Nonnegative Matrix Factorization: Models, Algorithms and Applications

Zhong-Yuan Zhang[1]

School of Statistics, Central University of Finance and Economics, P.R.China,
zhyuanzh@gmail.com

Abstract. In recent years, Nonnegative Matrix Factorization (NMF) has become a popular model in data mining society. NMF aims to extract hidden patterns from a series of high-dimensional vectors automatically, and has been applied for dimensional reduction, unsupervised learning (clustering, semi-supervised clustering and co-clustering, etc.) and prediction successfully. This chapter surveys NMF in terms of the model formulation and its variations and extensions, algorithms and applications, as well as its relations with K-means and Probabilistic Latent Semantic Indexing (PLSI). In summary, we draw the following conclusions: 1) NMF has a good interpretability due to its nonnegative constraints; 2) NMF is very flexible regarding the choices of its objective functions and the algorithms employed to solve it; 3) NMF has a variety of applications; 4) NMF has a solid theoretical foundation and a close relationship with the existing state-of-the-art unsupervised learning models. However, as a new and developing technology, there are still many interesting open issues remained unsolved and waiting for research from theoretical and algorithmic perspectives.

1 Introduction

Nonnegative Matrix Factorization (NMF,[1–3]) is evolved from Principal Component Analysis (PCA, [4, 5]). PCA is one of the basic techniques for extracting the principal components (basic factors) from a series of vectors such that each vector is the linear combination of the components, in other words, PCA tries to give the best low dimensional representation with a common basis for a set of vectors. Formally, given a set of samples $\{x_i, i = 1, 2, \cdots, m\}$ in $\mathbb{R}^n$, PCA aims to provide the best linear approximation of the samples in a lower dimensional space, say $\mathbb{R}^k$. This problem can be represented as a nonlinear programming problem: $\min\limits_{\mu,\{\lambda_i\},V} \sum_{i=1}^m \|x_i - \mu - V\lambda_i\|_2^2$, where μ is column vector of size $n \times 1$, V is matrix of size $n \times k$ and column orthogonal ($V^TV = I$), and each λ_i, $i = 1, 2, \cdots, m$ is column vector of size $k \times 1$. Fixing μ and V, one can get the optimal solution of $\lambda_i = V^T(x_i - \bar{x})$, $i = 1, 2, \cdots, m$, where $\bar{x} = \sum\limits_i x_i/m$; similarly, fixing λ_i and V, one can get the optimal solution of $\mu = \bar{x}$. Hence the optimization problem can be re-written as:

$$\min_{V} \sum_{i=1}^{m} \|(x_i - \bar{x}) - VV^T(x_i - \bar{x})\|_2^2. \tag{1}$$

The optimization problem can be solved by Singular Value Decomposition (SVD) applied on the matrix X, each column of which is $x_i - \bar{x}$, such that $X = ASB^T$, where A is an $n \times m$ matrix satisfying $A^TA = I$, B is a $m \times m$ matrix satisfying $B^TB = I$ and S is a $m \times m$ diagonal matrix with diagonal elements $s_{11} \geqslant s_{22} \geqslant s_{33} \cdots s_{mm}$ (they are singular values of X). The first k columns of A constitute the matrix V in (1). The columns of V are called the principal components of X ([5]).

Note that there are both positive and negative elements in each of the principal components and also both positive and negative coefficients in linear combinations (i.e., λ_i, $i = 1, 2 \cdots, m$, has mixed signs). However the mixed signs contradict our experience and make it hard to explain the results. For example, the pixels in an image should be non-negative, hence the principal components with negative elements extracted from the images cannot be intuitively interpreted ([6]). In fact, in many applications such as image processing, biology or text mining, nonnegative data analysis is often important and nonnegative constraints on the wanted principal components (basis matrix) and coefficients (coding matrix) can improve the interpretability of the results. NMF is thus proposed to address this problem. In particular, NMF aims to find the nonnegative basic representative factors which can be used for feature extraction, dimensional reduction, eliminating redundant information and discovering the hidden patterns behind a series of non-negative vectors.

Recent years, NMF has attracted considerable interests from research community. Various extensions are proposed to address the emerging challenges and have been successfully applied to the field of unsupervised learning in data mining including environmetrics ([3]), image processing ([1]) chemometrics ([7]), pattern recognition ([8]), multimedia data analysis ([9]), text mining ([10–13]) and bioinformatics ([14–16]), etc., and received lots of attention. In [17] it has been shown that when the least squares error is selected as the cost function, NMF is equivalent to the soft K-means model, which establishes the theoretical foundation of NMF used for data clustering. Besides the traditional least squares error (Frobenius norm), there are other divergence functions that can be used as the cost functions for NMF, such as K-L divergence and chi-square statistic ([2, 18]). In [18] it has been shown that constrained NMF using with K-L divergence is equivalent to Probabilistic Latent Semantic Indexing, another unsupervised learning model popularly used in text analysis ([19, 18]).

In this chapter, we give a systematic survey of Nonnegative Matrix Factorization, including the basic model, and its variations and extensions, the applications of NMF in text mining, image processing, bioinformatics, finance etc., and the relations with K-means and PLSI. This chapter will not cover all of the related works on NMF, but will try to address the most important ones that we are interested in.

The chapter is organized as follows: Section 2 gives the standard NMF model and several variations, Section 3 summarizes the divergence functions used in the standard NMF model and the algorithms employed for solving the model, Section 4 reviews some selected applications of NMF, Section 5 gives the theoretical analysis concerning the relations between NMF and the other two unsupervised learning models including K-means and Probabilistic Latent Semantic Indexing (PLSI), and Section 6 concludes.

2 Standard NMF and Variations

To easy explanation, Table 1 lists the notations used throughout the chapter.

Table 1. Notations used in this chapter.

a_i	Column vector indexed by i;
A	Matrix;
A_{ij}	Element of the ith row and the jth column in matrix A;
$A_{:,j}$	The jth column of matrix A;
$A_{i,:}$	The ith row of matrix A;
$A \geqslant 0$	A is element-wise nonnegative, i.e., $A_{ij} \geqslant 0$ for all i and j;
A_+	Matrix A that satisfies $A \geqslant 0$;
$A_\pm$	Matrix A that has mixed signs, i.e., there is no restriction on the elements' signs of A;
$\frac{A.}{B}$	Matrix whose $(i,j)-th$ element is $\frac{A_{ij}}{B_{ij}}$;
$A^{(t)}$	The updated matrix A at the end of $t-$th iteration in the algorithm;
$A^{(t)}_{ij}$	The $(i,j)-th$ element of matrix $A^{(t)}$.

2.1 Standard NMF

Nonnegative Matrix Factorization (NMF) is one of the models that focus on the analysis of non-negative data matrices which are often originated from text mining, images processing and biology. Mathematically, NMF can be described as follows: given an $n \times m$ matrix X composed of non-negative elements, the task is to factorize X into a non-negative matrix F of size $n \times r$ and another non-negative matrix G of size $m \times r$ such that $X \approx FG^T$. r is preassigned and should satisfy $r \ll m, n$. It is usually formulated as an optimization:

$$\begin{aligned} \min_{F,G} \quad & J(X\|FG^T) \\ s.t. \quad & F \geqslant 0, G \geqslant 0, \end{aligned} \tag{2}$$

where $J(X\|FG^T)$ is some divergence function that measures dissimilarity between X and FG^T, and will be discussed in Sect. 3. Meanings of F and G can be explained variously in different fields or for different purposes and will be discussed in Sect. 4.

As we can see, all the elements of F and G are variables that need to be decided, hence this is a large scale optimization problem and the standard algorithms are not suitable, and one can observe that $J(X\|FG^T)$ is individually convex in F and in G, hence in general, most of the algorithms designed for NMF are iteratively and alternatively minimizing or decreasing F and G, which is summarized in Algorithm 1. The details will be discussed in Sect. 3.

Algorithm 1 Nonnegative Matrix Factorization (General Case)

```
Input: $F^{(0)}, G^{(0)}, t = 1$.
Output: $F, G$.
while 1 do
    Fix $G^{(t-1)}$ and find $F^{(t)}$ such that $J(X\|F^{(t)}G^{(t-1)T}) \leqslant J(X\|F^{(t-1)}G^{(t-1)T})$;
    Fix $F^{((t))}$ and find $G^{(t)}$ such that $J(X\|F^{(t)}G^{(t)T}) \leqslant J(X\|F^{(t)}G^{(t-1)T})$;
    Test for convergence;
    if Some convergence condition is satisfied then
        $F = F^{(t)}$;
        $G = G^{(t)}$;
        Break
    end if
    $t = t + 1$;
end while
```

At last, we give an important property of NMF ([20, 21]) to close this subsection. As we have mentioned above, the factors in Singular Value Decomposition (SVD): $X = ASB^T = A'B'^T$, where $A' = AS^{1/2}$ and $B' = BS^{1/2}$, typically contain mixed sign elements. And NMF differs from SVD due to the absence of cancellation of plus and minus signs. But what is the fundamental signature of this absence of cancellation? It is the *Boundedness Property*.

Theorem 1. *(Boundedness Property, [20, 21]) Let $0 \leqslant X \leqslant M$*[1]*, where M is some positive constant, be the input data matrix. F, G are the nonnegative matrices satisfying*

$$X = FG^T. \tag{3}$$

There exists a diagonal matrix $D \geq 0$ such that

$$X = FG^T = (FD)(GD^{-1})^T = F^* G^{*T} \tag{4}$$

with

$$0 \leqslant F^*_{ij} \leqslant \sqrt{M},\ 0 \leqslant G^*_{ij} \leqslant \sqrt{M}. \tag{5}$$

If X is symmetric and $F = G^T$, then $G^* = G$.

Proof. See Appendix.

[1] $0 \leqslant X \leqslant M$ means $0 \leq X_{ij} \leqslant M$, $i = 1, 2, \cdots, n$, $j = 1, 2, \cdots, m$.

We note that SVD decomposition does not have the boundedness property. In this case, even if the input data are in the range of $0 \leq X_{ij} \leq M$, we can find some elements of A' and B' that are larger than $\sqrt{M}$.

In NMF, there is a scale flexibility, i.e., for any positive D, if (F, G) is a solution, so is (FD, GD^{-1}). This theorem assures the existence of an appropriate scale such that both F and G are bounded, i.e., their elements can not exceed the magnitude of the input data matrix. This ensures that F, G are in the same scale.

Consequently, we will briefly review the variations that are rooted from NMF and proposed from different perspectives. Note that only the motivations for the research and the model formulations are reviewed, their algorithms and the application results are omitted here due to space limitation. One can find more details in the corresponding references.

2.2 Semi-NMF ([22])

Semi-NMF is designed for the data matrix X that has mixed signs. In semi-NMF, G is restricted to be nonnegative while the other factor matrix F can have mixed signs, i.e., semi-NMF can take the following form[2]: $X_{\pm} \approx F_{\pm} G_{+}^{T}$. This model is motivated from the perspective of data clustering. When clustering the columns of data matrix X, the columns of F can be seen as the cluster centroids and the rows of G denote the cluster indicators, i.e., the column j of X belongs to cluster k if $k = \arg\max_{p}\{G_{jp}\}$. Hence the nonnegative constraint on F can be relaxed such that the approximation FG^{T} is tighter and the results are more interpretable. Naturally, semi-NMF can also take the form: $X_{\pm} \approx F_{+} G_{\pm}^{T}$ if we want to cluster the rows of matrix X.

2.3 Convex-NMF ([22])

Convex-NMF is also presented for reasons of interpretability. Since the factor F denotes the cluster centroids, the columns of F should lie within the column space of X, i.e., $F_{:,j},\ j = 1, 2, \cdots, r$, can be represented as the convex combination of the columns of X: $F_{:,j} = \sum_{i=1}^{m} W_{ij} X_{:,i}$ or $F = XW$ with constraints $W \geqslant 0$ and $\sum_{i=1}^{m} W_{ij} = 1,\ j = 1, 2, \cdots, r$. Hence the model can take the following form: $X_{\pm} \approx X_{\pm} W_{+} G_{+}^{T}$. An interesting conclusion is that the convex-NMF factors F and G are naturally sparse.

