same new representation. In particular, assuming the first $l$ data points are labeled with $c$ classes, then an indicator matrix $\mathbf{C}$ can be constructed, where $c_{i,j} = 1$ if $v_i$ is labeled with $j$th class; or $c_{i,j} = 0$ otherwise. Then, the label constraint matrix $\mathbf{A}$ can be defined as follows,

$$\mathbf{A} = \begin{pmatrix} \mathbf{C}_{l\times c} & 0 \\ 0 & \mathbf{I}_{n-l} \end{pmatrix}, \tag{6.2}$$

where $\mathbf{I}_{n-l}$ is a $(n-l) \times (n-l)$ identity matrix. For example, consider $n$ data points, among which $x_1$, $x_2$ are labeled with class I, $x_3$, $x_4$ are labeled with class II, $x_5$ is labeled with class III, and the other $n-5$ data points are unlabeled. The label constraint matrix $\mathbf{A}$ based on this example can be represented as follows:

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \mathbf{I}_{n-5} \end{pmatrix}. \tag{6.3}$$

Recall that NMF maps each data point $x_i$ to its new representation $h^i$ from $P$-dimensional space to $k$-dimensional space, where $h^i$ represents the $i$th row of $\mathbf{H}$. To incorporate label information, we introduce an auxiliary matrix $\mathbf{Z}$ with $\mathbf{H} = \mathbf{AZ}$. As we can see from $\mathbf{A}$, if $x_i$ and $x_j$ have the same label, then the $i$th row and $j$th row of $\mathbf{A}$ must be the same, and so $h^i$=$h^j$, which guarantees that data sharing the same label have the same new representation. Therefore, the objective function can be written as follows,

$$\min_{\mathbf{W}\geq 0, \mathbf{Z}\geq 0} \|\mathbf{X} - \mathbf{W}\mathbf{Z}^T\mathbf{A}^T\|_F^2. \tag{6.4}$$

## 6.3 Aaptive Multi-View NMF(AMVNMF)

The framework of AMVNMF to enhance multi-view learning by incorporating label information as constraints is shown in the Figure 6.1. Given a dataset with extracted $V$ features, represented by $\{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, ..., \mathbf{X}^{(V)}\}$ and a label matrix $\mathbf{A}$, AMVNMF aims to obtain a common consensus representation $\mathbf{Z}^*$ that contains information from different views with utilizing $\mathbf{A}$ to enforce the data points that are of the same label to have the same representation. The weight of each view $\alpha^{(v)}$ is learnt adaptively according to the differences between each corresponding representation $\mathbf{Z}^{(v)}$ and the consensus $\mathbf{Z}^*$. In the following, we will introduce how we construct the objective function of AMVNMF.

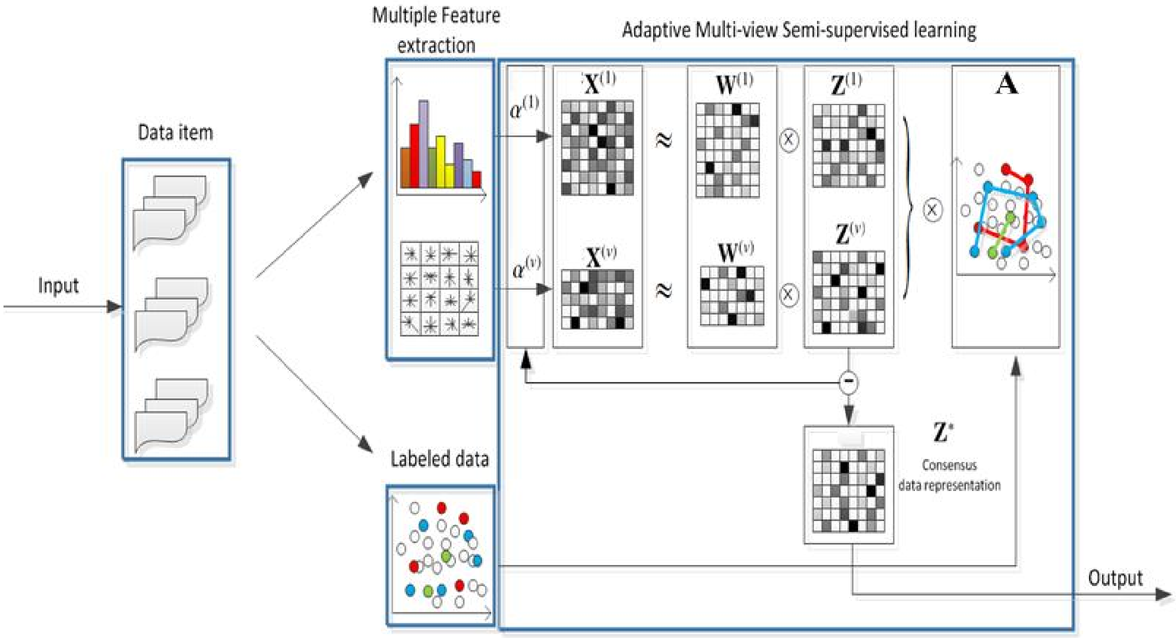


Figure 6.1: The framework of AMVNMF.

### 6.3.1 Objective function

As mentioned earlier, CNMF is a semi-supervised learning method which can be only used under single-view situation. In order to integrate all the $V$ available views, for each view $\mathbf{X}^{(v)}$, CNMF is extended straightforwardly as

$$\min_{\mathbf{W}\geq 0,\mathbf{Z}\geq 0} \sum_{v=1}^{V} \|\mathbf{X}^{(v)} - \mathbf{W}^{(v)}\mathbf{Z}^{(v)T}\mathbf{A}^{(v)T}\|_F^2, \tag{6.5}$$

where $\mathbf{W} \in \mathbb{R}^{d_v \times K}$ and $\mathbf{Z} \in \mathbb{R}^{K \times N}$. Since the matrix $\mathbf{A}$ above is constructed only based on the label information and consistent for different features, which means different features share the same constraint matrix $\mathbf{A}$. Thus, given $V$ types of heterogeneous features, $v = 1, 2, ...V$, we naturally integrate all these view together and propose the objective function as follows,

$$\min_{\mathbf{W}\geq 0,\mathbf{Z}\geq 0} \sum_{v=1}^{V} \|\mathbf{X}^{(v)} - \mathbf{W}^{(v)}(\mathbf{Z}^{(v)})^T\mathbf{A}^T\|_F^2. \tag{6.6}$$

Assuming that, new representation matrices of $V$ views are regularized towards a common consensus matrix $\mathbf{H}^*$, we aim to obtain $\mathbf{H}^*$, which uncovers the common latent structure shared by multiple views. With the constraint matrix $\mathbf{A}$ and a consensus auxiliary matrix $\mathbf{Z}^*$, we have $\mathbf{H}^* = \mathbf{A}\mathbf{Z}^*$. Since $\mathbf{A}$ is known, we turn the problem of finding $\mathbf{H}^*$ into the problem of finding $\mathbf{Z}^*$. The objective function can be rewritten as follows,

$$\begin{aligned}\min_{\mathbf{W}^{(v)},\mathbf{Z}^{(v)},\mathbf{Z}^*\geq 0} &\sum_{v=1}^{V} \|\mathbf{X}^{(v)} - \mathbf{W}^{(v)}(\mathbf{Z}^{(v)})^T\mathbf{A}^T\|_F^2 \\ &+\sum_{v=1}^{V} \lambda_v \|\mathbf{Z}^{(v)} - \mathbf{Z}^*\|_F^2,\end{aligned} \tag{6.7}$$

where $\lambda_v$ is the weight factor for $v$th view.

