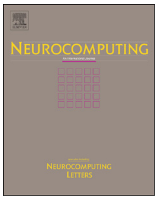
Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



General subspace constrained non-negative matrix factorization for data representation

Yong Liu [a,b,*], Yiyi Liao [b], Liang Tang [c], Feng Tang [a], Weicong Liu [b]

[a] State Key Lab of Industrial Technology Control Technology, Zhejiang University, Hangzhou 310027, China
[b] Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, China
[c] China Ship Development and Design Center, Wuhan 430064, China

ARTICLE INFO

Article history:
Received 14 June 2014
Received in revised form
10 November 2014
Accepted 16 November 2014
Available online 3 September 2015

Keywords:
Non-negative matrix factorization
Subspace
Generalize framework

ABSTRACT

Nonnegative matrix factorization (NMF) has been proved to be a powerful data representation method, and has shown success in applications such as data representation and document clustering. However, the non-negative constraint alone is not able to capture the underlying properties of the data. In this paper, we present a framework to enforce general subspace constraints into NMF by augmenting the original objective function with two additional terms. One on constraints of the basis, the other on preserving the structural properties of the original data. This framework is general as it can be used to regularize NMF with a wide variety of subspace constraints that can be formulated into a certain form such as PCA, Fisher LDA and LPP. In addition, we present an iterative optimization algorithm to solve the general subspace constrained non-negative matrix factorization (GSC NMF). We show that the resulting subspace has enriched representation power as shown in our experiments.

1. Introduction

Finding a suitable representation is a fundamental problem in many machine learning tasks, such as pattern recognition and object detection [1–7]. A good representation can capture the underlying structure of the data and can reduce the dimensionality so as to make the higher level inference easier. Subspace representations construct a subspace from the original high dimensional space and represent them as the projection on the subspace. It has been shown that subspace methods not only reduces the computational cost due to the lower dimensionality but also makes the higher level inference easier.

Subspace methods such as principal component analysis (PCA) [8,9,1], linear discriminative analysis (LDA) [10,11] and locality preserving projection (LPP) [12] can be understood as matrix factorization subject to different constraints. These constraints are usually designed to find basis functions satisfying certain properties. Principal components analysis enforces an orthogonality constraint of the basis vectors, resulting in an orthogonal subspace to capture the major variance of the data. As a well-known dimension reduction method, PCA is extended in different ways, such as incremental learning and tensor analysis [13–15]. Extension approaches of LDA and LPP are also proposed for performance improvement [16–19]. However, the resulting basis and coefficient vectors can be negative, which does not have intuitive psychological interpretation. Non-negative Matrix Factorization (NMF) is a subspace method with nonnegative constraints on both the basis and coefficients. The non-negative constraints lead to a parts-based representation because they allow only additive, not subtractive combinations. Such a representation encodes the data using few active components, which makes the basis easy to interpret. The previous research works have shown the superior performance of NMF on document clustering [20], text mining [21,22], pattern recognition [23,24] and audio analysis [25,26].

However, the non-negative constraints alone may not be enough to capture the underlying structure of the data as other subspace methods for example PCA do. In this paper, we present a framework to enforce general subspace constraints into NMF. This framework is general as it can be used to regularize NMF with a wide variety of subspace constraints that can be formulated into a certain form such as PCA, LDA and LPP. In addition, we present an iterative optimization algorithm to solve the general subspace constrained NMF. We show that the resulting subspace has enriched representation power as shown in our experiments.

There are also some other work that tries to incorporate constraints into the NMF. Local non-negative matrix factorization (LNMF) [27] has been proposed to achieve a more localized NMF algorithm with the aim of computing spatially localized basis adding orthogonality constraints that modify the objective function.

[*] Corresponding author at: State Key Lab of Industrial Technology Control Technology, Zhejiang University, Hangzhou 310027, China
E-mail address: yongliu@iipc.zju.edu.cn (Y. Liu).

http://dx.doi.org/10.1016/j.neucom.2014.11.099
0925-2312/© 2015 Elsevier B.V. All rights reserved.

Some similar works focus on constraining the orthogonality such as [28] and [29]. The former solves the optimization problem with the orthogonality constraints, while the later embeds the constraints as part of the cost function. In Sparse NMF [30], the author enforces the sparseness constraints explicitly in the objective function. However, these algorithms and their solutions are specifically designed for a particular constraint, which are in contrast with our approach since we aim at providing a general theoretical framework and solution.

The rest of the paper is organized as follows: Section 2 gives a brief review of the NMF. The general theoretical framework for NMF with subspace constraints and their three examples PCA NMF, Fisher NMF and LPP NMF are presented in Section 3. The optimization algorithm is discussed in Section 4. The experimental results will be shown in Section 5 and we conclude the paper in Section 6.

Table 1
Various subspace constrained NMF via different parameter settings.

	C	α	β	$\mathbf{L}$
LNMF [27,33]	C_1 or C_2	$\alpha > 0$	$\beta > 0$	$\mathbf{I}$
PCA NMF	C_1 or C_2	$\alpha > 0$	$\beta > 0$	$\mathbf{L} = \frac{1}{n}\mathbf{I} - \frac{1}{n^2}\mathbf{ee}^{\mathrm{T}}$
Fisher NMF	C_1 or C_2	$\alpha = 0$	$\beta < 0$	$\mathbf{L} = \mathbf{I} - 2\mathbf{W} + \frac{1}{n}\mathbf{ee}^{T}$
LPP NMF	C_1 or C_2	$\alpha = 0$	$\beta < 0$	$\mathbf{L} = \mathbf{D} - \mathbf{S}$