2.4 Tri-NMF ([23])

Tri-NMF is presented to address the co-clustering problem (See Sect. 4.4), i.e., it presents a framework for clustering the rows and columns of the objective matrix X simultaneously. This model aims to find three factors F, S and G such that $X_{+} \approx F_{+} S_{+} G_{+}^{T}$ with constraints $F^{T}F = I$ and $G^{T}G = I$. F and G are the

[2] The subscripts ± and + are used frequently to indicate the application scopes of the models.

membership indicator matrices of the rows and the columns of X respectively, and S is an additional degree of freedom which makes the approximation tighter.

2.5 Kernel NMF ([24])

For the element-wise mapping ϕ: $X_\pm \mapsto \phi(X_\pm) : \phi(X)_{ij} = \phi(X_{ij})$, kernel NMF is designed as: $\phi(X_\pm) \approx \phi(X_\pm)W_+G_+^T$, from which one can see that the kernel NMF is just an extension of the convex-NMF. Kernel-NMF is well-defined since $\|\phi(X) - \phi(X)WG^T\|^2 = \text{trace}(\phi^T(X)\phi(X) - 2\phi^T(X)\phi(X)WG^T + GW^T\phi^T(X)\phi(X)WG^T)$ only depends on the kernel $K = \phi^T(X)\phi(X)$. Note that the standard NMF or Semi-NMF does not have the kernel extension on $\phi(X)$ since, in that case, F and G will depend explicitly on the mapping $\phi(\cdot)$ which is unknown.

2.6 Local Nonnegative Matrix Factorization, LNMF ([25, 26])

As we have mentioned above, NMF is presented as a “part of whole” factorization model and tries to mine localized part-based representation that can help to reveal low dimensional and more intuitive structures of observations. But it has been shown that NMF may give holistic representation instead of part-based representation ([25, 27]). Hence many efforts have been done to improve the sparseness of NMF in order to identify more localized features that are building parts for the whole representation. Here we introduce several sparse variants of NMF, including LNMF, NNSC, SNMF, NMFSC, nsNMF, SNMF/R and SNMF/L, as the representative results on this aspect.

LNMF was presented by [25]. In simple terms, it imposes the sparseness constraints on G and locally constraints on F based on the following three considerations:

- Maximizing the sparseness in G;
- Maximizing the expressiveness of F;
- Maximizing the column orthogonality of F.

The objective function in the model of LNMF can take the following form: $\sum_{i,j}(X_{ij} \log \dfrac{X_{ij}}{(FG^T)_{ij}} - X_{ij} + (FG^T)_{ij}) + \alpha \sum\limits_{i,j}(F^TF)_{ij} - \beta \sum\limits_{i}(G^TG)_{ii}$.

2.7 Nonnegative Sparse Coding, NNSC ([28])

NNSC only maximizes the sparseness in G. The objective function to be minimized can be written as: $\|X - FG^T\|_F^2 + \lambda \sum_{i,j} G_{ij}$.

2.8 Spares Nonnegative Matrix Factorization, SNMF ([29–31])

The objective function in the above model of NNSC can be separated into a least squares error term $\|X - FG^T\|_F^2$ and an additional penalty term $\sum_{i,j} G_{ij}$.

Ref [29] replaced the least squares error term with the KL divergence to get the following new objective function: $\sum_{i,j}[X_{ij}\log\frac{X_{ij}}{(FG^T)_{ij}} - X_{ij} + (FG^T)_{ij}] + \lambda\sum\limits_{i,j}G_{ij}$. Similarly, ref. [30] revised the penalty term to get another objective function:

$$\|X - FG^T\|_F^2 + \lambda\sum_{i,j}G_{ij}^2. \tag{6}$$

Furthermore, ref. [31] added an additional constraint on F, similar to that on G, into the objective function (6) to give the following CNMF model:

$$\min_{F\geqslant 0, G\geqslant 0}\|X - FG^T\|_F^2 + \alpha\sum_{i,j}F_{ij}^2 + \beta\sum_{i,j}G_{ij}^2.$$

2.9 Nonnegative Matrix Factorization with Sparseness Constraints, NMFSC ([32])

NMFSC employs the following measure to control the sparseness of F and G directly:

$$\mathrm{Sp}(a) = \frac{\sqrt{n} - \sum|a_j|/\sqrt{\sum a_j^2}}{\sqrt{n-1}}.$$

In other words, the model can be written as:

$$\begin{aligned} \min\ & \|X - FG^T\|_F^2 \\ s.t.\ & \mathrm{Sp}(F_{:,j}) = S_F, \\ & \mathrm{Sp}(G_{:,j}) = S_G, j = 1, 2, \cdots, r, \end{aligned}$$

where S_F and S_G are constants in [0,1], and it is easy to verify that the larger S_F and S_G, the more sparse F and G are.

2.10 Nonsmooth Nonnegative Matrix Factorization, nsNMF ([15])

nsNMF is also motivated by sparseness requirement of many applications and can be formulated as: $X = FSG^T$, where $S = (1-\theta)I + \frac{\theta}{k}II^T$ is a "smoothing" matrix, I is identity matrix and the parameter $\theta \in [0, 1]$ can indirectly control the sparseness of both the basis matrix F and the coding matrix G. One can observe that the larger the parameter θ, the more smooth (non-sparse) FS and GS are, in other words, each column of FS tends to be the constant vector with values equal to the average of the corresponding column of F as $\theta \to 1$. This is also the case for GS. But when updating G while fixing FS, the smoothness in FS will naturally enforce the sparseness in G and when updating F while fixing GS, the smoothness in GS will also enforce the sparseness in F. Hence F and G are enforced to be sparse iteratively. Note that $\theta = 0$ corresponds to the standard NMF.

2.11 Sparse NMFs: SNMF/R, SNMF/L ([33])

Sparse NMFs includes two formulations: SNMF/R for sparse G and SNMF/L for sparse F. SNMF/R is formulated as: $\min\limits_{F\geqslant 0, G\geqslant 0} \|X - FG^T\|_F^2 + \eta \sum_{i,j} F_{ij}^2 + \beta \sum_i (\sum_j G_{ij})^2$ and SNMF/L is formulated as: $\min\limits_{F\geqslant 0, G\geqslant 0} \|X - FG^T\|_F^2 + \eta \sum_{i,j} G_{ij}^2 + \alpha \sum_i (\sum_j F_{ij})^2$.

We note that there is still lack of systematic comparisons of the concordances and differences among the above seven sparse variants of NMF, which is an interesting topic.

2.12 CUR Decomposition ([34])

Instead of imposing the sparseness constraints on F and G, CUR decomposition constructs F from selected columns of X and G from selected rows of X respectively. In other words, the columns of F are composed of a small number of the columns in X and the columns of G are composed of a small number of the rows in X. The model can be formulated as follows: $X \approx FSG^T$[3], where S is introduced to make the approximation tighter, as its counterpart has done in Tri-NMF.

2.13 Binary Matrix Factorization, BMF ([20, 21])

Binary Matrix Factorization (BMF) wants to factorize a binary matrix X (that is, elements of X are either 1 or 0) into two binary matrices F and G (thus conserving the most important integer property of the objective matrix X) satisfying $X \approx FG^T$. It has been shown that the bi-clustering problem (See Sect. 4.4) can be formulated as a BMF model ([21]). Unlike the greedy strategy-based models/algorithms, BMF are more likely to find the global optima. Experimental results on synthetic and real datasets demonstrate the advantages of BMF over existing bi-clustering methods. BMF will be further discussed in Sect. 4.

Table 2 summarizes the variations and extensions of NMF mentioned above.

3 Divergence Functions and Algorithms for NMF

In this part, we will review the divergence functions used for NMF and the algorithms employed for solving the model. We will consider several important divergence functions and the algorithmic extensions of NMF developed to accommodate these functions.

[3] In the original research, this model was presented as: $A \approx CUR$, which is the origin of the name CUR, and A may have mixed signs.

Table 2. Summary of different models based on NMF. Each row lists a variant and its associated constraints. ± means that the matrix in the corresponding column may have mixed signs, + means that the matrix is nonnegative, $0-1$ means that the elements in the matrix can only be zero or one and I denotes identity matrix.

Models	Cost Function	X	F	S	G
NMF1 [1, 2]	Least Squares Error	+	+	I	+
NMF2 [1, 2]	K-L Divergence	+	+	I	+
Semi-NMF [22]	Least Squares Error	±	±	I	+
Convex-NMF [22]	Least Squares Error	±	±, $F_{:,j}$ is the convex combination of $\{X_{:,j}; j = 1, \cdots, n\}$	I	+
Tri-NMF [23]	Least Squares Error	+	+, $F^TF = I$	+	+, $G^TG = I$
Symmetric-NMF [24]	Least Squares Error	+, symmetric	+, $F = G$	+	+
K-means [17]	Least Squares Error	+, symmetric	+, $F = G$	I	+, $G^TG = I$
PLSI[a] [18, 19]	K-L Divergence	$\sum_{i,j} X_{ij} = 1$	$\sum_i F_{ik} = 1, i = 1, \cdots m$	Diagonal, $\sum_k S_{kk} = 1$	$\sum_j G_{jk} = 1, j = 1, \cdots n$
LNMF [25, 26]	K-L Divergence with penalty terms[b]	+	+	I	+
NNSC [28]	Least Squares Error with penalty terms[c]	+	+	I	+
SNMF1 [29]	K-L Divergence with penalty terms[d]	+	+	I	+
SNMF2 [30]	Least Squares Error with penalty terms[e]	+	+	I	+
SNMF3 [31]	Least Squares Error with penalty terms[f]	+	+	I	+
NMFSC [32]	Least Squares Error	+	+, $\mathrm{Sp}(F_{:,j}) = S_F$[g], $j = 1, \cdots, k$	I	+, $\mathrm{Sp}(G_{:,j}) = S_G$[h], $j = 1, \cdots, k$
NMF/L [33]	Least Squares Error with penalty terms[i]	+	+	I	+
NMF/R [33]	Least Squares Error with penalty terms[j]	+	+	I	+
nsNMF [15]	K-L Divergence	+	+	$S = (1-\theta)I + \frac{\theta}{k}II^T$	+
CUR [34]	Least Squares Error	+	+[k]	+	+[l]
BMF [20, 21]	Least Squares Error	$0-1$	$0-1$	I	$0-1$

[a] The relations between NMF and K-means, between NMF and PLSI will be reviewed in Sect. 5.

[b] $\sum_{i,j}(X_{ij} \log \frac{X_{ij}}{(FG^T)_{ij}} - X_{ij} + (FG^T)_{ij}) + \alpha \sum_{i,j}(F^TF)_{ij} - \beta \sum_i (G^TG)_{ii}$.

[c] $\|X - FG^T\|_F^2 + \lambda \sum_{i,j} G_{ij}$.

[d] $\sum_{i,j}[X_{ij} \log \dfrac{X_{ij}}{(FG^T)_{ij}} - X_{ij} + (FG^T)_{ij}] + \lambda \sum\limits_{i,j} G_{ij}$.

[e] $\|X - FG^T\|_F^2 + \lambda \sum_{i,j} G_{ij}^2$.

[f] $\|X - FG^T\|_F^2 + \alpha \sum_{i,j} F_{ij}^2 + \beta \sum_{i,j} G_{ij}^2$.

[g] $\mathrm{Sp}(a) = (\sqrt{n} - \sum |a_j| / \sqrt{\sum a_j^2}) / \sqrt{n-1}$, S_F is a constant.

[h] S_G is a constant.

[i] $\|X - FG^T\|_F^2 + \eta \|G\|_F^2 + \beta \sum_i \|F_{i,:}\|_1^2$.

[j] $\|X - FG^T\|_F^2 + \eta \|F\|_F^2 + \beta \sum_i \|G_{i,:}\|_1^2$.