Note that different views may not be comparable at the same scale. Thus, without loss of generality, we assume $\|\mathbf{X}^{(v)}\|_1 = 1$. Also, in order to make different $\mathbf{Z}^{(v)}$ comparable and meaningful, we constrain $\|\mathbf{W}^{(v)}\|_1 = 1$. To do so, we introduce

$$\mathbf{Q}^{(v)} = Diag(\sum_{i=1}^{V} \mathbf{W}_{i,1}^{(v)}, \sum_{i=2}^{V} \mathbf{W}_{i,2}^{(v)}, ..., \sum_{i=1}^{V} \mathbf{W}_{i,k}^{(v)}), \tag{6.8}$$

to normalize $\mathbf{W}^{(v)}$ by using $\mathbf{W}^{(v)} = \mathbf{W}^{(v)}\mathbf{Q}^{(v)^{-1}}$. In this way, we can approximately constrain $\|(\mathbf{Z}^{(v)})^T\mathbf{A}^T\|_1 = 1$ so that $\mathbf{Z}^{(v)}$ is within the same range [65].

Due to $\mathbf{W}^{(v)}\mathbf{Z}^{(v)T}\mathbf{A}^T = \mathbf{W}^{(v)}{\mathbf{Q}^{(v)}}^{-1}(\mathbf{Z}^{(v)}\mathbf{Q})^{(v)^T}\mathbf{A}^T$, (6.7) could then be written as

$$\begin{aligned} \min_{\mathbf{W}^{(v)},\mathbf{Z}^{(v)},\mathbf{Z}^*\geq 0} \sum_{v=1}^{V} \|\mathbf{X}^{(v)} - \mathbf{W}^{(v)}(\mathbf{Z}^{(v)})^T\mathbf{A}^T\|_F^2 \\ + \sum_{v=1}^{V} \lambda_v \|\mathbf{Z}^{(v)}\mathbf{Q}^{(v)} - \mathbf{Z}^*\|_F^2. \end{aligned} \tag{6.9}$$

Normally, for all $V$ views, one needs to specify each parameter $\lambda_v$ which reflects each view's importance. Apparently, without any prior knowledge, it is hard to decide which view will contribute more or less. In order to reduce the number of these parameters, and also learn the weights of each view adaptively, we propose the following formula:

$$\begin{aligned} J = \min_{\mathbf{W}^{(v)},\mathbf{Z}^{(v)},\mathbf{Z}^*,\alpha^{(v)}\geq 0} \sum_{v=1}^{V} \|\mathbf{X}^{(v)} - \mathbf{W}^{(v)}(\mathbf{Z}^{(v)})^T\mathbf{A}^T\|_F^2 \\ + \sum_{v=1}^{V} (\alpha^{(v)})^\gamma \|\mathbf{Z}^{(v)}\mathbf{Q}^{(v)} - \mathbf{Z}^*\|_F^2 \\ s.t. \sum_{v=1}^{V} \alpha^{(v)} = 1. \end{aligned} \tag{6.10}$$

We can see that instead of setting $V$ fixed values separately, we use a single parameter $\gamma$ to control the distribution of weight factors, such as the important views will get bigger weights adaptively during the multi-view clustering.

However, the objective function uses $F$-norm to measure the approximation errors is unstable and sensitive to outliers, because large errors are squared so can easily dominate the objective function. We incorporate $L_{2,1}$-norm loss function to alleviate noises and outliers effectively, which is defined as

$$\|\mathbf{G}\|_{2,1} = \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{V} \mathbf{G}_{ji}^2} = \sum_{i=1}^{n} \|g_i\|, \tag{6.11}$$

where $g_i$ is the $i$th column of $\mathbf{G}$. Thus, for traditional NMF, the robust formulation of the error function can be written as

$$\begin{aligned} \|\mathbf{X}-\mathbf{W}\mathbf{H}^T\|_{2,1} &= \sum_{i=1}^{n}\sqrt{\sum_{j=1}^{V}(\mathbf{X}-\mathbf{W}\mathbf{H}^T)_{ji}^2} \\ &= \sum_{i=1}^{n}\|x_i-\mathbf{W}h^i\|. \end{aligned} \tag{6.12}$$

Comparing this robust formulation with (6.1), we can see that the error for each data point is $\|x_i-\mathbf{W}h^i\|$, which is not squared, and thus the large errors due to outliers do not dominate the objective function. Here, for each view $v$, by incorporating $L_{2,1}$-norm, the robust formulation of the error function can be written as,

$$\begin{aligned} &\|\mathbf{X}^{(v)}-\mathbf{W}^{(v)}(\mathbf{Z}^{(v)})^T\mathbf{A}^T\|_{2,1}=\sum_{i=1}^{N}\sqrt{\sum_{j=1}^{d_v}(\mathbf{X}^{(v)}-\mathbf{W}^{(v)}(\mathbf{Z}^{(v)})^T\mathbf{A}^T)_{ji}^2} \\ &= \sum_{i=1}^{N}\|x_i^{(v)}-\mathbf{W}^{(v)}(z_i^{(v)})^T\mathbf{A}^T\|, \end{aligned} \tag{6.13}$$

where $z_i^{(v)}$ is the $i$th column of $\mathbf{Z}^{(v)}$. In this formulation, we can see that the error for each data is $\|x_i^{(v)}-\mathbf{W}^{(v)}(z_i^{(v)})^T\mathbf{A}^T\|$, which is not squared, and thus preventing the large errors from dominating the objective function.

Therefore, the overall error of the objective function (6.9) could be reduced greatly. Taking above into consideration, we propose the final formula as follows,

$$\begin{aligned} J = \min_{\mathbf{W}^{(v)},\mathbf{Z}^{(v)},\mathbf{Z}^*,\alpha^{(v)}\geq 0} &\sum_{v=1}^{V}\|\mathbf{X}^{(v)}-\mathbf{W}^{(v)}(\mathbf{Z}^{(v)})^T\mathbf{A}^T\|_{2,1} \\ &+\sum_{v=1}^{V}(\alpha^{(v)})^{\gamma}\|\mathbf{Z}^{(v)}\mathbf{Q}^{(v)}-\mathbf{Z}^*\|_F^2 \\ &s.t. \sum_{v=1}^{V}\alpha^{(v)}=1. \end{aligned} \tag{6.14}$$

### 6.3.2 Optimization

To solve this optimization problem, an iterative updating algorithm is presented.

When $\mathbf{Z}^*$ is fixed, for each given $v$, the computation of $\mathbf{W}^{(v)}$ or $\mathbf{Z}^{(v)}$ does not depend on $\mathbf{W}^{(v')}$ or $\mathbf{Z}^{(v')}$, where $v \neq v'$. Therefore, we use $\mathbf{X}$, $\mathbf{W}$, $\mathbf{Z}$, and $\mathbf{Q}$ to represent $\mathbf{X}^{(v)}$, $\mathbf{W}^{(v)}$, $\mathbf{Z}^{(v)}$ and $\mathbf{Q}^{(v)}$ for brevity.

The objective function is defined as follows,

$$\begin{aligned} J = \min_{\mathbf{W},\mathbf{Z},\mathbf{Z}^*,\alpha^{(v)}\geq 0} & \|\mathbf{X} - \mathbf{W}\mathbf{Z}^T\mathbf{A}^T\|_{2,1} \\ & +(\alpha^{(v)})^{\gamma}\|\mathbf{Z}\mathbf{Q} - \mathbf{Z}^*\|_F^2. \end{aligned} \tag{6.15}$$

Then the following multiplicative updating rules for $\mathbf{W}$, $\mathbf{Z}$ and $\mathbf{D}$ are applied to update their values sequentially and iteratively.