[k] Columns of F are composed of a small number of columns in X.

[l] Columns of G are composed of a small number of rows in X.

3.1 Divergence Functions

One of the main advantages of NMF is its flexibility in the selection of the objective divergence functions. Here we will review several important divergence functions and the relations among them. These functions play important roles in solving NMF model, and may lead to different numerical performance. Hence research on the relations between the divergence functions and the appropriate applications is of great interest. Detailed theoretical analysis addressing this problem is in pressing need though some related numerical results have been given.

Csiszár's φ Divergence ([35]) The Csiszár's φ divergence is defined as: $D_{\varphi}(X||FG^T) = \sum\limits_{i,j}(FG^T)_{ij}\varphi(\dfrac{X_{ij}}{(FG^T)_{ij}})$, where $X_{ij} \geqslant 0, (FG^T)_{ij} \geqslant 0$ and $\varphi : [0, \infty) \to (-\infty, \infty)$ is some convex function and continuous at point zero. Based on the flexibility of φ, the divergence has many instances. For example:

- $\varphi = (\sqrt{x} - 1)^2$ corresponds to Hellinger divergence;
- $\varphi = (x - 1)^2$ corresponds to Pearson's χ^2 divergence;
- $\varphi = x(x^{\alpha-1} - 1)/(\alpha^2 - \alpha) + (1 - x)/\alpha$ corresponds to Amari's $\alpha-$divergence, which will be introduced later.

Note that though the selection of φ is flexible, Csiszár's φ divergence does not include the traditional least squares error: $D_{\mathrm{LSE}}(X\|FG^T) = \sum_{i,j}(X_{ij} - (FG^T)_{ij})^2$.

$\alpha-$Divergence, ([36–39]) The $\alpha-$divergence is defined as: $D_{\alpha}(X||FG^T) = \dfrac{1}{\alpha(1-\alpha)}\sum\limits_{i,j}(\alpha X_{ij} + (1-\alpha)(FG^T)_{ij} - X_{ij}^{\alpha}(FG^T)_{ij}^{1-\alpha})$, where $\alpha \in (-\infty, \infty)$. Different selection of α may corresponds to different specific divergence. For example:

- $\lim\limits_{\alpha\to 0} D_{\alpha}(X||FG^T)$ corresponds to K-L divergence $D_{\mathrm{KL}}(FG^T||X)$;
- $\alpha = \dfrac{1}{2}$ corresponds to Hellinger divergence;
- $\lim\limits_{\alpha\to 1} D_{\alpha}(X||FG^T)$ corresponds to K-L divergence $D_{\mathrm{KL}}(X||FG^T)$;
- $\alpha = 2$ corresponds to Pearson's χ^2 divergence.

Since $\alpha-$divergence is a special case of Csiszár's φ divergence, as we have mentioned above, it does not include the least squares error either.

Bregman Divergence ([40]) The Bregman divergence can be defined as: $D_{\mathrm{Breg}}(X\|FG^T) = \sum_{i,j}\varphi(X_{ij}) - \varphi((FG^T)_{ij}) - \varphi'((FG^T)_{ij})(X_{ij} - (FG^T)_{ij})$, where $\varphi : S \subseteq \mathbb{R} \to \mathbb{R}$ is some strictly convex function that has continuous first derivative, and $(FG^T)_{ij} \in \mathrm{int}(S)$ (the interior of set S). Some instances of Bregman divergence are listed as follows:

- $\varphi = \frac{x^2}{2}$ corresponds to least squares error;
- $\varphi = x \log x$ corresponds to K-L divergence;
- $\varphi = -\log x$ corresponds to Itakura-Saito (IS) divergence.

$\beta-$Divergence ([41, 39]) The $\beta-$divergence is defined as: $D_\beta(X||FG^T) = \sum_{i,j}(X_{ij}\frac{X_{ij}^\beta - (FG^T)_{ij}^\beta}{\beta} - \frac{X_{ij}^{\beta+1} - (FG^T)_{ij}^{\beta+1}}{\beta+1})$ where $\beta \neq 0, -1$. This divergence is also a big family including K-L divergence, least squares error, etc. Specifically:

- $\lim_{\beta \to 0} D_\beta(X||FG^T)$ corresponds to K-L divergence $D_{\mathrm{KL}}(X||FG^T)$;
- $\beta = 1$ corresponds to least squares error $D_{\mathrm{LSE}}(X||FG^T)$;
- $\lim_{\beta \to -1} D_\beta(X||FG^T)$ corresponds to Itakura-Saito (IS) divergence which will be introduced later.

Note that $\beta-$divergence $D_\beta(x\|y)$ can be got from $\alpha-$divergence $D_\alpha(x\|y)$ by nonlinear transformation: $x = x^{\beta+1}$, $y = y^{\beta+1}$ and supposing $\alpha = \frac{1}{\beta+1}$ ([42]).

Itakura-Saito (IS) Divergence ([43]) The Itakura-Saito divergence is defined as: $D_{\mathrm{IS}}(X\|FG^T) = \sum_{i,j}(\frac{X_{ij}}{(FG^T)_{ij}} - \log \frac{X_{ij}}{(FG^T)_{ij}} - 1)$. Note that IS divergence is a special case of both the Bregman divergence ($\phi(x) = -\log x$) and the β-divergence ($\beta = -1$).

K-L Divergence ([2]) The K-L divergence is defined as: $D_{\mathrm{KL}}(X\|FG^T) = \sum_{i,j}[X_{ij} \log \frac{X_{ij}}{(FG^T)_{ij}} - X_{ij} + (FG^T)_{ij}]$. As we have discussed above, the K-L divergence is a special case of $\alpha-$divergence, Bregman divergence and $\beta-$divergence.

Least Squares Error ([2]) The least squares error is defined as: $D_{\mathrm{LSE}}(X\|FG^T) = \|X - FG^T\|_F^2 = \sum_{i,j}(X_{ij} - (FG^T)_{ij})^2$, which is a special case of Bregman divergence and $\beta-$divergence.

We summarize the different divergence functions and the corresponding multiplicative update rules (See Sect. 3.2) in Table 3. The other algorithms such as Newton algorithm or Quasi-Newton algorithm that are specially designed for some of the divergence functions will be reviewed in the next subsection.

3.2 Algorithms for NMF

The algorithm design for solving NMF is an important direction and several algorithms, according to different objective divergence functions and different

Table 3. Summary of the different divergence functions and the corresponding multiplicative update rules. Note that "Convergence" only says whether the update rules have been proven to be monotonically decreasing. Even if this is proven, the algorithm does not necessarily converge to a local minimum ([44]).

Divergence Function	Multiplicative Update Rules of F and G	Convergence	Comments
Csiszár's φ Divergence	——	——	——
$\alpha-$Divergence	$F_{ik} := F_{ik}(\frac{((\frac{X.}{FG^T})^\alpha G)_{ik}}{\sum\limits_l G_{lk}})^{\frac{1}{\alpha}}$ $G_{ik} := G_{ik}(\frac{((\frac{X^T.}{GF^T})^\alpha F)_{ik}}{\sum\limits_l F_{lk}})^{\frac{1}{\alpha}}$	proved	special case of $\varphi-$divergence $(\varphi(x) = x(x^{\alpha-1} - 1)/(\alpha^2 - \alpha) + (1 - x)/\alpha)$
Bregman Divergence	$F_{ik} := F_{ik} \frac{\sum_j \nabla^2 \phi(FG^T)_{ij} X_{ij} G_{jk}}{\sum_j \nabla^2 \phi(FG^T)_{ij} (FG^T)_{ij} G_{jk}}$ $G_{ik} := G_{ik} \frac{\sum_j \nabla^2 \phi(GF^T)_{ij} X_{ji} F_{jk}}{\sum_j \nabla^2 \phi(GF^T)_{ij} (GF^T)_{ij} F_{jk}}$	proved	——
$\beta-$Divergence	$F_{ik} := F_{ik} \frac{\sum_j ((FG^T)_{ij}^{\beta-1} X_{ij}) G_{jk}}{\sum_j (FG^T)_{ij}^{\beta} G_{jk}}$ $G_{ik} := G_{ik} \frac{\sum_j ((FG^T)_{ji}^{\beta-1} X_{ji}) F_{jk}}{\sum_j (FG^T)_{ji}^{\beta} F_{jk}}$	proved when $0 \leqslant \beta \leqslant 1$ ([43])	——
Itakura-Saito (IS) Divergence	$F_{ik} := F_{ik} \frac{\sum_j \frac{X_{ij}}{(FG^T)_{ij}^2} G_{jk}}{\sum_j \frac{G_{jk}}{(FG^T)_{ij}}}$ $G_{ik} := G_{ik} \frac{\sum_j \frac{X_{ji}}{(FG^T)_{ji}^2} F_{jk}}{\sum_j \frac{F_{jk}}{(FG^T)_{ji}}}$	not proved	special case of Bregman divergence $(\varphi(x) = -\log x)$ and $\beta-$divergence $(\beta = -1)$
K-L Divergence	$F_{ik} := \frac{F_{ik}}{\sum_j G_{jk}} \sum_j \frac{X_{ij}}{(FG^T)_{ij}} G_{jk}$ $G_{ik} := \frac{G_{ik}}{\sum_j F_{jk}} \sum_j \frac{X_{ji}}{(FG^T)_{ji}} F_{jk}$	proved	special case of $\alpha-$divergence $(\alpha = 1)$, Bregman divergence $(\varphi(x) = x \log x)$ and $\beta-$divergence $(\beta = 0)$
Least Squares Error	$F_{ik} := F_{ik} \frac{(XG)_{ik}}{(FG^T G)_{ik}}$ $G_{ik} := G_{ik} \frac{(X^T F)_{ik}}{(GF^T F)_{ik}}$	proved	special case of Bregman divergence $(\varphi(x) = \frac{x^2}{2})$ and $\beta-$divergence $(\beta = 1)$

application purposes, have been proposed. In this part, we will briefly review the representative ones. Note that to simplify the complexity of the problem, we only consider the standard NMF model, i.e., only the optimization problem (2) is considered. The algorithms for its variations presented in Sect. 2 can be obtained by simple derivations and can be found in the corresponding literature.

Multiplicative Update Algorithm ([1, 2]) The multiplicative update rules of NMF with its convergence proof (indeed, only the monotonic decreasing property is proved) was firstly presented by Lee & Seung ([1, 2]). Because of the simplicity and effectiveness, it has become one of the most influential algorithms that are widely used in the data mining community. This algorithm is gradient-descent-based and similar to the Expectation Maximization Algorithm (EM). Specifically when the K-L divergence is selected as the objective function, the multiplicative update algorithms can be summarized as Algorithm 2. In addition, there are several interesting properties of the relations between the multiplicative update rules with K-L divergence and the EM algorithm employed in Probabilistic Latent Semantic Indexing (PLSI), which will be discussed in Sect. 5.

The update rules in line 2 and line 3 of Algorithm 2 vary with the user-selected objective functions and have been summarized in Table 3.

Algorithm 2 Nonnegative Matrix Factorization (K-L divergence, Multiplicative Update Rules)

```
Input: $F^{(0)}, G^{(0)}, t = 1$.
Output: $F, G$.
 1: while 1 do
 2:     Update $F_{ik}^{(t)} := \frac{F_{ik}^{(t-1)}}{\sum_j G_{jk}^{(t-1)}} \sum_j \frac{X_{ij}}{(F^{(t-1)}G^{(t-1)T})_{ij}} G_{jk}^{(t-1)}$;
 3:     Update $G_{jk}^{(t)} := \frac{G_{jk}^{(t-1)}}{\sum_i F_{ik}^{(t)}} \sum_i \frac{X_{ij}}{(F^{(t)}G^{(t-1)T})_{ij}} F_{ik}^{(t)}$;
 4:     Test for convergence;
 5:     if Some convergence condition is satisfied then
 6:         $F = F^{(t)}$;
 7:         $G = G^{(t)}$;
 8:         Break
 9:     end if
10:     $t = t + 1$;
11: end while
```

Project Gradient Algorithm ([45]) The project gradient descent method is generally designed for bound-constrained optimization problems. In order to use this method, a sufficiently large upper bound U is firstly set for F and G (since the upper bound U is sufficiently large, the solutions of the revised model will be identical with the original one). The objective optimization function is selected as the least squares error. The K-L divergence is not suitable because this divergence is not well-defined on the boundary of the constraints (the log function is defined for positive reals). The method can then be summarized in Algorithm 3. Note that $(P[\bullet])_{ij} = \begin{cases} \bullet_{ij}, & 0 \leqslant \bullet_{ij} \leqslant U, \\ 0. & \bullet_{ij} < 0, \\ U, & \bullet_{ij} > U. \end{cases}$

Newton Algorithm ([46]) The Newton algorithm is designed for the least squares error (Indeed, the idea of quasi-Newton method is employed). Basically, it can be summarized in Algorithm 4. Note that D is an appropriate positive definite gradient scaling matrix, and $[Z_+(X)]_{ij} = \begin{cases} X_{ij}, & (i,j) \notin I_+, \\ 0. & \text{otherwise} \end{cases}$ and I_+ will be given in the algorithm. The details are omitted due to space limitation.

The Newton algorithm and the Quasi-Newton algorithm presented below have utilized the second order information of the model (Hessian matrix), hence one can expect that they have better numerical performance than the multiplicative update rules and the projected gradient descent though they should be more time-consuming.

Quasi-Newton Method ([47]) The Quasi-Newton algorithm is designed for the $\alpha-$divergence. As we have discussed above, this divergence is a general case of several useful objective optimization functions including the K-L divergence. But note that the least squares error is not included. The proposed Quasi-Newton algorithm is summarized in Algorithm 5. Note that $H_J^{(F)}$ and $H_J^{(G)}$ are the Hessian matrices of F and G, and $\nabla_F J$ and $\nabla_G J$ are the gradients of F and G.

Active Set Algorithm ([48]) The active set algorithm is designed for the least squares error. The basic idea is to decompose the original optimization problem $\min\limits_{F \geqslant 0, G \geqslant 0} \|X - FG^T\|_F^2$ into several separate subproblems, then solve them independently using the standard active set method and finally merge the solutions obtained. In other words, firstly, fixing F, decompose the problem $\min\limits_{F \geqslant 0, G \geqslant 0} \|X - FG^T\|_F^2$ into the following series of subproblems: $\min\limits_{G_{i,:} \geqslant 0} \|X_{:,i} - FG_{i,:}^T\|_F^2$, $i = 1, 2, \cdots, m$, then solve them independently and finally update G. Then fixing G, update F similarly.

Hereto, we have reviewed several newly developed algorithms, most of which are nonlinear-programming-originated but are specially designed for NMF model. Note that the technical details are omitted here due to space limitation. One can get more information from the corresponding references.

Algorithm 3 Nonnegative Matrix Factorization (Least Squares Error, Projected Gradient Method)

```
Input: $F^{(0)}, G^{(0)}, t = 1$.
Output: $F, G$.
while 1 do
  $F^{(old)} = F^{(t-1)}$;
  while 1 do
    Compute the gradient matrix $\nabla_F J(X, F^{(old)} G^{(t-1)T})$;
    Compute the step length $\alpha$;
    Update $F^{(old)}$:
      $F^{(new)} = P[F^{(old)} - \alpha \nabla_F J(X, F^{(old)} G^{(t-1)T})]$;
    $F^{(old)} = F^{(new)}$;
    Test for convergence;
    if Some convergence condition is satisfied then
      $F^{(t)} = F^{(old)}$;
      Break
    end if
  end while
  $G^{(old)} = G^{(t-1)}$;
  while 1 do
    Compute the gradient matrix $\nabla_G J(X, F^{(t)} G^{(old)T})$;
    Compute the step length $\alpha$;
    Update $G^{(old)}$:
      $G^{(new)} = P[G^{(old)} - \alpha \nabla_G J(X, F^{(t)} G^{(old)T})]$;
    $G^{(old)} = G^{(new)}$;
    Test for convergence;
    if Some convergence condition is satisfied then
      $G^{(t)} = G^{(old)}$;
      Break
    end if
  end while
  if Some stopping criteria are met then
    $F = F^{(t)}$; $G = G^{(t)}$;
    Break
  end if
  $t = t + 1$;
end while
```

Algorithm 4 Nonnegative Matrix Factorization (Least Squares Error, Newton Algorithm)

```
Input: $F^{(0)}, G^{(0)}, D, t = 1$.
Output: $F, G$.
while 1 do
    $F^{(old)} = F^{(t-1)}$;
    while 1 do
        Compute the gradient matrix $\nabla_F J(X, F^{(old)} G^{(t-1)T})$;
        Compute fixed set $I_+ := \{(i,j) : F_{ij}^{(old)} = 0, [\nabla_F J(X, F^{(old)} G^{(t-1)T})]_{ij} > 0\}$
        for $F^{(old)}$;
        Compute the step length vector $\alpha$;
        Update $F^{(old)}$:
            $U = Z_+[\nabla_F J(X, F^{(old)} G^{(t-1)T})]; \;\; U = Z_+(DU);$
            $F^{(new)} = \max(F^{(old)} - U diag(\alpha), 0);$
        $F^{(old)} = F^{(new)}$;
        Update $D$ if necessary;
        Test for convergence;
        if Some convergence condition is satisfied then
            $F^{(t)} = F^{(old)}$;
            Break
        end if
    end while
    $G^{(old)} = G^{(t-1)}$;
    while 1 do
        Compute the gradient matrix $\nabla_G J(X, F^{(t)} G^{(old)T})$;
        Compute fixed set $I_+ := \{(i,j) : G_{ij}^{(old)} = 0, [\nabla_G J(X, F^{(t)} G^{(old)T})]_{ij} > 0\}$ for
        $G^{(old)}$;
        Compute the step length vector $\alpha$;
        Update $G^{(old)}$:
            $U = Z_+[\nabla_G J(X, F^{(t)} G^{(old)T})]; \;\; U = Z_+(DU);$
            $G^{(new)} = \max(G^{(old)} - U diag(\alpha), 0);$
        $G^{(old)} = G^{(new)}$;
        Update $D$ if necessary;
        Test for convergence;
        if Some convergence condition is satisfied then
            $G^{(t)} = G^{(old)}$;
            Break
        end if
    end while
    if Some stopping criteria are met then
        $F = F^{(t)}$; $G = G^{(t)}$;
        Break
    end if
    $t = t + 1$;
end while
```

Algorithm 5 Nonnegative Matrix Factorization ($\alpha-$Divergence, Quasi-Newton Algorithm)

```
Input: $F^{(0)}, G^{(0)}, t = 1$.
Output: $F, G$.
while 1 do
    Update $F^{(t)} := \max(F^{(t-1)} - [H_J^{(F)}]^{-1}\nabla_F J, 0)$;
    Update $G^{(t)} := \max(G^{(t-1)} - [H_J^{(G)}]^{-1}\nabla_G J, 0)$;
    Test for convergence;
    if Some convergence condition is satisfied then
        $F = F^{(t)}$;
        $G = G^{(t)}$;
        Break
    end if
    $t = t + 1$;
end while
```

4 Applications of NMF

Nonnegative Matrix Factorization has been proved to be valuable in many fields of data mining, especially in unsupervised learning. In this part, we will briefly review its applications in image processing, data clustering, semi-supervised clustering, bi-clustering (co-clustering) and financial data mining. Note that we cannot cover all the interesting applications of NMF, but generally speaking, the special point on NMF is its ability to recover the hidden patterns or trends behind the observed data automatically, which makes it suitable for image processing, feature extraction, dimensional reduction and unsupervised learning. The preliminary theoretical analysis concerning this ability will be reviewed in the next section, in other words, the relations between NMF and some other unsupervised learning models will be discussed.

4.1 Image Processing

Though the history of Nonnegative Matrix Factorization was traced back to 1970's, NMF was attracted lots of attention due to the research of Lee & Seung ([1, 2]). In their works, the model was applied to image processing successfully. Hence we review the applications of NMF on this aspect firstly.

In image processing, the data can be represented as $n \times m$ nonnegative matrix X, each column of which is an image described by n nonnegative pixel values. Then NMF model can find two factor matrices F and G such that $X \approx FG^T$. F is the so-called basis matrix since each column can be regarded as a part of the whole such as nose, ear or eye, etc. for facial image data. G is the coding matrix and each row is the weights by which the corresponding image can be reconstructed as the linear combination of the columns of F.

In summary, NMF can discover the common basis hidden behind the observations and the way how the images are reconstructed by the basis. Indeed, the psychological and physiological researches have shown evidence for part-based representation in the brain, which is also the foundation of some computational theories ([1]). But further researches have also shown that the standard NMF model does not necessarily give the correct part-of-whole representations ([25, 27]), hence many efforts have been done to improve the sparseness of NMF in order to identify more localized features that are building parts for the whole representation (See Sect. 2).

4.2 Clustering

One of the most interesting and successful applications of NMF is to cluster data such as text, image or biology data, i.e. discovering patterns automatically from data. Given a nonnegative $n \times m$ matrix X, each column of which is a sample and described by n features, NMF can be applied to find two factor matrices F and G such that $X \approx FG^T$, where F is $n \times r$ and G is $m \times r$, and r is the cluster number. Columns of F can be regarded as the cluster centroids while G is the cluster membership indicator matrix. In other words, the sample i is of cluster k if G_{ik} is the largest value of the row $G_{i,:}$.

The good performance of NMF in clustering has been validated in several different fields including bioinformatics (tumor sample clustering based on microarray data, [14]), community structure detection of the complex network ([49]) and text clustering ([10–12]).

4.3 Semi-supervised Clustering

In many cases, some background information concerning the pairwise relations of some samples are known and we can add them into the clustering model in order to guide the clustering process. The resulting constrained problem is called semi-supervised clustering. Specifically, the following two types of pairwise relations are often considered:

- *Must-link* specifies that two samples should have the same cluster label;
- *Cannot-link* specifies that two samples should not have the same cluster label.

Then, one can establish two nonnegative matrices $W_{\text{reward}} = \{w_{ij} : \text{ sample } i \text{ and sample } j \text{ are in the same class}\}$ and $W_{\text{penalty}} = \{w_{ij} : \text{ sample } i \text{ and sample } j \text{ are not in the same class}\}$ based on the above information, and the similarity matrix $W = X^T X$ of the samples (columns of X are samples) can then be replaced by $W - W_{\text{reward}} + W_{\text{penalty}}$ (note that it is still a symmetric matrix). Finally, NMF is applied:

$$\min_{S \geqslant 0, G \geqslant 0} \|(W - W_{\text{reward}} + W_{\text{penalty}}) - GSG^T\|_F^2,$$

where G is the cluster membership indicator, i.e., sample i is of cluster k if the element G_{ik} is the largest value of the row $G_{i,:}$. Theoretical analysis and practical applications have been contributed by [50]. We summarize the main theoretical results but omit the details here.

Theorem 2. *Orthogonal Semi-Supervised NMF clustering is equivalent to Semi-Supervised Kernel K-means ([51]).*

Theorem 3. *Orthogonal Semi-Supervised NMF clustering is equivalent to Semi-Supervised Spectral clustering with Normalized Cuts ([52]).*

4.4 Bi-clustering (co-clustering)

Bi-clustering was recently introduced by Cheng & Church ([53]) for gene expression data analysis. In practice, many genes are only active in some conditions or classes and remain silent under other cases. Such gene-class structures, which are very important to understand the pathology, can not be discovered using the traditional clustering algorithms. Hence it is very necessary to develop bi-clustering models/algorithms to identify the local structures. Bi-clustering models/algorithms are different from the traditional clustering methodologies which assign the samples into specific classes based on the genes' expression levels across *ALL* the samples, they try to cluster the rows (features) and the columns (samples) of a matrix simultaneously.