(1) Fixing $\mathbf{Z}^*$, $\mathbf{W}$, $\mathbf{Z}$ and $\alpha^{(v)}$, updating $\mathbf{D}$

$\mathbf{D} \in \mathbb{R}^{N\times N}$ is the diagonal matrix corresponding to the i-th entry with the diagonal elements given by

$$D_{ii} = \frac{1}{\|\mathbf{X}_i - \mathbf{W}(\mathbf{Z}^T\mathbf{A}^T)_i\|}. \tag{6.16}$$

(2) Fixing $\mathbf{Z}^*$, $\mathbf{Z}$, $\mathbf{D}$ and $\alpha^{(v)}$, updating $\mathbf{W}$

Let $\Phi_{i,k}$ be the Lagrange multiplier matrix for the constraint $\mathbf{W}_{i,k} \geq 0$, and $\mathbf{\Phi} = [\Phi_{i,k}]$. The Lagrange function is $L_1 = J + Tr(\mathbf{\Phi}\mathbf{W})$, we only care the terms that are relevant to $\mathbf{W}^{(v)}$.

$$\begin{aligned} L_1 = & Tr(-2\mathbf{X}\mathbf{D}\mathbf{A}\mathbf{Z}\mathbf{W}^T + \mathbf{W}\mathbf{Z}^T\mathbf{A}^T\mathbf{D}\mathbf{A}\mathbf{Z}\mathbf{W}^T) \\ & +(\alpha^{(v)})^{\gamma}Tr(\mathbf{Z}\mathbf{Q}\mathbf{Q}^T\mathbf{Z}^T - 2\mathbf{Z}\mathbf{Q}(\mathbf{Z}^*)^T) + Tr(\mathbf{\Phi}\mathbf{W}). \end{aligned} \tag{6.17}$$

Taking the derivatives of $L_1$ with respect to $\mathbf{W}$ gives

$$\frac{\partial L_1}{\partial \mathbf{W}} = -2\mathbf{X}\mathbf{D}\mathbf{A}\mathbf{Z} + 2\mathbf{W}\mathbf{Z}^T\mathbf{A}^T\mathbf{D}\mathbf{A}\mathbf{Z}+(\alpha^{(v)})^{\gamma}\mathbf{R} + \Phi, \tag{6.18}$$

where

$$\mathbf{R} = 2(\sum_{f=1}^{d_v}\mathbf{W}_{f,k}\sum_{j=1}^{N-l+c}\mathbf{Z}_{j,k}^2 - \sum_{j=1}^{N-l+c}\mathbf{Z}_{j,k}\mathbf{Z}_{j,k}^*), , \forall 1 \leq i \leq d_v, 1 \leq k \leq K. \tag{6.19}$$

Using the Kuhn-Tucker condition $\Phi_{i,k}\mathbf{W}_{i,k} = 0$, we get the following equations

**Algorithm 6.1** The algorithm of AMVNMF

```
Input:
  Data for V views $\{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, ..., \mathbf{X}^{(V)}\}$ and $\mathbf{X}^{(v)} \in \mathbb{R}^{d_v \times n}$.
  Parameter $\gamma$.
  Number of clusters $k$.
Output:
  Basis matrices $\{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, ..., \mathbf{W}^{(V)}\}$.
  Coefficient matrices $\{\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}, ..., \mathbf{Z}^{(V)}\}$.
  Consensus matrix $\mathbf{Z}^*$.
  The learned weight $\alpha^{(v)}$ for each view.
  Normalizing each view $\mathbf{X}^{(v)}$ such that $\|\mathbf{X}^{(v)}\|_1 = 1$
  Initializing $\mathbf{W}^{(v)}, \mathbf{Z}^{(v)}, \mathbf{D}^{(v)}$ and $\mathbf{Z}^*$
  Initializing the weight factor $\alpha^{(v)} = \frac{1}{V}$
  repeat
    for $v = 1$ to $V$ do
      repeat
        Fixing $\mathbf{Z}^*$, $\mathbf{W}^{(v)}$, $\mathbf{Z}^{(v)}$ and $\alpha^{(v)}$, updating $\mathbf{D}^{(v)}$ by (6.16)
        Fixing $\mathbf{Z}^*$, $\mathbf{Z}^{(v)}$, $\mathbf{D}^{(v)}$ and $\alpha^{(v)}$, updating $\mathbf{W}^{(v)}$ by (6.21)
        Normalizing $\mathbf{W}^{(v)}$ and $\mathbf{Z}^{(v)}$ as in (6.22)
        Fixing $\mathbf{Z}^*$, $\mathbf{W}^{(v)}$, $\mathbf{D}^{(v)}$ and $\alpha^{(v)}$, updating $\mathbf{Z}^{(v)}$ by (6.27)
      until (6.15) converges.
    end for
    Fixing $\mathbf{W}^{(v)}$, $\mathbf{Z}^{(v)}$, $\mathbf{D}^{(v)}$ and $\alpha^{(v)}$, updating $\mathbf{Z}^*$ by (6.29)
    Fixing $\mathbf{Z}^*$, $\mathbf{W}^{(v)}$, $\mathbf{Z}^{(v)}$ and $\mathbf{D}^{(v)}$ updating $\alpha^{(v)}$ by (6.34)
  until (6.14) converges.
```

for $\mathbf{W}_{i,k}$

$$\begin{aligned}&((\mathbf{XDAZ})_{i,k} + (\alpha^{(v)})^{\gamma}\sum_{j=1}^{N-l+c}\mathbf{Z}_{j,k}\mathbf{Z}^*_{j,k})\mathbf{W}_{i,k}\\ -&((\mathbf{WZ}^T\mathbf{A}^T\mathbf{DAZ})_{i,k} + (\alpha^{(v)})^{\gamma}\sum_{f=1}^{d_v}\mathbf{W}_{f,k}\sum_{j=1}^{N-l+c}\mathbf{Z}^2_{j,k})\mathbf{W}_{i,k} = 0.\end{aligned} \tag{6.20}$$

The following update rule can be derived based on this condition,

$$\mathbf{W}_{i,k} = \mathbf{W}_{i,k} \cdot \frac{(\mathbf{XDAZ})_{i,k} + (\alpha^{(v)})^{\gamma}\sum_{j=1}^{N-l+c}\mathbf{Z}_{j,k}\mathbf{Z}^*_{j,k}}{(\mathbf{WZ}^T\mathbf{A}^T\mathbf{DAZ})_{i,k} + (\alpha^{(v)})^{\gamma}\sum_{f=1}^{d_v}\mathbf{W}_{f,k}\sum_{j=1}^{N-l+c}\mathbf{Z}^2_{j,k}}. \tag{6.21}$$

(3) Fixing $\mathbf{Z}^*$, $\mathbf{W}$, $\mathbf{D}$ and $\alpha^{(v)}$, updating $\mathbf{Z}$

Note that $\mathbf{Z}^{(v)}$ in different views might not be comparable at the same scale. To let them theoretically meaningful for clustering, a $L_1$-norm with respect to $\mathbf{W}^{(v)}$ is proposed to constraint different $\mathbf{Z}^{(v)}$ in the same range and then compute the distance measure. The normalization is shown as follows,

$$\mathbf{W} \leftarrow \mathbf{W}\mathbf{Q}^{-1}, \mathbf{Z} \leftarrow \mathbf{Z}\mathbf{Q}. \tag{6.22}$$

Thus, the object function equals

$$\min_{\mathbf{W},\mathbf{Z},\mathbf{Z}^*,\alpha^{(v)}\geq 0} \|\mathbf{X} - \mathbf{W}\mathbf{Z}^T\mathbf{A}^T\|_{2,1} + (\alpha^{(v)})^\gamma \|\mathbf{Z} - \mathbf{Z}^*\|_F^2. \tag{6.23}$$

Let $\Psi$ be the Lagrange multiplier matrix for the constraint $\mathbf{Z} \geq 0$, and $\boldsymbol{\Psi} = [\Psi_{j,k}]$. Requiring that

$$\begin{aligned} L_2 = Tr(&-2\mathbf{X}\mathbf{D}\mathbf{A}\mathbf{Z}\mathbf{W}^T + 2\mathbf{W}\mathbf{Z}^T\mathbf{A}^T\mathbf{D}\mathbf{A}\mathbf{Z}\mathbf{W}^T) \\ &+(\alpha^{(v)})^\gamma Tr(2\mathbf{Z}\mathbf{Z}^T - 2\mathbf{Z}(\mathbf{Z}^*)^T) + Tr(\boldsymbol{\Psi}\mathbf{Z}). \end{aligned} \tag{6.24}$$

Taking derivative of $L_2$ with respect to $\mathbf{Z}$, we have

$$\begin{aligned} \frac{\partial L_2}{\partial \mathbf{Z}} &= -2\mathbf{A}^T\mathbf{D}\mathbf{X}^T\mathbf{W} + 2\mathbf{A}^T\mathbf{D}\mathbf{A}\mathbf{Z}\mathbf{W}^T\mathbf{W} \\ &\quad +(\alpha^{(v)})^\gamma(\mathbf{2Z} - \mathbf{2Z}^*) + \Psi, \\ &\quad \forall 1 \leq j \leq N - l + c, 1 \leq k \leq K. \end{aligned} \tag{6.25}$$