In other words, the idea of bi-clustering is to characterize each sample by a subset of genes and to define each gene in a similar way. As a consequence, bi-clustering algorithms can select the groups of genes that show similar expression behaviors in a subset of samples that belong to some specific classes such as some tumor types, thus identify the local structures of the microarray matrix data [53, 54]. Binary Matrix Factorization (BMF) has been presented for solving bi-clustering problem: the input binary gene-sample matrix X[4] is decomposed into two binary matrices F and G such that $X \approx FG^T$. The binary matrices F and G can explicitly designate the cluster memberships for genes and samples. Hence BMF offers a framework for simultaneously clustering the genes and samples.

An example is given here[5] to demonstrate the biclustering capability of BMF. Given the original data matrix

$$X = \begin{pmatrix} 0\,0\,0\,0\,0\,0\,1\,1 \\ 0\,0\,0\,0\,0\,1\,1\,0 \\ 0\,1\,1\,1\,0\,1\,1\,1 \\ 1\,0\,1\,1\,0\,1\,1\,1 \\ 0\,1\,0\,1\,0\,0\,0\,0 \end{pmatrix}.$$

One can see two biclusters, one in the upper-right corner, and one in lower-left corner. Our BMF model gives

[4] [21] has discussed the details on how to discretize the microarray data into a binary matrix

[5] Another example is given in the appendix to illustrate the limitations of NMF for discovering bi-clustering structures.

$$F = (F_{:,1}, F_{:,2}) = \begin{pmatrix} 0 & 1 \\ 0 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix}; \qquad G = (G_{:,1}, G_{:,2}) = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix};$$

The two discovered biclusters are recovered in a clean way:

$$FG^T = \begin{pmatrix} 0\,0\,0\,0\,0\,1\,1\,1 \\ 0\,0\,0\,0\,0\,1\,1\,1 \\ 0\,1\,1\,1\,0\,1\,1\,1 \\ 0\,1\,1\,1\,0\,1\,1\,1 \\ 0\,1\,1\,1\,0\,0\,0\,0 \end{pmatrix}.$$

4.5 Financial Data Mining

Underlying Trends in Stock Market : In the stock market, it has been observed that the stock price fluctuations does not behave independently of each other but are mainly dominated by several underlying and unobserved factors. Hence identification the underlying trends from the stock market data is an interesting problem, which can be solved by NMF. Given an $n \times m$ nonnegative matrix X, columns of which are the records of the stock prices during n time points, NMF can be applied to find two nonnegative factors F and G such that $X \approx FG^T$, where columns of F are the underlying components. Note that identifying the common factors that drive the prices is somewhat similar to blind source separation (BSS) in signal processing. Furthermore, G can be used to identify the cluster labels of the stocks (see Sect. 4.2) and the most interesting result is that the stocks of the same sector are not necessarily assigned into the same cluster and vice versa, which is of potential use to guide diversified portfolio, in other words, investors should diversify their money into not only different sectors, but also different clusters. More details can be found in [55].

Discriminant Features Extraction in Financial Distress Data : Building appropriate financial distress prediction model based on the extracted discriminative features is more and more important under the background of financial crisis. In [56] it has presented a new prediction model which is indeed a combination of K-means, NMF and Support Vector Machine (SVM). The basic idea is to train a SVM classifier in the reduced dimensional space which is spanned by the discriminative features extracted by NMF, the algorithm of which is initialized by K-means. The details can be found in [56].

5 Relations with Other Relevant Models

Indeed, the last ten years have witnessed the boom of Nonnegative Matrix Factorization in many fields including bioinformatics, images processing, text mining, physics, multimedia, etc. But it is still not very clear that why the model works. Researches on the relations between NMF and other unsupervised learning models such as K-means and Probabilistic Latent Semantic Indexing try to give us a preliminary interpretation of this question. The basic results of this part are: i) the model of soft K-means can be rewritten as symmetric-NMF model. Hence K-means and NMF are equivalent, which justifies the ability of NMF for data clustering. But this does not mean that K-means and NMF will generate identical cluster results since they employ different algorithms; ii) Probabilistic Latent Semantic Indexing (PLSI) and NMF optimize the same objective function (K-L divergence), but PLSI has additional constraints. The algorithms of the two models can generate equivalent solutions, but they are different in essence.

5.1 Relations between NMF and K-means

In [17] it has been shown that the model of K-means can be written in a special form of NMF with orthogonal constraints, in which the objective function is the least squares error and the objective matrix W is the similarity matrix of the original samples and symmetric. This result is important and interesting because it gives a solid foundation for NMF used for data clustering.

K-means is one of the most famous and traditional methods for clustering analysis. It aims to partition m samples into $K-$clusters. The motivation is very intuitive: the samples that are close to each other should share the same cluster indicators. Hence K-means algorithm alternatively gives the cluster index of each sample by the nearest cluster center and gives the cluster center by the centroid of its members. The major drawback of K-means is that it is very sensitive to the initializations and prone to local minima. Mathematically, K-means can be formulated as minimizing a sum of squares cost function: min $J_K = \sum\limits_{k=1}^{K} \sum\limits_{i \in C_k} \|x_i - m_k\|^2$, where $x_i, i = 1, 2, \cdots m$ are the data samples and $X = (x_1, x_2, \cdots, x_m)$ is the data matrix, $m_k = \sum\limits_{i \in C_k} x_i / n_k$ is the centroid of cluster C_k with n_k samples. This optimization problem can be equivalently solved by a special type of nonnegative matrix factorization $W = HH^T$, where $W = X^T X$, with orthogonal constraint $H^T H = I$, i.e., nonnegative matrix factorization is equivalent to soft K-means (i.e., $H^T H = I$ is relaxed).

Theorem 4. min $\|W - HH^T\|^2$, *where* $W = X^T X$, *is equivalent to soft K-means.*

Proof. See Appendix.

The model equivalence of K-means and NMF has established the theoretical foundation of NMF used for data clustering. Though NMF has been applied for clustering successfully, there is still a lack of theoretical analysis until this equivalent result is proved. But one should be noted that it does not mean that NMF and K-means generate identical results. The algorithms that used to solve NMF and K-means are quite different. NMF uses gradient descent method while

K-means uses coordinate descent method ([57]). A general conclusion is that NMF almost always outperforms K-means. Maybe this is due to the flexibility of NMF which has more parameters to be decided. In fact, K-means always wants to find the ellipsoidal-shaped clusters while NMF does not. When the data distribution is far from an ellipsoidal-shaped clustering, which is often the case for real data, NMF may have advantages ([22]). In summary, though NMF is equivalent to K-means, it often generates a different and better result.

Moreover, it has been proved that the solution of soft K-means can also be given by Principal Component Analysis (PCA), which builts closer relationships between PCA and NMF ([58]) . A systematic numerical comparison and analysis of K-means, PCA and NMF is of interesting, but is beyond the scope of this chapter.

5.2 Relations between NMF and PLSI

Probabilistic Latent Semantic Indexing (PLSI) is one of the state-of-the-art unsupervised learning models in data mining, and has been widely used in many applications such as text clustering, information retrieval and collaborative filtering. In this section, relations between NMF and PLSI, including the differences of their models and the differences of their algorithms will be given. In summary, NMF and PLSI optimize the same objective function; but their algorithms are different due to the additional constraints in PLSI.

Probabilistic Latent Semantic Indexing (PLSI, [59]) is a probabilistic model stemmed from Latent Semantic Analysis (LSA, [60]). Compared to LSA, PLSI has a more solid theoretical foundation in statistics and thus is a more principled approach for analyzing text, discovering latent topics and information retrieval, etc. ([59, 61, 62]). PLSI is a kind of topic model and, given a joint probabilistic matrix X (i.e., $\sum_{i,j} X_{ij} = 1.$), aims to get three nonnegative matrices C, diagonal S and H such that CSH^T is the approximation of X. The parameters in PLSI model are trained by the Expectation Maximization (EM) algorithm which iteratively increases the objective likelihood function until some convergence condition is satisfied and, at each step, PLSI maintains the column normalization property of C, S and H ($\sum_i C_{ik} = 1, \sum_k S_{kk} = 1, \sum_j H_{jk} = 1$).

For simplifying explanation, we take the document analysis task as an example. Given a document collection $X_{n\times m}$ of m documents and a vocabulary of n words, where each element X_{ij} indicates whether a word w_i occurs in document d_j, the learning task in PLSI is to find three matrices C,H and S, such that X is approximated by CSH^T, where C_{ik} is the probability of $P(w_i|z_k)$[6], H_{jk} is the probability of $P(d_j|z_k)$ and S is diagonal matrix with diagonal element $S_{kk} = P(z_k)$.

To learn the PLSI model, we can consider maximizing the log-likelihood of the PLSI model $L = \sum_{i,j} n(i,j)logP(w_i, d_j)$, where $n(i,j)$ is the co-occurrence number of word i and document j, and $P(w_i, d_j) = \sum_k P(w_i|z_k)P(z_k)P(d_j|z_k) =$

[6] z_k means the kth latent topic.

$\sum_k C_{ik}S_{kk}H_{jk}$. Here we normalize X to satisfy $\sum_{i,j} X_{ij} = 1$, and the log-likelihood function can then be rewritten as:

$$L = \sum_{i,j} X_{ij} \log P(w_i, d_j). \tag{7}$$

The parameters C, S and H are then iteratively got by Expectation-Maximization (EM) algorithm. The EM algorithm begins with some initial values of C, H, S and iteratively updates them according to the following formulas:

$$C_{ik} := \frac{\sum\limits_j X_{ij}P_{ij}^k}{\sum\limits_{i,j} X_{ij}P_{ij}^k}; \quad S_{kk} := \sum_{i,j} X_{ij}P_{ij}^k; \quad H_{jk} := \frac{\sum\limits_i X_{ij}P_{ij}^k}{\sum\limits_{i,j} X_{ij}P_{ij}^k}. \tag{8}$$

where P_{ij}^k is the probability of

$$P(z_k|w_i, d_j) = \frac{S_{kk}C_{ik}H_{jk}}{\sum_k S_{kk}C_{ik}H_{jk}}. \tag{9}$$

.

By combining (8) and (9), one can get:

$$\begin{aligned} C_{ik} &:= \frac{\sum\limits_j X_{ij}\frac{S_{kk}C_{ik}H_{jk}}{\sum\limits_k S_{kk}C_{ik}H_{jk}}}{\sum\limits_{i,j} X_{ij}\frac{S_{kk}C_{ik}H_{jk}}{\sum\limits_k S_{kk}C_{ik}H_{jk}}} & H_{jk} &:= \frac{\sum\limits_i X_{ij}\frac{S_{kk}C_{ik}H_{jk}}{\sum\limits_k S_{kk}C_{ik}H_{jk}}}{\sum\limits_{i,j} X_{ij}\frac{S_{kk}C_{ik}H_{jk}}{\sum\limits_k S_{kk}C_{ik}H_{jk}}} & S_{kk} &:= S_{kk}\frac{\sum\limits_{ij} X_{ij}C_{ik}H_{jk}}{\sum\limits_k S_{kk}C_{ik}H_{jk}} \\ &= C_{ik}\frac{((\frac{X.}{CSH^T})H)_{ik}}{(C^T\frac{X.}{CSH^T}H)_{kk}}; & &= H_{jk}\frac{((\frac{X.}{CSH^T})^T C)_{jk}}{(C^T\frac{X.}{CSH^T}H)_{kk}}; & &= S_{kk}(C^T\frac{X.}{CSH^T}H)_{kk}. \end{aligned} \tag{10}$$

The algorithm of PLSI is summarized in Algorithm 6:

Consequently, we will review the relations between NMF and PLSI. The basic conclusions are: 1) maximizing the objective likelihood function in PLSI is equivalent to minimizing the K-L divergence in NMF. Hence NMF and PLSI optimize the same objective function, i.e., K-L divergence ([18]); 2) their solutions are equivalent because of the fixed row sum and fixed column sum property of NMF with K-L divergence; 3) Their algorithms are different because of the additional constraints in PLSI.

To begin with, we give the following lemma:

Lemma 1 (fixed row and column sums property, [18, 63]). *In NMF, under the update rules:*

$$F_{ik} := \frac{F_{ik}}{\sum_j G_{jk}} \sum_j \frac{X_{ij}}{(FG^T)_{ij}} G_{jk};$$

$$G_{jk} := \frac{G_{jk}}{\sum_i F_{ik}} \sum_i \frac{X_{ij}}{(FG^T)_{ij}} F_{ik},$$

Algorithm 6 Probabilistic Latent Semantic Indexing

```
Input: $C^0, S^0, H^0, t = 1$.
Output: $C, S, H$.
while 1 do
    Update $C_{ik}^{(t)} := C_{ik}^{(t-1)} \frac{(\frac{X.}{C^{(t-1)}S^{(t-1)}H^{(t-1)T}})H^{(t-1)})_{ik}}{(C^{(t-1)T}\frac{X.}{C^{(t-1)}S^{(t-1)}H^{(t-1)T}}H^{(t-1)})_{kk}}$;
    Update $S_{kk}^{(t)} := S_{kk}^{(t-1)}(C^{(t)T}\frac{X.}{C^{(t)}S^{(t-1)}H^{(t-1)T}}H^{(t-1)})_{kk}$;
    Update $H_{jk}^{(t)} := H_{jk}^{(t-1)} \frac{(\frac{X.}{C^{(t)}S^{(t)}H^{(t-1)T}})^T C^{(t)})_{jk}}{(C^{(t-1)T}\frac{X.}{C^{(t)}S^{(t)}H^{(t-1)T}}H^{(t-1)})_{kk}}$;
    Test for convergence.
    if Some convergence condition is satisfied then
        $C = C^{(t)}$;
        $S = S^{(t)}$;
        $H = H^{(t)}$;
        Break
    end if
    $t = t + 1$;
end while
```

we have, at convergence:

$$\sum_{i=1}^{n}(FG^T)_{ij} = \sum_{i=1}^{n} X_{ij}; \quad \sum_{j=1}^{m}(FG^T)_{ij} = \sum_{j=1}^{m} X_{ij}.$$

Proof. See Appendix.

Now we proceed to prove the model equivalence between NMF and PLSI.

Theorem 5. *NMF and PLSI optimize the same objective function.*

Proof. See Appendix.

Theorem 6. *([18, 19]) Any local maximum likelihood solution (C, S, H) of PLSI is a solution of NMF with K-L divergence and vice versa.*

Proof. This is obviously true by letting $F = C$ and $G = HS$ (or $F = CS$ and $G = H$) at convergence.

The conclusion that any local minimum solution (F, G) of NMF is a solution of PLSI can be proved similarly by normalizing F and G at convergence. □

From above analysis, one can see that NMF and PLSI optimize the same objective function, and the solution (F, G) of NMF and the solution (C, S, H) of PLSI are equivalent. Furthermore, we observe that at convergence, $FG^T = CSH^T$.

Consequently we will show that the algorithms of NMF and PLSI are different. To show this, we will firstly study the normalization of NMF. In other words, to compare the differences between NMF and PLSI more explicitly, we column normalize F and G at each step in NMF.

Obviously, in Algorithm 2, it holds that $F^{(t)}G^{(t-1)T} = (F^{(t)}A)(G^{(t-1)}B)^T$ for any two matrices A and B as long as $AB^T = I$ and $F^{(t)}A \geqslant 0$, $G^{(t-1)}B \geqslant 0$. If we select special A and B such that A is diagonal with $A_{kk} = \sum_i F_{ik}$ and $B = A^{-1}$, then $(F^{(t)}A)$ is column normalization of $F^{(t)}$. Similarly, we can get the column normalization of $G^{(t)}$. Based on these observations, we can revise the standard NMF algorithm as follows: after line 2 in Algorithm 2, we firstly column normalize $F^{(t)}$, and then replace $G^{(t-1)}$ by $(G^{(t-1)}B)^T$, consequently update $G^{(t-1)}$, then normalize $G^{(t)}$ and so on. Thus we get the normalization version of NMF algorithm:

Consequently, we give a conclusion on normalization of NMF. This conclusion can help us understand the algorithm differences between PLSI and NMF more clearly.

Theorem 7. *For NMF, at the $t-th$ iteration, given the triple factors $C^{(t-1)}$, diagonal matrix $S^{(t-1)}$ and $H^{(t-1)}$, which satisfy $\sum_i C_{ik}^{(t-1)} = 1, \sum_k S_{kk}^{(t-1)} = 1$ and $\sum_j H_{jk}^{(t-1)} = 1$, as initializations such that $F^{(t-1)} = C^{(t-1)}S^{(t-1)}$ and $G^{(t-1)} = H^{(t-1)}$ or $F^{(t-1)} = C^{(t-1)}$ and $G^{(t-1)} = H^{(t-1)}S^{(t-1)}$, the result $F^{(t)}$ can be equivalently formulated as*

$$C_{ik}^{(t)} := C_{ik}^{(t-1)} \frac{(\frac{X.}{C^{(t-1)}S^{(t-1)}H^{(t-1)T}} H^{(t-1)})_{ik}}{(C^{(t-1)T} \frac{X.}{C^{(t-1)}S^{(t-1)}H^{(t-1)T}} H^{(t-1)})_{kk}}, \tag{11}$$

$$S_{kk}^{(t)} := S_{kk}^{(t-1)} (C^{(t-1)T} \frac{X.}{C^{(t-1)}S^{(t-1)}H^{(t-1)^T}} H^{(t-1)})_{kk} \tag{12}$$

such that

$$F^{(t)} = C^{(t)}S^{(t)}. \tag{13}$$

The proof is omitted due to space limitation.

From above theorem, we can see that $C^{(t)}$ is column normalization of $F^{(t)}$, and the update rule of C is given. In corollary 1, we give an interesting property of $S^{(t)}$.

Corollary 1. *For NMF, at the $t-th$ iteration, $\sum\limits_i C_{ik}^{(t)} = 1$ and $\sum\limits_k S_{kk}^{(t)} = 1$.*

For G in NMF, we have similar result.

Corollary 2. *For NMF, at the $t-th$ iteration, given the triple factors $C^{(t-1)}$, diagonal matrix $S^{(t-1)}$ and $H^{(t-1)}$, which satisfy $\sum_i C_{ik}^{(t-1)} = 1, \sum_k S_{kk}^{(t-1)} = 1$ and $\sum_j H_{jk}^{(t-1)} = 1$, as initializations such that $F^{(t-1)} = C^{(t-1)}S^{(t-1)}$ and $G^{(t-1)} = H^{(t-1)}$ or $F^{(t-1)} = C^{(t-1)}$ and $G^{(t-1)} = H^{(t-1)}S^{(t-1)}$, the result $G^{(t)}$*

can be equivalently formulated as

$$H_{jk}^{(t)} := H_{jk}^{(t-1)} \frac{((\frac{X.}{C^{(t-1)}S^{(t-1)}H^{(t-1)T}})^T C^{(t-1)})_{jk}}{(C^{(t-1)T}\frac{X.}{C^{(t-1)}S^{(t-1)}H^{(t-1)T}}H^{(t-1)})_{kk}},$$

$$S_{kk}^{(t)} := S_{kk}^{(t-1)}(C^{(t-1)T}\frac{X.}{C^{(t-1)}S^{(t-1)}H^{(t-1)T}}H^{(t-1)})_{kk}$$

such that $G^{(t)} = H^{(t)}S^{(t)}$.

Based on the above discussions, we can revise Algorithm 2 to Algorithm 7.

Algorithm 7 Nonnegative Matrix Factorization*

```
Input: $C^{(0)}, S^{(0)}, H^{(0)}, t = 1$.
Output: $C, S, H$.
while 1 do
    Update $C_{ik}^{(t)} := C_{ik}^{(t-1)} \frac{(\frac{X.}{C^{(t-1)}S^{(t-1)}H^{(t-1)T}}H^{(t-1)})_{ik}}{(C^{(t-1)T}\frac{X.}{C^{(t-1)}S^{(t-1)}H^{(t-1)T}}H^{(t-1)})_{kk}}$;
    Update $S_{kk}^{(t)} := S_{kk}^{(t-1)}(C^{(t-1)T}\frac{X.}{C^{(t-1)}S^{(t-1)}H^{(t-1)T}}H^{(t-1)})_{kk}$;
    Update $H_{jk}^{(t)} := H_{jk}^{(t-1)} \frac{((\frac{X.}{C^{(t)}S^{(t)}H^{(t-1)T}})^T C^{(t)})_{jk}}{(C^{(t)T}\frac{X.}{C^{(t)}S^{(t)}H^{(t-1)T}}H^{(t-1)})_{kk}}$;
    Update $S_{kk}^{(t)} := S_{kk}^{(t)}(C^{(t)T}\frac{X.}{C^{(t)}S^{(t)}H^{(t-1)T}}H^{(t-1)})_{kk}$;
    Test for convergence.
    if Some convergence condition is satisfied then
        $C = C^{(t)}$;
        $S = S^{(t)}$;
        $H = H^{(t)}$;
        Break
    end if
    $t = t + 1$;
end while
```

Note that the normalization version of NMF will converge to a different local optimum from the standard NMF. But the revised version has a close relation with the standard one: any local optimum of Algorithm 7 is also a solution of Algorithm 2, and vice versa.

Theorem 8. *Any local optimum of Algorithm 7 is a solution of Algorithm 2.*

Proof. This is obviously true by joining line 2 and line 3, line 4 and line 5 in Algorithm 7.

After studying normalization of NMF carefully, we can now have a better insight into the algorithm differences between PLSI and NMF.

The following conclusions give the relations of C (in PLSI) and F (in NMF), H (in PLSI) and G (in NMF).

Theorem 9. *For PLSI and NMF, at the $t-th$ iteration, given the triple factors $C^{(t-1)}, S^{(t-1)}$ and $H^{(t-1)}$ as initializations of PLSI and $F^{(t-1)}, G^{(t-1)}$ as initializations of NMF such that $C^{(t-1)}S^{(t-1)} = F^{(t-1)}$ and $H^{(t-1)} = G^{(t-1)}$ or $C^{(t-1)} = F^{(t-1)}$ and $H^{(t-1)}S^{(t-1)} = G^{(t-1)}$ (i.e., $C^{(t-1)}S^{(t-1)}H^{(t-1)T} = F^{(t-1)}G^{(t-1)T}$), the update rules of C and F have the following relations: except for additional normalization, the update rule of C is identical with that of F in NMF, i.e., $C^{(t)} = F^{(t)}D_F^{-1}$, where D_F is diagonal matrix and the diagonal element $(D_F)_{kk} = \sum\limits_i F_{ik}^{(t)}$.*

Proof. The result is obviously true from (10), (11) , (12) and (13).

Corollary 3. *For PLSI and NMF, at the $t-th$ iteration, given the triple factors $C^{(t-1)}, S^{(t-1)}$ and $H^{(t-1)}$ as initializations of PLSI and $F^{(t-1)}, G^{(t-1)}$ as initializations of NMF such that $C^{(t-1)}S^{(t-1)} = F^{(t-1)}$ and $H^{(t-1)} = G^{(t-1)}$ or $C^{(t-1)} = F^{(t-1)}$ and $H^{(t-1)}S^{(t-1)} = G^{(t-1)}$ (i.e., $C^{(t-1)}S^{(t-1)}H^{(t-1)T} = F^{(t-1)}G^{(t-1)T}$), the update rules of H and G have the following relations: except for additional normalization, the update rule of H is identical with that of G in NMF, i.e., $H^{(t)} = G^{(t)}D_G^{-1}$, where D_G is diagonal matrix and the diagonal element $(D_F)_{kk} = \sum\limits_j G_{jk}^{(t)}$.*

Hence, NMF with normalization at each iteration has close relationship with PLSI. But this does not mean that PLSI can be replaced by NMF by normalizing F and G at each step, which can be observed from Algorithm 6 and Algorithm 7.