Using the Kuhn-Tucker condition $\Psi_{j,k}\mathbf{Z}_{j,k} = 0$, we get the following equations for $\mathbf{Z}_{j,k}$

$$\begin{aligned} &((\mathbf{A}^T\mathbf{D}\mathbf{X}^T\mathbf{W})_{j,k} + (\alpha^{(v)})^\gamma \mathbf{Z}^*_{j,k})\mathbf{Z}_{j,k} \\ -&((\mathbf{A}^T\mathbf{D}\mathbf{A}\mathbf{Z}\mathbf{W}^T\mathbf{W})_{j,k} + (\alpha^{(v)})^\gamma \mathbf{Z}_{j,k})\mathbf{Z}_{j,k} = 0. \end{aligned} \tag{6.26}$$

This leads to the updating rule as follows,

$$\mathbf{Z}_{j,k} = \mathbf{Z}_{j,k} \cdot \frac{(\mathbf{A}^T\mathbf{D}\mathbf{X}^T\mathbf{W})_{j,k} + (\alpha^{(v)})^\gamma \mathbf{Z}^*_{j,k}}{(\mathbf{A}^T\mathbf{D}\mathbf{A}\mathbf{Z}\mathbf{W}^T\mathbf{W})_{j,k} + (\alpha^{(v)})^\gamma \mathbf{Z}_{j,k}}. \tag{6.27}$$

(4) Fixing $\mathbf{W}$, $\mathbf{Z}$, $\mathbf{D}$ and $\alpha^{(v)}$, updating $\mathbf{Z}^*$

Taking the derivative of the objective function $J$ in (6.14),

$$\begin{aligned}\frac{\partial J}{\partial \mathbf{Z}^*} &= \frac{\partial \sum_{v=1}^{M} (\alpha^{(v)})^{\gamma} \|\mathbf{Z}^{(v)}\mathbf{Q}^{(v)} - \mathbf{Z}^*\|_F^2}{\partial \mathbf{Z}^*} \\ &= \sum_{v=1}^{M} (\alpha^{(v)})^{\gamma} (-2\mathbf{Z}^{(v)}\mathbf{Q}^{(v)} + 2\mathbf{Z}^*) = 0,\end{aligned} \tag{6.28}$$

we get

$$\mathbf{Z}^* = \frac{\sum_{v=1}^{M} (\alpha^{(v)})^{\gamma} \mathbf{Z}^{(v)}\mathbf{Q}^{(v)}}{\sum_{v=1}^{M} (\alpha^{(v)})^{\gamma}}. \tag{6.29}$$

(5) Fixing $\mathbf{Z}^*$, $\mathbf{W}$, $\mathbf{Z}$ and $\mathbf{D}$ updating $\alpha^{(v)}$

We only consider the term that relevant to $\alpha$, thus, it is equal to minimize

$$J = \sum_{v=1}^{M} (\alpha^{(v)})^{\gamma} \|\mathbf{Z}^{(v)}\mathbf{Q}^{(v)} - \mathbf{Z}^*\|_F^2. \tag{6.30}$$

By setting $\mathbf{G}^{(v)} = \|\mathbf{Z}^{(v)}\mathbf{Q}^{(v)} - \mathbf{Z}^*\|_F^2$, and due to $\sum_{v=1}^{M} \alpha^{(v)} = 1$, (6.30) equals

$$J = \sum_{v=1}^{M} (\alpha^{(v)})^{\gamma} \mathbf{G}^{(v)} - \lambda \sum_{v=1}^{M} (\alpha^{(v)} - 1). \tag{6.31}$$

The solution can be obtained by the following updating rule:

$$\frac{\partial J}{\partial \alpha^{(v)}} = \gamma \mathbf{G}^{(v)} (\alpha^{(v)})^{\gamma - 1} - \lambda = 0. \tag{6.32}$$

Thus,

$$\alpha^{(v)} = (\frac{\lambda}{\gamma \mathbf{G}^{(v)}})^{\frac{1}{\gamma-1}}. \tag{6.33}$$

Substituting the result in (6.33) to the condition, i.e., $\sum_{v=1}^{M} \alpha^{(v)} = 1$, $\alpha^{(v)}$ can be obtained as

$$\alpha^{(v)} = \frac{(\gamma \mathbf{G}^{(v)})^{\frac{1}{1-\gamma}}}{\sum_{v=1}^{M} (\gamma \mathbf{G}^{(v)})^{\frac{1}{1-\gamma}}}. \tag{6.34}$$

We summarize the proposed algorithm in the Algorithm 6.1.

## 6.4 Experiments

### 6.4.1 Description of datasets

In this chapter, we compare our method, AMVNMF, to the state-of-the-art methods on five benchmark multi-view datasets.

• **SensIT**: This is obtained from wireless distributed sensor networks (WDSN). It uses two sensors, acoustic and seismic, to record different signals and to classify three types of vehicle in an intelligent transportation system. We download the processed data from LIBSVM [10] and randomly sample 100 data for each class. Thus, we utilize 300 data samples, 2 views and 3 classes.

• **ORL**: This data set consists of 400 facial images belonging to 40 different subjects. For each subject, the images are in great varieties because of different taking time with changing lighting variance, facial details and facial expressions. The images are gray scale and have been normalized to $112 \times 92$ pixels. The first view contains the raw pixel values and the second view contains GIST [38].

• **Reuters**: This data set contains feature characteristics of documents originally written in five different languages, and their translations, over a common set of 6 categories. We use the original English documents as the first view, their French and German translations are regarded as the second and third views. We randomly sample 600 documents from this collection, with each of the 6 clusters having 100 documents [65]. In the experiment, the frequency of words is used as the features of each document,.

• **Citeseer** and • **Cora** are are composed of publications. These publications are linked via citations. Both of them take contents and citations as two views.

### 6.4.2 Methods to compare

To demonstrate how the clustering performance can be improved by the proposed approach, we compared with the following algorithms:

1. Best Single View (BSV): Using the most informative view which achieves the best performance with our AMVNMF.

2. ConCNMF: The method firstly concatenates the features of all views and applies CNMF [64] to extract the low dimensional subspace representation.

3. MultiNMF: The NMF-based multi-view clustering method proposed in

[65].

4. RMKMC : The multi-view $k$-means proposed in [8].

5. CoRegSPC : The co-regularized pairwise multi-view spectral clustering method proposed in [53].

Table 6.1: Clustering results on five real-world datasets (%)

| Metrics | Datasets | BSV | ConCNMF | MultiNMF | RMKMC | CoRegSPC | AMVNMF |
|---|---|---|---|---|---|---|---|
| AC | SensIT | 69.66 | 52.30 | 55.04 | 60.07 | 61.67 | **71.33** |
| | ORL | 74.3 | 49.59 | 54.6 | 45.5 | 78.20 | **80.5** |
| | Reuters | 57.50 | 49.59 | 51.87 | 39.80 | 54.40 | **59.88** |
| | Citeseer | 50.08 | 40.70 | 34.36 | 43.21 | 47.42 | **53.14** |
| | Cora | 33.42 | 32.42 | 44.83 | 43.90 | 37.20 | **48.71** |
| NMI | SensIT | 30.14 | 15.67 | 19.87 | 14.84 | 17.75 | **31.73** |
| | ORL | 89.29 | 51.32 | 75.23 | 65.34 | 90.84 | **91.73** |
| | Reuters | 41.95 | 30.37 | 36.14 | 21.82 | 36.57 | **42.75** |
| | Citeseer | 21.38 | 13.34 | 20.97 | 20.61 | 21.10 | **26.13** |
| | Cora | 26.73 | 9.87 | 27.95 | 21.27 | 15.44 | **34.59** |

### 6.4.3 Experimental setup

Prior to clustering, for each type of features, we normalize the data first, making the sum of values of each view equal to 1. For fair comparison with previous works, we follow the experimental settings as in [65]. In our experiments, the parameter $\gamma$ in (6.34) varies in the range from 2 to 902 with an incremental step 100, and the best parameter $\gamma$ is selected in the smaller and more robust ranges for all views and data sets. The parameters for all competitors are also tuned to achieve the best performance. 30% of labeled data are randomly picked up from each view as priors for semi-supervised learning (AMVNMF and ConCNMF). Since clustering performances depend on the initializations, we repeat each method 10 times with random initializations and report the average performance.