The key reason is that PLSI imposes normalization conditions on the factors explicitly. In [18] it has been shown that PLSI and NMF optimize the same objective function, hence PLSI can be seen as NMF-based model with additional normalization constraints ($\sum_i C_{ik} = 1, \sum_j H_{jk} = 1, \sum_k S_{kk} = 1$). The derivation process of PLSI update rules of C and H can be separated into two steps. Take the update rule of C while fixing S and H for example: firstly one gets the un-normalized C by gradient descent (identical with NMF), and then normalizes C to satisfy the constraint $\sum_i C_{ik} = 1$. The update rule of H is got in a similar way. The update rule of S can be got even more simply, just by gradient descent, and the normalization constraints will be satisfied automatically. In detail, at the $t-$th iteration, firstly, the derivative of the cost function $J(X, CSH^T)$ with respect to S while fixing C and H is:

$$\begin{aligned}\frac{\partial}{\partial S_{kk}}J &= -\sum_{ij}\frac{X_{ij}C_{ia}H_{ja}}{\sum\limits_{k} S_{kk}C_{ik}H_{jk}} + \sum_{ij} C_{ia}H_{ja}\\ &= -\sum_{ij}\frac{X_{ij}C_{ia}H_{ja}}{\sum\limits_{k} S_{kk}C_{ik}H_{jk}} + 1.\end{aligned}$$

Let the step size $\eta_{kk} = S_{kk}$, then the update rule of S is:

$$\begin{aligned}S_{kk} &= S_{kk} + \eta_{kk}(\sum_{ij}\frac{X_{ij}C_{ia}H_{ja}}{\sum\limits_{k} S_{kk}C_{ik}H_{jk}} - 1)\\ &= S_{kk}(C^T\frac{X}{CSH^T}H)_{kk}.\end{aligned}$$

Theorem 6 has shown that any local optimal solution of PLSI is also a solution of NMF with K-L divergence, and vice versa, and Theorem 8 has shown similar results between normalized NMF and standard NMF. These results mean that given the same initializations, PLSI, NMF and normalized NMF will give equivalent solutions. Furthermore, we observe that their solution values are always identical:

$$CSH^{T7} = FG^{T8} = F^*G^{*T9}. \tag{14}$$

Indeed, this phenomenon is very common in NMF. Roughly speaking, the standard NMF algorithm can be expressed like this: `update` F, then `update` G and so on. Now we revise it to: $\underbrace{\texttt{update } F,\ \texttt{update } F,\ \cdots, \texttt{update } F}_{\texttt{m times}}$, then $\underbrace{\texttt{update } G, \texttt{update } G, \cdots, \texttt{update } G}_{\texttt{n times}}$, and so on. Choosing different m and n, we can get infinitely many solutions even if given the same initializations. But these solutions are all having the same solution values.

Note that since PLSI has to update S at each iteration, it needs more running time than NMF.

6 Conclusions and Future Works

This chapter presents an overview of the major directions for research on Nonnegative Matrix Factorization, including the models, objective functions and algorithms, and the applications, as well as its relations with other models. We highlights the following conclusions: 1) Compared with Principal Component Analysis, NMF is more interpretable due to its nonnegative constraints; 2) NMF is very flexible. There are several choices of objective functions and algorithms to

[7] Results by PLSI

[8] Results by NMF

[9] Results by normalized NMF

accommodate a variety of applications; 3) NMF has linked K-means and PLSI, the two state-of-the-art unsupervised learning models, under the same framework; 4) NMF has a wide variety of applications and often has better numerical performance when compared with the other models/algorithms.

Finally, we list several open problems that are related to this chapter:

- there is still lack of systematic comparisons of the concordances and differences among the sparse variants of NMF. Note that generally speaking, the penalty that uses 1-norm should give more sparse results when compared with 2-norm since 2-norm often gives values that are very small rather than zeros, but 2-norm penalty is easier to calculate ([64, 5]);
- what are the relationships among the objective divergence functions, the algorithms and the applications? There is still lack of systematic analysis;
- why (14) holds? In other words, since they converge to different local solutions, why the solution values are always identical?
- how to tackle very large scale dataset in real applications? Distributed NMF([65]) seems an interesting direction.

Acknowledgement

The author is very appreciated the valuable comments of Dr. Jie Tang (Department of Computer Science and Technology, Tsinghua University, P.R.China) and Dr. Yong Wang (Academy of Mathematics and Systems Science, Chinese Academy of Sciences, P.R.China). This work is supported by the Foundation of Academic Discipline Program at Central University of Finance and Economics, P.R.China.

Appendix

Proof of Theorem 1:

First of all, rewrite $F = (F_{:,1}, F_{:,2}, \cdots, F_{:,r}), G = (G_{:,1}, G_{:,2}, \cdots, G_{:,r})$. Let

$$\begin{aligned} D_F &= diag(\max(F_{:,1}), \max(F_{:,2}), \cdots, \max(F_{:,r})), \\ D_G &= diag(\max(G_{:,1}), \max(G_{:,2}), \cdots, \max(G_{:,r})), \end{aligned}$$

where $\max(\bullet)$ is the largest element of column $\bullet$.

Note

$$D_F = D_F^{1/2} D_F^{1/2}, \qquad D_G = D_G^{1/2} D_G^{1/2}.$$

$$D_F^{-1} = D_F^{-1/2} D_F^{-1/2}, \qquad D_G^{-1} = D_G^{-1/2} D_G^{-1/2}.$$

We obtain

$$\begin{aligned} X = FG^T &= (FD_F^{-1})(D_F D_G)(GD_G^{-1})^T \\ &= (FD_F^{-1/2} D_G^{1/2})(GD_G^{-1/2} D_F^{1/2})^T. \end{aligned}$$

Construct D as $D = D_G^{-1/2} D_F^{1/2}$, then

$$F^* = F D_F^{-1/2} D_G^{1/2}, \qquad G^* = G D_G^{-1/2} D_F^{1/2}.$$

Thus (4) is proved.

Furthermore,

$$\begin{aligned}(F D_F^{-1/2} D_G^{1/2})_{ij} &= F_{ij} \cdot \sqrt{\frac{\max(G_{:,j})}{\max(F_{:,j})}} \\ &= \frac{F_{ij}}{\max(F_{:,j})} \cdot \sqrt{\max(F_{:,j}) \max(G_{:,j})}.\end{aligned}$$

Without loss of generality, assuming that

$$\max(F_{:,j}) = F_{tj}, \qquad \max(G_{:,j}) = G_{lj},$$

then we have

$$\begin{aligned}\max(F_{:,j}) \cdot \max(G_{:,j}) &\leq F_{t1} G_{1l}^T + \cdots F_{tj} G_{jl}^T + \cdots + F_{tr} G_{rl}^T \\ &= \sum_k F_{tk} G_{kl}^T = X_{tl} \leq M.\end{aligned}$$

So $0 \leq F_{ij}^* \leq \sqrt{M}$ and $0 \leq G_{ij}^* \leq \sqrt{M}$.
If X is symmetric and $F = G^T$,

$$G_{ij}^* = G_{ij} \cdot \sqrt{\frac{\max(G_{:,i})}{\max(G_{:,i})}} = G_{ij},$$

which implies $G^* = G$. □

Proof of Theorem 4:

Firstly, J_K can be rewritten as:

$$\begin{aligned}J_K &= \sum_{k=1}^{K} \sum_{i \in C_k} \|x_i - m_k\|^2 \\ &= c_2 - \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,j \in C_k} x_i^T x_j,\end{aligned}$$

where $c_2 = \sum_i \|x_i\|^2$. The clustering result can be represented by K nonnegative indicator vectors:

$$H = (h_1, h_2, \cdots, h_K), \quad h_k^T h_l = \delta_{kl} = \begin{cases} 1 & k = l \\ 0 & k \neq l \end{cases}$$

where $h_k = (0, \cdots, 0, \underbrace{1, \cdots, 1}_{n_k}, 0, \cdots, 0)^T / n_k^{1/2}$.

Now J_K becomes: $J_K = Tr(X^T X) - Tr(H^T X^T X H)$, where $Tr(\bullet)$ is the trace of matrix $\bullet$. Thus $\min J_K$ becomes

$$\max_{H^T H = I, H \geqslant 0} Tr(H^T W H), \tag{15}$$

where $W = X^T X$.

But $-2Tr(H^T W H) = \|W\|^2 - 2Tr(H^T W H) + \|H^T H\|^2 = \|W - H^T H\|^2$, hence,

$$\begin{aligned} H &= \arg \min_{H^T H = I, H \geqslant 0} -2Tr(H^T W H) \\ &= \arg \min_{H^T H = I, H \geqslant 0} \|W - H^T H\|^2. \end{aligned}$$

Relaxing the orthogonal constraint $H^T H = I$ completes the proof. □

Proof of Lemma 1:

At convergence, one has:

$$G_{jk} = \frac{G_{jk}}{\sum_i F_{ik}} \sum_i \frac{X_{ij} F_{ik}}{(FG^T)_{ij}}.$$

Hence

$$\begin{aligned} \sum_{i'} (FG^T)_{i'j} &= \sum_{i',k} F_{i'k} G_{jk} \\ &= \sum_{i',k} F_{i'k} \frac{G_{jk}}{\sum_i F_{ik}} \sum_i \frac{X_{ij} F_{ik}}{(FG^T)_{ij}} \\ &= \sum_k G_{jk} \sum_i \frac{X_{ij} F_{ik}}{(FG^T)_{ij}} \\ &= \sum_{i=1}^m X_{ij}. \end{aligned}$$

The other equality can be proven similarly. □

Proof of Theorem 5:

Firstly, we note that maximizing (7) can be rewritten as:

$$\min - \sum_{i=1}^m \sum_{j=1}^n X_{ij} \log P(w_i, d_j),$$

which is equivalent to

$$\min \sum_{i=1}^{m} \sum_{j=1}^{n} -X_{ij} \log P(w_i, d_j) + \sum_{i=1}^{m} \sum_{j=1}^{n} (X_{ij} \log X_{ij} - X_{ij} + (FG^T)_{ij}),$$

or

$$\min \sum_{i=1}^{m} \sum_{j=1}^{n} (X_{ij} \log \frac{X_{ij}}{P(w_i, d_j)} - X_{ij} + (FG^T)_{ij}),$$

since $\sum_{i=1}^{m} \sum_{j=1}^{n} X_{ij} \log X_{ij}$ is a constant and $\sum\limits_{i=1}^{m} \sum\limits_{j=1}^{n} (-X_{ij} + (FG^T)_{ij})$ cancels out at convergence by Lemma 1. Hence, by Theorem 6, PLSI and NMF optimize the same objective function. □

An Example to Illustrate the Limitations of NMF for Discovering Bi-clustering Structures

In fact, several papers [14, 15] have discussed about the bi-clustering aspect of NMF. But the key difficulty is that one can not identify the binary relationship of genes and samples exactly since the resulting matrices F and G are not binary. Here we give an example to illustrate the limitations of NMF for discovering bi-clustering structures. Given the original data matrix

$$X = \begin{pmatrix} 0.8 & 0.8 & 0.8 & 0.64 & 0.64 & 0.64 \\ 0.76 & 0.76 & 0.76 & 0.68 & 0.68 & 1.68 \\ 0.64 & 0.64 & 0.64 & 0.80 & 0.80 & 0.80 \\ 0.68 & 0.68 & 0.68 & 0.76 & 0.76 & 0.76 \\ 0.64 & 0.64 & 0.64 & 0.80 & 0.80 & 0.80 \end{pmatrix}.$$

Each row of X is a feature and each column of X is a sample.