### 6.4.4 Results analysis

Table 6.1 summarizes the clustering performances of different algorithms on the five datasets. It is clear to see that AMVNMF outperforms the second best algorithm in all cases. Furthermore, BSV always gets the second best performance. It outperforms other multi-view methods greatly, i.e., 7.99%/10.27% on SensIT and 3.10%/5.38% on Reuters in terms of AC and NMI, respectively. This is mainly due to AMVNMF guarantees that all the data sharing the same labels are grouped together, regardless they are come from the same or different views. Therefore, both AMVNMF and BSV ( running AMVNMF with single view) produce superior results.

### 6.4.5 Parameter analysis

We show the parameter tuning on SensIT, ORL and Reuters as examples in Figure 6.2. The parameter $\gamma$ controls the distribution of weight factors $\alpha^{(v)}$ for different views. More preciously, when $\gamma \rightarrow \infty$, the weight for all views is equal. When $\gamma \rightarrow 1$, the weight factor of 1 is assigned to the most important view whose $\mathbf{G}^{(v)}$ value is the smallest and 0 is assigned to the weights of the other views. Hence, this strategy allows well adjusting the ratio of each view and saves the cost of tuning multiple parameters. As shown in Figure 6.2, AMVNMF performs stably with varying $\gamma$ (from 2 to 902). Please note that even the worst results of AMVNMF are always better than other approaches in most cases.

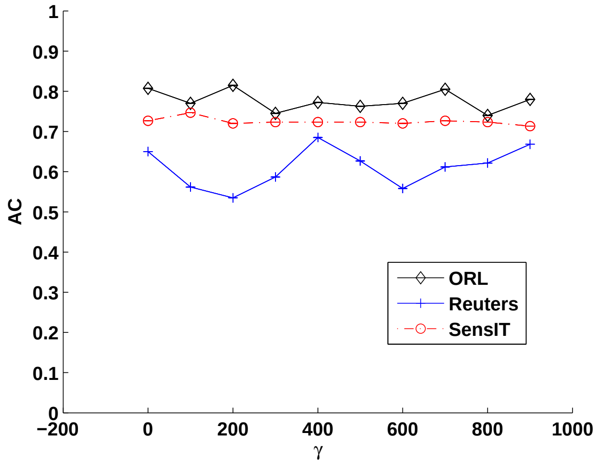


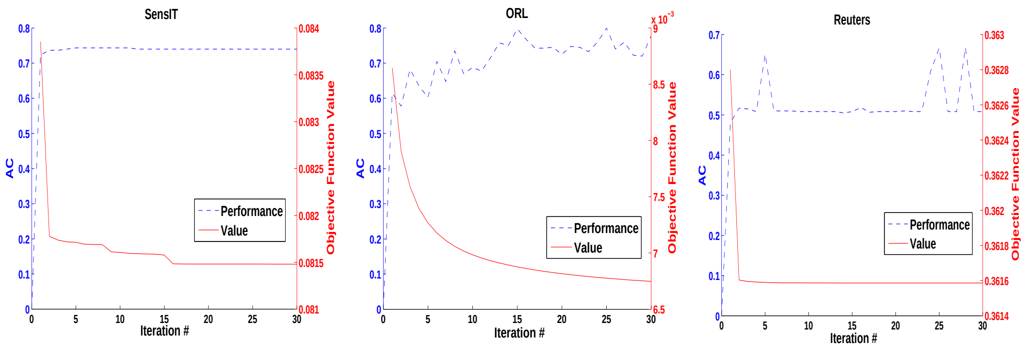
Figure 6.2: Performance of AMVNMF w.r.t. parameter $\gamma$.

### 6.4.6 Study of convergence

The updating rules for minimizing the objective function of AMVNMF in (6.14) are essentially iterative. We now check its convergence property. Fig. 6.3 shows the convergence curve together with performance. The blue solid line shows the value of the objective function and the red dashed line indicates the accuracy. It can be seen that the value of the objective function decreases steadily with more iterations and converges after around 20 times.

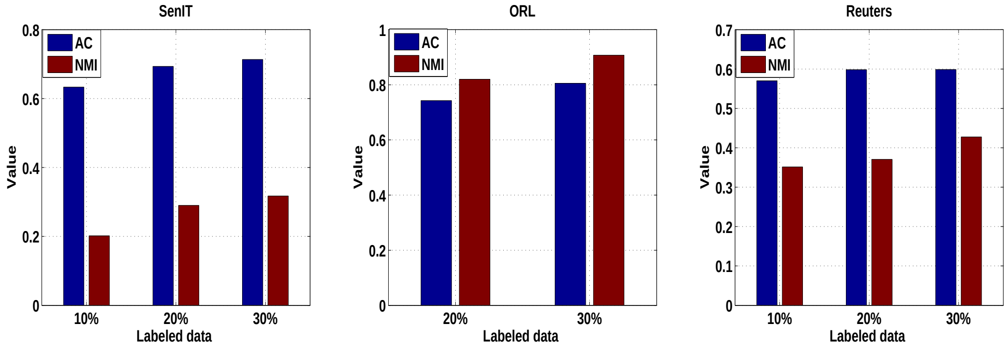
### 6.4.7 Effect of labeled data

Since AMVNMF is a semi-supervised method, we also randomly pick up 10% and 20% labeled data to further demonstrate the benefits of priors. Notice that ORL has only 10 images for each category, thus 10% gives one image only. However, one label is meaningless for AMVNMF since this algorithm maps the images with the same label onto the same coordinate in the new representation space. Thus, we omit the result with 10% labeled data. From Figure 6.4, it can be seen that both AC and NMI are improved with more labeled data. Also, it is worth pointing out that even with only 10% labeled data, AMVNMF performs better than other approaches when 30% labeled data are applied. For example, for the SensIT dataset, AMVNMF achieves 62% AC and 20% NMI with 10% labeled data, which is better than the best performance of other approaches, i.e., 61.67% AC and 19.87% NMI (as shown in Table 6.1).



(a) (b) (c)

Figure 6.3: Convergence and corresponding performance curve.



(a) (b) (c)

Figure 6.4: Performance of AMVNMF w.r.t. labeled data.

## 6.5 Conclusion

A novel NMF-based multi-view method, AMVNMF, is proposed in this chapter. It efficiently learns the underlying clustering structure embedded in multiple views, by regularizing the new representation matrices learnt from different views towards a common consensus. The advantages of AMVNMF are shown in three aspects. First, it guarantees that labeled data come with multiple views can be clustered into the same low-dimension space. Second, it learns each view's corresponding weight adaptively with a single parameter $\gamma$. Third, it handles the noises more effectively.

# Chapter 7

# Conclusion and Future Work

## 7.1 Conclusion

Data clustering is the task of partitioning data into different groups such that the data in the same group are highly similar. In many real applications, such as information retrieval, digital image processing and bioinformatics, it has been an active research field where many approaches have been developed using various objective functions. Recently there has been significant development in the use of non-negative matrix factorization (NMF) methods for various clustering tasks. NMF approximates a nonnegative data matrix by a product of two low-rank factorizing matrices under nonnegative constraints. The additive nature of NMF can often result in parts-based representation of the data, and this property is especially desired for data clustering.

This thesis has presented advances in NMF with application on data clustering. Firstly, the research background is introduced in the Chapter 1. Acomprehensive review of standard NMF as well as existing different variants on NMF formulations is then provided in Chapter 2. Generally, NMF-based approaches for data clustering could be divided into two steps: 1) They all aim to find a lower dimensional feature representation matrix via factorizing the input high-dimensional feature matrix; 2) A post-processing such as $k$-means is conducted on the representation matrix to achieve the final clustering results. The first step is more important as the success of the NMFs largely depends on learning an effective feature representation matrix. In Chapter 3, 4, 5 and 6, we have presented four novel approaches by exploring more useful feature information for accurate representation learning. In detail, considering that real data are usually complex and contain various components (e.g. face images have expressions and genders), a novel multi-component NMF approach has been proposed to explore the semantic information of multiple components as well as the diversity among them in chapter 3. Different from current approaches that factorizing the data matrix into a single basis and representation matrix, MCNMF learns multiple representations based on different basis matrices. The diverse information contained by the multiple learned representations are enforced with the Hilbert-Schmidt Independence Criterion (HSIC) [32] as a diversity regularization term. Thus, MCNMF captures more comprehensive information by integrating these multiple representations and hence increase clustering accuracy. Besides, clustering is conducted on each learned representation, so that multiple semantic clusters are achieved simultaneously with each one represents one property of data. However, MCNMF assumes that features of each data points are independently distribution, which fails to take full advantage of the sequential nature embedded in the sequential data such as motion captures. To fully exploits this nature as prior information to guide for more accurate learning, an ordered NMF is proposed in chapter 4. A $L_{2,1}$-norm based neighbour penalty term is proposed and incorporated to standard NMF to enforce the similarity of neighbouring data presentations, so that ORNMF has achieved more discriminative and explicit data representations. In fact, both MCNMF and ORNMF are single-view approaches which can deal with one type of feature of data only, such as pixels of images. Given that real-world datasets are often comprised of multiple features or views which describe data from various perspectives, it is important to exploit diversity from multiple views for comprehensive and accurate data representations. Hence, in chapter 5, we have proposed a diverse multi-view NMF(DiNMF) to exploit diverse information among multiple views of data. Through enhancing each view's independence and co-regularizing different corresponding representations, the mutually redundancy are reduced and diverse information are guaranteed by DiNMF. DiNMF is also computationally linear thus has good scalability to large-scale datasets. Since DiNMF fails to discover intrinsic geometrical structure of the data in each view, we have further proposed a Locality Preserved DiNMF (LP-DiNMF) to ensure diversity from multiple views while preserving the local geometry structure of data in each view, which lead to a more accurate clustering results.

Essentially, all the three approaches, i.e., MCNMF, ORNMF and DiNMF, could be regarded as unsupervised approaches. They are not able to utilize priors, such as label information, to guide learning process when the information are available, although some literatures have shown that utilizing a small amount of labeled data can produce considerable improvements in learning accuracy [2, 119]. Therefore, in chapter 6, we have proposed a novel semi-supervised multi-view NMF approach, called Adaptive Multi-view NMF (AMVNMF) to explore the effectiveness by incorporating label information. With a constructed constraint label matrix, AMVNMF ensures data are of the same label to have the same new representation regardless whether or not they come from the same view, which enhances the discriminability of representations. Moreover, considering that different views may have different contributions to the performance, AMVNMF uses a single parameter adjusts weight factor of each view automatically which saves the cost for parameter tuning.

To validate the effectiveness of the proposed approaches, we have conducted experiments on several benchmark datasets in comparison with existing NMF-based approaches. Experiment results have well demonstrated that the proposed approaches not only have achieved higher clustering accuracies than state-of-the-art approaches, but also converge fast. Besides, their performance are robust to parameters tuning, which allows wide applications in reality.

## 7.2 Future Work

Although the proposed approaches have shown better performance in comparison with state-of-the-arts, there are still some rooms for improvement that provide us research directions for future work. Specifically, they mainly include:

- **Online NMF for streaming data**

All the NMF approaches introduced in this thesis are designed for static data. This requires the input data matrix to reside in the memory during processing, which could be problematic when the datasets are huge(e.g., they may not even fit in the memory). Moreover, in modern applications, the data often arrive in a streaming fashion and may not be stored on disk [118]. For example, in the news mining problem domain, the news articles on a certain event appear one after another, reflecting the development of an event. In the future, we tend to research on online NMF learning to process steaming data one by one (or chunk by chunk), as well as learn the representation matrix and update the basis matrix simultaneously.

- **Improvements on robustness for complex noise**

In reality, data are often contaminated by different noises or outliers. Under different assumptions of the noise and outliers distribution, the loss functions are in various forms. Generally, the loss function used in both Chapter 2 and Chapter 3 is $L_2$-norm, which is only optimal for Gaussian noise. $L_{2,1}$-norm adopted in Chapter 4 is only for data that are corrupted in columns. However, real data are often corrupted by noise with an unknown distribution. Then any specific form of loss function for one specific kind of noise often fails to tackle such real data with complex noise. Therefore, how to improve the robustness of NMF to adapt more complex noise will be another research direction.

- **Multi-view learning with incomplete views**

With the advance of technology, real data are often with multiple modalities or coming from multiple sources which are called multi-view data. The proposed approaches in chapter 2 and 3 deal with multi-view data are based on the assumption that all of the views of data are complete, i.e., each instance appears in all views. However, in real-world applications, due to the nature of the data or the cost of data collection, some views may suffer from the incompleteness of data. For example, one news story may be reported by different news sources (views), but not all the news stories are covered by all the news sources, i.e., each news source cannot cover all the news stories. Thus, all the views are incomplete. Dealing with multi-view data with incomplete views could be a potential direction in near future.

# Bibliography

[1] C. C. Aggarwal and C. K. Reddy. *Data clustering: algorithms and applications*. CRC Press, 2013. 18

[2] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *The Journal of Machine Learning Research*, 7:2399–2434, 2006. 29, 32, 78, 94, 112

[3] V. Blanz, M. J. Tarr, H. H. Bülthoff, and T. Vetter. What object attributes determine canonical views? *Perception-London*, 28(5):575–600, 1999. 95

[4] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998. 30, 68

[5] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004. 41, 55, 74, 80

[6] D. Cai, X. He, J. Han, and T. S. Huang. Graph regularized nonnegative matrix factorization for data representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1548–1560, 2011. 27, 29, 40, 45, 59, 78, 85

[7] X. Cai, F. Nie, W. Cai, and H. Huang. Heterogeneous image features integration via multi-modal semi-supervised learning model. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1737–1744, 2013. 84

[8] X. Cai, F. Nie, and H. Huang. Multi-view k-means clustering on big data. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 2598–2604. AAAI Press, 2013. 85, 106

[9] X. Cao, C. Zhang, H. Fu, S. Liu, and H. Zhang. Diversity-induced multi-view subspace clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–594, 2015. 30, 45

[10] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm. 105

[11] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan. Multi-view clustering via canonical correlation analysis. In *Proceedings of the 26th annual international conference on machine learning*, pages 129–136. ACM, 2009. 68

[12] A. M. Cheriyadat and R. J. Radke. Non-negative matrix factorization of partial track data for motion segmentation. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 865–872. IEEE, 2009. 51

[13] V. C. Cheung, K. Devarajan, G. Severini, A. Turolla, and P. Bonato. Decomposing time series data by a non-negative matrix factorization algorithm with temporally constrained coefficients. In *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pages 3496–3499. IEEE, 2015. 60

[14] Y. Cho and L. K. Saul. Nonnegative matrix factorization for semi-supervised dimensionality reduction. *arXiv preprint arXiv:1112.3714*, 2011. 29

[15] M. Chu and R. Plemmons. Nonnegative matrix factorization and applications. *Bulletin of the International Linear Algebra Society*, 34:2–7, 2005. 26

[16] A. Cichocki, R. Zdunek, A. H. Phan, and S.-i. Amari. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation.* John Wiley & Sons, 2009. 26

[17] N. Courty, X. Gong, J. Vandel, and T. Burger. Saga: Sparse and geometry-aware non-negative matrix factorization through non-linear local embedding. *Machine Learning*, 97(1-2):205–226, 2014. 19

[18] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005. 30, 67, 84

[19] C. Ding, X. He, and H. D. Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proceedings of the 2005 SIAM International Conference on Data Mining*, pages 606–610. SIAM, 2005. 27

[20] C. Ding, T. Li, M. Jordan, et al. Convex and semi-nonnegative matrix factorizations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(1):45–55, 2010. 27, 43, 44, 56, 57