We get the factor matrices F and G as follows:

$$F = \begin{pmatrix} 0.80 & 0.40 \\ 0.70 & 0.50 \\ 0.40 & 0.80 \\ 0.50 & 0.70 \\ 0.40 & 0.80 \end{pmatrix}; \; G^T = \begin{pmatrix} 0.8 & 0.8 & 0.8 & 0.4 & 0.4 & 0.4 \\ 0.4 & 0.4 & 0.4 & 0.8 & 0.8 & 0.8 \end{pmatrix}.$$

One can easily observe the clustering structures of the columns from G, but when identifying the bi-clustering structures, he(or she) has difficulties to identify an appropriate threshold to select which features should be involved in bi-clustering structures. From this small example we can see that standard NMF has limitations to discover bi-clustering structures explicitly.

References

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

2. Lee, D.D., Seung, H.S..: Algorithms for non-negative matrix factorization. In: Annual Conference on Neural Information Processing Systems. (2000) 556–562
3. Paatero, P., Tapper, U.: Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. Environmetrics **5**(2) (1994) 111–126
4. Jolliffe, I.T.: Principal Component Analysis. Second edn. Springer (2002)
5. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics). 2nd ed. 2009. corr. 3rd printing edn. Springer (2009)
6. Tropp, J.A.: Literature survey: Non-negative matrix factorization. Unpublished document, University of Texas at Austin, Austin, TX. (2003)
7. Xie, Y.L., Hopke, P., Paatero, P.: Positive matrix factorization applied to a curve resolution problem. Journal of Chemometrics **12**(6) (1999) 357–364
8. Li, S. Z., Hou, X. W., Zhang, H. J., Cheng, Q. S., Learning spatially localized, parts-based representation. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR. **1** (2001) I-207-I-212.
9. Cooper, M., Foote, J.: Summarizing video using non-negative similarity matrix factorization. In: Multimedia Signal Processing, 2002 IEEE Workshop on. (2002) 25 – 28
10. Pauca, V.P., Shahnaz, F., Berry, M.W., Plemmons, R.J.: Text mining using non-negative matrix factorizations. In: Proceedings of the Fourth SIAM International Conference on Data Mining. (2004)
11. Shahnaz, F., Berry, M.W., Pauca, Plemmons, R.J.: Document clustering using nonnegative matrix factorization. Information Processing & Management **42**(2) (2006) 373–386
12. Xu, W., Liu, X., Gong, Y.: Document clustering based on non-negative matrix factorization. In: SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, New York, NY, USA, ACM Press (2003) 267–273
13. Nielsen, F.A., Balslev, D., Hansen, L.K.: Mining the posterior cingulate: Segregation between memory and pain components. NeuroImage **27**(3) (2005) 520–532
14. Brunet, J.P., Tamayo, P., Golub, T.R., Mesirov, J.P.: Metagenes and molecular pattern discovery using matrix factorization. Proc Natl Acad Sci U S A **101**(12) (2004) 4164–4169
15. Pascual-Montano, A., Carazo, J.M., Kochi, K., Lehmann, D., Pascual-Marqui, R.D.: Nonsmooth nonnegative matrix factorization (nsNMF). IEEE transactions on Pattern Analysis and Machine Intelligence **28**(3) (2006) 403–415
16. Devarajan, K.: Nonnegative matrix factorization: An analytical and interpretive tool in computational biology. PLoS Comput Biol **4**(7) (2008) e1000029
17. Ding, C., He, X., Simon, H.D.: On the equivalence of nonnegative matrix factorization and spectral clustering. In: SIAM Data Mining Conf. (2005)
18. Ding, C., Li, T., Peng, W.: On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. Comput. Stat. Data Anal. **52**(8) (2008) 3913–3927
19. Gaussier, E., Goutte, C.: Relation between PLSA and NMF and implications. In: SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM (2005) 601–602
20. Zhang, Z.Y., Li, T., Ding, C., Zhang, X.S.: Binary matrix factorization with applications. Data Mining, IEEE International Conference on (2007) 391–400

21. Zhang, Z.Y., Li, T., Ding, C., Ren, X.W., Zhang, X.S.: Binary matrix factorization for analyzing gene expression data. Data Min. Knowl. Discov. **20**(1) (2010) 28–52
22. Ding, C.H.Q., Li, T., Jordan, M.I.: Convex and semi-nonnegative matrix factorizations. IEEE Trans. Pattern Anal. Mach. Intell. **32**(1) (2010) 45–55
23. Ding, C., Li, T., Peng, W., Park, H.: Orthogonal nonnegative matrix t-factorizations for clustering. In: KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM (2006) 126–135
24. Li, T., Ding, C.: The relationships among various nonnegative matrix factorization methods for clustering. In: ICDM '06: Proceedings of the Sixth International Conference on Data Mining, Washington, DC, USA, IEEE Computer Society (2006) 362–371
25. Li, S.Z., Hou, X.W., Zhang, H.J., Cheng, Q.S.: 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. (2001)
26. Feng, T., Li, S., Shum, H.Y., Zhang, H.: Local non-negative matrix factorization as a visual representation. In: Development and Learning, International Conference on. (2002)
27. Hoyer, P.O.: Non-negative matrix factorization with sparseness constraints. Journal of Machine Learning Research **5** (2004) 1457-1469
28. Hoyer, P.O.: Non-negative sparse coding. In: Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on. (2002) 557–565
29. Liu, W., Zheng, N., Lu, X.: Non-negative matrix factorization for visual coding. In: Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'2003)
30. Gao, Y., Church, G.: Improving molecular cancer class discovery through sparse non-negative matrix factorization. Bioinformatics **21**(21) (2005) 3970–3975
31. Pauca, V.P., Piper, J., Plemmons, R.J.: Nonnegative matrix factorization for spectral data analysis. Linear Algebra and its Applications **416**(1) (2006) 29–47
32. Hoyer, P.O.: Non-negative matrix factorization with sparseness constraints. J. Mach. Learn. Res. **5** (2004) 1457–1469
33. Kim, H., Park, H.: Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. Bioinformatics **23**(12) (2007) 1495–1502
34. Mahoney, M.W., Drineas, P.: CUR matrix decompositions for improved data analysis. Proc Natl Acad Sci U S A **106**(3) (2009) 697–702
35. Cichocki, A., Zdunek, R., Amari, S.: Csiszár's divergences for non-negative matrix factorization: Family of new algorithms. In Proc. Int'l Conf. Independent Component Analysis and Blind Signal Separation. (2006) 32–39
36. Cichocki, A., Lee, H., Kim, Y.D., Choi, S.: Non-negative matrix factorization with α-divergence. Pattern Recogn. Lett. **29**(9) (2008) 1433–1440
37. Cichocki, A., Amari, S., Zdunek, R., Kompass, R., Hori, G., He, Z.: Extended smart algorithms for non-negative matrix factorization. In: Artificial Intelligence and Soft Computing - ICAISC 2006, 8th International Conference, Zakopane, Poland, June 25-29, 2006, Proceedings. (2006) 548–562
38. Liu, W., Yuan, K., Ye, D. On alpha-divergence based nonnegative matrix factorization for clustering cancer gene expression data. Artif Intell Med. **44**(1) (2008) 1-5
39. Cichocki, A., Zdunek, R., Choi, S., Plemmons, R., Amari, S.: Nonnegative tensor factorization using alpha and beta divergencies. In: Proc. IEEE International

Conference on Acoustics, Speech, and Signal Processing (ICASSP07). (2007) 1393–1396

40. Dhillon, I.S., Sra, S.: Generalized nonnegative matrix approximations with bregman divergences. In: Proc. Advances in Neural Information Proc. Systems (NIPS). (2005) 283–290
41. Kompass, R.: A generalized divergence measure for nonnegative matrix factorization. Neural Comput. **19**(3) (2007) 780–791
42. Cichocki, A., Zdunek, R., Phan, A.H., ichi Amari, S.: Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation. A John Wiley and Sons, Ltd, Publication (2009)
43. Févotte, C., Bertin, N., Durrieu, J.L.: Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis. Neural Comput. **21**(3) (2009) 793–830
44. Gonzalez, E.F., Zhang, Y.: Accelerating the lee-seung algorithm for nonnegative matrix factorization. Technical Report (2005)
45. Lin, C.J.: Projected gradient methods for nonnegative matrix factorization. Neural Comput. **19**(10) (2007) 2756–2779
46. Kim, D., Sra, S., Dhillon, I.S.: Fast newton-type methods for the least squares nonnegative matrix approximation problem. In: Data Mining, Proceedings of SIAM Conference on. (2007) 343–354
47. Zdunek, R., Cichocki, A.: Non-negative matrix factorization with quasi-newton optimization. In: Eighth International Conference on Artificial Intelligence and Soft Computing, ICAISC, Springer (2006) 870–879
48. Kim, H., Park, H.: Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method. SIAM J. Matrix Anal. Appl. **30**(2) (2008) 713–730
49. Long, B., Wu, X., Zhang, Z., Yu, P.S.: Community learning by graph approximation. In: Proceedings of the 2007 Seventh IEEE International Conference on Data Mining, ICDM 2007 (2007): 232-241
50. Chen, Y., Rege, M., Dong, M., Hua, J.: Incorporating user provided constraints into document clustering. In: Proceedings of the 2007 Seventh IEEE International Conference on Data Mining, ICDM 2007 (2007) 103–112
51. Kulis, B., Basu, S., Dhillon, I., Mooney, R.: Semi-supervised graph clustering: a kernel approach. In: ICML '05: Proceedings of the 22nd international conference on Machine learning, New York, NY, USA, ACM (2005) 457–464
52. Ji, X., Xu, W.: Document clustering with prior knowledge. In: SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM (2006) 405–412
53. Cheng, Y., Church, G.M.: Biclustering of expression data. In: Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology, AAAI Press (2000) 93–103
54. Prelić, A., Bleuler, S., Zimmermann, P., Wille, A., Bühlmann, P., Gruissem, W., Hennig, L., Thiele, L., Zitzler, E.: A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics **22**(9) (2006) 1122–1129
55. Drakakis, K., Rickard, S., de Frein, R., Cichocki, A.: Analysis of financial data using non-negative matrix factorization. International Mathematical Forum **3**(38) (2008) 1853–1870
56. Ribeiro, B., Silva, C., Vieira, A., das Neves, J.C.: Extracting discriminative features using non-negative matrix factorization in financial distress data. In: Proceedings of

the 9th International Conference on Adaptive and Natural Computing Algorithms, ICANNGA 2009. (2009) 537–547

57. Zha, H., He, X., Ding, C., Simon, H.: Spectral relaxation for k-means clustering. In: Proc. Advances in Neural Information Proc. Systems (NIPS). (2001) 1057–1064
58. Ding, C., He, X.: K-means clustering via principal component analysis. In: Proceedings of the twenty-first international conference on Machine learning (ICML04). (2004) 225–232
59. Hofmann, T.: Probabilistic latent semantic indexing. In: SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, ACM Press (1999) 50–57
60. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. Journal of the American Society of Information Science **41**(6) (1990) 391–407
61. Wu, X., Yan, J., Liu, N., Yan, S., Chen, Y., Chen, Z.: Probabilistic latent semantic user segmentation for behavioral targeted advertising. In: ADKDD '09: Proceedings of the Third International Workshop on Data Mining and Audience Intelligence for Advertising, New York, NY, USA, ACM (2009) 10–17
62. Cohn, D., Hofmann, T.: The missing link - a probabilistic model of document content and hypertext connectivity. In: Proc. Advances in Neural Information Proc. Systems (NIPS). (2001)
63. Ho, N.D., Dooren, P.V.: Non-negative matrix factorization with fixed row and column sums. Linear Algebra and its Applications **429** (2008) 1020–1025
64. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society, Series B **58** (1996) 267–288
65. Liu, C., Yang, H.c., Fan, J., He, L.W., Wang, Y.M.: Distributed nonnegative matrix factorization for web-scale dyadic data analysis on mapreduce. In: Proceedings of the 19th international conference on World wide web (WWW10). (2010) 681–690