[21] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 126–135. ACM, 2006. 27, 33, 87

[22] L. Du, X. Li, and Y.-D. Shen. Robust nonnegative matrix factorization via half-quadratic minimization. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 201–210. IEEE, 2012. 28

[23] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. John Wiley & Sons, 2012. 19

[24] J. Eggert and E. Korner. Sparse coding and nmf. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 4, pages 2529–2533. IEEE, 2004. 28

[25] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006. 84

[26] D. G. Ferrari and L. N. De Castro. Clustering algorithm selection by meta-learning systems: A new distance-based problem characterization and ranking combination methods. *Information Sciences*, 301:181–194, 2015. 18

[27] C. Févotte, N. Bertin, and J.-L. Durrieu. Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis. *Neural computation*, 21(3):793–830, 2009. 27

[28] H. Gao, F. Nie, X. Li, and H. Huang. Multi-view subspace clustering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4238–4246, 2015. 30

[29] R. Gaujoux and C. Seoighe. Semi-supervised nonnegative matrix factorization for gene expression deconvolution: a case study. *Infection, Genetics and Evolution*, 12(5):913–921, 2012. 94

[30] A. Gersho and R. M. Gray. *Vector quantization and signal compression*. Springer, 1992. 19

[31] A. Gersho and R. M. Gray. *Vector quantization and signal compression*, volume 159. Springer Science & Business Media, 2012. 19

[32] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *International conference on algorithmic learning theory*, pages 63–77. Springer, 2005. 38, 39, 111

[33] N. Guan, D. Tao, L. Lan, Z. Luo, and X. Yang. Activity recognition in still images with transductive non-negative matrix factorization. In *Computer Vision-ECCV 2014 Workshops*, pages 802–817. Springer, 2014. 51

[34] N. Guan, D. Tao, Z. Luo, and B. Yuan. Manifold regularized discriminative nonnegative matrix factorization with fast gradient descent. *IEEE Transactions on Image Processing*, 20(7):2030–2048, 2011. 27

[35] X. Guo. Exclusivity regularized machine. *arXiv preprint arXiv:1603.08318*, 2016. 71

[36] Y. Guo, J. Gao, F. Li, S. Tierney, and M. Yin. Low rank sequential subspace clustering. In *Neural Networks (IJCNN), 2015 International Joint Conference on*, pages 1–8. IEEE, 2015. 51

[37] A. B. Hamza and D. J. Brady. Reconstruction of reflectance spectra using robust nonnegative matrix factorization. *IEEE Transactions on Signal Processing*, 54(9):3637–3642, 2006. 28

[38] D. Hidru and A. Goldenberg. Equinmf: Graph regularized multiview nonnegative matrix factorization. *arXiv preprint arXiv:1409.4018*, 2014. 105

[39] R. Hochberg. Matrix multiplication with cudaa basic introduction to the cuda programming model, 2012. 44

[40] P. O. Hoyer. Non-negative sparse coding. In *Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on*, pages 557–565. IEEE, 2002. 28

[41] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *The Journal of Machine Learning Research*, 5:1457–1469, 2004. 27, 28

[42] J. Huang, F. Nie, H. Huang, and C. Ding. Robust manifold nonnegative matrix factorization. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 8(3):11, 2014. 29, 87

[43] S. Jia and Y. Qian. Constrained nonnegative matrix factorization for hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 47(1):161–173, 2009. 27

[44] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2005. 19

[45] I. T. Jolliffe. Principal component analysis and factor analysis. In *Principal component analysis*, pages 115–128. Springer, 1986. 19

[46] M. M. Kalayeh, H. Idrees, and M. Shah. Nmf-knn: Image annotation using weighted multi-view non-negative matrix factorization. In *2014 IEEE*

*Conference on Computer Vision and Pattern Recognition*, pages 184–191. IEEE, 2014. 27, 32

[47] J. Kim and H. Park. Sparse nonnegative matrix factorization for clustering. 2008. 28

[48] Y.-D. Kim and S. Choi. Weighted nonnegative matrix factorization. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 1541–1544. IEEE, 2009. 27

[49] K. Kimura, M. Kudo, and Y. Tanaka. A column-wise update algorithm for nonnegative matrix factorization in bregman divergence with an orthogonal constraint. *Machine Learning*, pages 1–22, 2016. 51

[50] D. Kong, C. Ding, and H. Huang. Robust nonnegative matrix factorization using l21-norm. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 673–682. ACM, 2011. 28, 40, 45, 53, 57, 58, 59

[51] D. Kuang, C. Ding, and H. Park. Symmetric nonnegative matrix factorization for graph clustering. In *Proceedings of the 2012 SIAM international conference on data mining*, pages 106–117. SIAM, 2012. 27

[52] A. Kuhn, S. Ducasse, and T. Gírba. Semantic clustering: Identifying topics in source code. *Information and Software Technology*, 49(3):230–243, 2007. 18

[53] A. Kumar, P. Rai, and H. Daume. Co-regularized multi-view spectral clustering. In *Advances in Neural Information Processing Systems*, pages 1413–1421, 2011. 85, 106

[54] L. Lan, N. Guan, X. Zhang, D. Tao, and Z. Luo. Soft-constrained nonnegative matrix factorization via normalization. In *Neural Networks (IJCNN), 2014 International Joint Conference on*, pages 3025–3030. IEEE, 2014. 27

[55] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999. 19, 25, 26, 36, 54, 59, 85

[56] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001. 40, 42, 45, 56, 75, 82, 85

[57] K. Leung and C. Leckie. Unsupervised anomaly detection in network intrusion detection using clusters. In *Proceedings of the Twenty-eighth Australasian conference on Computer Science-Volume 38*, pages 333–342. Australian Computer Society, Inc., 2005. 18

[58] P. Li, J. Bu, C. Chen, Z. He, and D. Cai. Relational multimanifold coclustering. *IEEE Transactions on cybernetics*, 43(6):1871–1881, 2013. 29

[59] P. Li, J. Bu, C. Chen, C. Wang, and D. Cai. Subspace learning via locally constrained a-optimal nonnegative projection. *Neurocomputing*, 115:49–62, 2013. 29

[60] P. Li, J. Bu, B. Xu, B. Wang, and C. Chen. Locally discriminative spectral clustering with composite manifold. *Neurocomputing*, 119:243–252, 2013. 29, 78

[61] S. Z. Li, X. Hou, H. Zhang, and Q. Cheng. Learning spatially localized, parts-based representation. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–207. IEEE, 2001. 27

[62] Q.-L. Lin, B. Sheng, Y. Shen, Z.-F. Xie, Z.-H. Chen, and L.-Z. Ma. Fast image correspondence with global structure projection. *Journal of Computer Science and Technology*, 27(6):1281–1288, 2012. 19

[63] C. H. Liu and A. Chaudhuri. Reassessing the 3/4 view effect in face recognition. *Cognition*, 83(1):31–48, 2002. 95

[64] H. Liu, Z. Wu, X. Li, D. Cai, and T. S. Huang. Constrained nonnegative matrix factorization for image representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(7):1299–1311, 2012. 19, 27, 29, 33, 51, 63, 87, 94, 96, 105

[65] J. Liu, C. Wang, J. Gao, and J. Han. Multi-view clustering via joint nonnegative matrix factorization. In *Proc. of SDM*, volume 13, pages 252–260. SIAM, 2013. 27, 31, 32, 84, 85, 98, 105, 106

[66] T. Liu, M. Gong, and D. Tao. Large-cone nonnegative matrix factorization. *IEEE Transactions on Neural Networks and Learning Systems*, 2016. 45

[67] W. Liu, J. He, and S.-F. Chang. Large graph construction for scalable semi-supervised learning. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 679–686, 2010. 27

[68] Y. Liu, R. Jin, and L. Yang. Semi-supervised multi-label learning by constrained non-negative matrix factorization. In *AAAi*, volume 6, pages 421–426, 2006. 94

[69] A. Liutkus, D. Fitzgerald, and R. Badeau. Cauchy nonnegative matrix factorization. In *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2015 IEEE Workshop on*, pages 1–5. IEEE, 2015. 45

[70] X. Long, H. Lu, Y. Peng, and W. Li. Graph regularized discriminative non-negative matrix factorization for face recognition. *Multimedia Tools and Applications*, 72(3):2679–2699, 2014. 51

[71] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 30, 67

[72] Q. Mo and B. A. Draper. Semi-nonnegative matrix factorization for motion segmentation with missing data. In *Computer Vision–ECCV 2012*, pages 402–415. Springer, 2012. 51

[73] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber. Documentation mocap database hdm05. 2007. 66

[74] C.-T. Nguyen, D.-C. Zhan, and Z.-H. Zhou. Multi-modal image annotation with multi-instance multi-label lda. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 1558–1564. AAAI Press, 2013. 30

[75] B. Ni, P. Moulin, and S. Yan. Order-preserving sparse coding for sequence classification. In *Computer Vision–ECCV 2012*, pages 173–187. Springer, 2012. 51

[76] F. Nie, H. Wang, H. Huang, and C. Ding. Early active learning via robust representation and structured sparsity. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 1572–1578. AAAI Press, 2013. 53

[77] F. Nie, J. Yuan, and H. Huang. Optimal mean robust principal component analysis. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1062–1070, 2014. 60

[78] D. Niu, J. G. Dy, and M. I. Jordan. Multiple non-redundant spectral clustering views. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 831–838, 2010. 39

[79] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175, 2001. 30, 67

[80] W. Ou, S. Yu, G. Li, J. Lu, K. Zhang, and G. Xie. Multi-view non-negative matrix factorization by patch alignment framework with view consistency. *Neurocomputing*, 2016. 32

[81] P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994. 25

[82] M. D. Plummer and L. Lovász. *Matching theory*. Elsevier, 1986. 33

[83] I. Ramirez, P. Sprechmann, and G. Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3501–3508. IEEE, 2010. 18

[84] F. Shahnaz, M. W. Berry, V. P. Pauca, and R. J. Plemmons. Document clustering using nonnegative matrix factorization. *Information Processing & Management*, 42(2):373–386, 2006. 27

[85] F. Shang, L. Jiao, J. Shi, and J. Chai. Robust positive semidefinite l-isomap ensemble. *Pattern Recognition Letters*, 32(4):640–649, 2011. 29, 78

[86] J. Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000. 59

[87] A. P. Singh and G. J. Gordon. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 650–658. ACM, 2008. 85

[88] L. Song, A. Smola, A. Gretton, K. M. Borgwardt, and J. Bedo. Supervised feature selection via dependence estimation. In *Proceedings of the 24th international conference on Machine learning*, pages 823–830. ACM, 2007. 39

[89] A. Sturn, J. Quackenbush, and Z. Trajanoski. Genesis: cluster analysis of microarray data. *Bioinformatics*, 18(1):207–208, 2002. 18

[90] D. Tao, X. Li, X. Wu, and S. J. Maybank. Geometric mean for subspace selection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(2):260–274, 2009. 19

[91] S. Tierney, J. Gao, and Y. Guo. Subspace clustering for sequential data. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1019–1026. IEEE, 2014. 51, 61

[92] I. Tosic and P. Frossard. Dictionary learning. *IEEE Signal Processing Magazine*, 28(2):27–38, 2011. 18

[93] J. A. Tropp. Literature survey: Nonnegative matrix factorization. *University of Texas at Asutin*, 2003. 26

[94] J. A. Tropp. Just relax: Convex programming methods for identifying sparse signals in noise. *IEEE transactions on information theory*, 52(3):1030–1051, 2006. 28

[95] K. Wagstaff and C. Cardie. Clustering with instance-level constraints. *AAAI/IAAI*, 1097, 2000. 95

[96] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl, et al. Constrained k-means clustering with background knowledge. In *ICML*, volume 1, pages 577–584, 2001. 95

[97] H. Wang, F. Nie, and H. Huang. Multi-view clustering and feature learning via structured sparsity. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 352–360, 2013. 30, 87

[98] J. Wang, F. Tian, C. H. Liu, and X. Wang. Robust semi-supervised nonnegative matrix factorization. In *Neural Networks (IJCNN), 2015 International Joint Conference on*, pages 1–8. IEEE, 2015. 94

[99] J. Wang, X. Wang, F. Tian, C. H. Liu, H. Yu, and Y. Liu. Adaptive multi-view semi-supervised nonnegative matrix factorization. In *International Conference on Neural Information Processing*, pages 435–444. Springer, 2016. 32

[100] J. J.-Y. Wang, H. Bensmail, and X. Gao. Multiple graph regularized nonnegative matrix factorization. *Pattern Recognition*, 46(10):2840–2847, 2013. 29

[101] J. J.-Y. Wang and X. Gao. Beyond cross-domain learning: Multiple-domain nonnegative matrix factorization. *Engineering Applications of Artificial Intelligence*, 28:181–189, 2014. 30

[102] J. J.-Y. Wang, J. Z. Huang, Y. Sun, and X. Gao. Feature selection and multi-kernel learning for adaptive graph regularized nonnegative matrix factorization. *Expert Systems with Applications*, 42(3):1278–1286, 2015. 29

[103] W. Wang and Z.-H. Zhou. Analyzing co-training style algorithms. In *Machine Learning: ECML 2007*, pages 454–465. Springer, 2007. 68

[104] X.-Y. Wang and J. M. Garibaldi. Simulated annealing fuzzy clustering in cancer diagnosis. *Informatica*, 29(1), 2005. 18

[105] Y.-X. Wang and Y.-J. Zhang. Nonnegative matrix factorization: A comprehensive review. *Knowledge and Data Engineering, IEEE Transactions on*, 25(6):1336–1353, 2013. 27

[106] F. Wu, Y. Hu, J. Gao, Y. Sun, and B. Yin. Ordered subspace clustering with block-diagonal priors. *Cybernetics, IEEE Transactions on*, 2015. 51, 59, 64

[107] R. Xia, Y. Pan, L. Du, and J. Yin. Robust multi-view spectral clustering via low-rank and sparse decomposition. In *AAAI Conference on Artificial Intelligence*, 2014. 30, 85

[108] Y.-H. Xiao, Z.-F. Zhu, Y. Zhao, and Y.-C. Wei. Class-driven non-negative matrix factorization for image representation. *Journal of Computer Science and Technology*, 28(5):751–761, 2013. 19

[109] C. Xu, D. Tao, and C. Xu. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634*, 2013. 13, 30, 31

[110] C. Xu, D. Tao, and C. Xu. Multi-view self-paced learning for clustering. In *Proceedings of the 24th International Conference on Artificial Intelligence. AAAI Press*, pages 3974–3980, 2015. 30

[111] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273. ACM, 2003. 27

[112] J. Yang and J. Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 587–596. ACM, 2013. 27

[113] S. Yang, C. Hou, C. Zhang, and Y. Wu. Robust non-negative matrix factorization via joint sparse and graph regularization for transfer learning. *Neural Computing and Applications*, 23(2):541–559, 2013. 28

[114] H. Zhang, Z.-J. Zha, Y. Yang, S. Yan, and T.-S. Chua. Robust (semi) nonnegative graph embedding. *IEEE Transactions on Image Processing*, 23(7):2996–3012, 2014. 94

[115] L. Zhang, Z. Chen, M. Zheng, and X. He. Robust non-negative matrix factorization. *Frontiers of Electrical and Electronic Engineering in China*, 6(2):192–200, 2011. 27, 28

[116] X. Zhang, L. Zhao, L. Zong, X. Liu, and H. Yu. Multi-view clustering via multi-manifold regularized nonnegative matrix factorization. In *Data Mining (ICDM), 2014 IEEE International Conference on*, pages 1103–1108. IEEE, 2014. 32, 78, 85, 87

[117] Y. Zhang and Z.-H. Zhou. Multilabel dimensionality reduction via dependence maximization. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 4(3):14, 2010. 39

[118] S. Zhong. Efficient streaming text clustering. *Neural Networks*, 18(5):790–798, 2005. 112

[119] X. Zhu, Z. Ghahramani, J. Lafferty, et al. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, volume 3, pages 912–919, 2003. 32, 94, 112