2. A brief review of NMF

Generally, NMF [31] can be presented as the following optimization problem:

$$\min_{\mathbf{B},\mathbf{H}} C(\mathbf{X} \approx \mathbf{BH}), \quad \text{s. t.} \quad \mathbf{B}, \mathbf{H} \geq 0$$

Here, $\mathbf{X} = [x_{ij}] \in \mathcal{R}^{d \times n}$, each column of $\mathbf{X}$ is a sample vector. NMF aims to find two non-negative matrices $\mathbf{B} = [b_{ij}] \in \mathcal{R}^{d \times r}$ and $\mathbf{H} = [h_{ij}] \in \mathcal{R}^{r \times n}$ whose product can well approximate the original matrix $\mathbf{X}$. $C(\cdot)$ denotes the cost function. There are normally two kinds of cost functions to represent the approximation in the NMF optimization. Let $\mathbf{Y}=\mathbf{BH}$, the first cost function is the Euclidean distance between two matrices:

$$C_1 = \|\mathbf{X} - \mathbf{Y}\|^2 = \sum_{ij} \left(x_{ij} - y_{ij}\right)^2$$

The second cost function is the K–L divergence between two matrices:

$$C_2 = D(\mathbf{X} \,\|\, \mathbf{Y}) = \sum_{ij} \left(x_{ij} \log \frac{x_{ij}}{y_{ij}} - x_{ij} + y_{ij} \right)$$

Although the C_1 and C_2 are convex in $\mathbf{B}$ only or $\mathbf{H}$ only, they are not convex in both variables together. Thus Lee and Seung [32] presented iterative update algorithms to find the local minima of the objective function C_1 and C_2 [32].

3. General subspace constrained NMF

In this section, we present a general subspace constrained non-negative matrix factorization (GSC NMF) framework, which can enforce various subspace constraints into NMF. Let $\mathbf{U} = [u_{ij}] = \mathbf{B}^{\mathbf{T}}\mathbf{B}$, the problem is formulated as follows:

$$O_U = C(\mathbf{X} \approx \mathbf{BH}) + \alpha \sum_{i,j} u_{ij} - \beta \,\mathrm{Tr}(\mathbf{HLH}^{\mathbf{T}}) \tag{1}$$

α, β are const real number, and $\mathbf{L} \in \mathcal{R}^{n \times n}$ is the parameter matrix. The cost function C is either C_1 or C_2. When $\alpha > 0$, minimizing $\sum_{ij} u_{ij}$ leads to the basis (b_i), which are orthogonal [33,27]. Table 1 shows different parameter settings for the GSC NMF and their corresponding subspace constrains. With page limits, we will only introduce the PCA NMF, Fisher NMF and LPP NMF in detail in the following sections.

3.1. PCA NMF

The main idea of classical PCA is trying to maximize the representation vectors' variance while keeping the orthogonality of the basis. Assuming that the sample data set is $\{x_1, x_2, \ldots, x_n\}$, and the linear transform for PCA can be denoted as $b^{\mathrm{T}}x_i = y_i$, here b is the basis vector of $\mathbf{B}$ and y_i are the representation vector. Then the optimization function of PCA can be denoted as

$$\max_{\mathbf{b}} \sum_{i=1}^{n} (y_i - \bar{y})^2, \quad \bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$$

The vectors of $b_1, b_2, \ldots, b_r$ are orthogonal.

When concerning the NMF form of $\mathbf{X} \approx \mathbf{BH}$, we have $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$, $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_r]$ and $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n]$. Then the column vector of $\mathbf{H}$ can be viewed as the projection of original data set $\mathbf{X}$ in the subspace constructing with the column vectors of $\mathbf{B}$, thus $\mathbf{x}_i \approx \mathbf{Bh}_i$.

We then let $\mathbf{L} = (1/n)\mathbf{I} - (1/n^2)\mathbf{ee}^{\mathrm{T}}$. $\mathbf{I}$ is the identity matrix with order of n and $\mathbf{e}$ is the n dimensional vector with all the elements equaling to 1. We use $\mathbf{m}$ to denote the mean of the project vectors, that is $\mathbf{m} = (1/n) \sum_{i=1}^{n} \mathbf{h}_i$, and then

$$\begin{aligned}
\mathbf{HLH}^{\mathrm{T}} &= \frac{1}{n}\mathbf{H}(\mathbf{I} - \frac{1}{n}\mathbf{ee}^{\mathrm{T}})\mathbf{H}^{\mathrm{T}} \\
&= \frac{1}{n}\mathbf{HH}^{\mathrm{T}} - \frac{1}{n^2}(\mathbf{He})(\mathbf{He})^{\mathrm{T}} \\
&= \frac{1}{n}\sum_i \mathbf{h}_i\mathbf{h}_i^{\mathrm{T}} - \frac{1}{n^2}(n\mathbf{m})(n\mathbf{m})^{\mathrm{T}} \\
&= \frac{1}{n}\sum_i (\mathbf{h}_i - \mathbf{m})(\mathbf{h}_i - \mathbf{m})^{\mathrm{T}} \\
&\quad + \frac{1}{n}\sum_i \mathbf{h}_i\mathbf{m}^{\mathrm{T}} + \frac{1}{n}\sum_i \mathbf{m}\mathbf{h}_i^{\mathrm{T}} - \frac{1}{n}\sum_i \mathbf{mm}^{\mathrm{T}} - \mathbf{mm}^{\mathrm{T}} \\
&= E[(\mathbf{h} - \mathbf{m})(\mathbf{h} - \mathbf{m})^{\mathrm{T}}] + 2\mathbf{mm}^{\mathrm{T}} - 2\mathbf{mm}^{\mathrm{T}} \\
&= E[(\mathbf{h} - \mathbf{m})(\mathbf{h} - \mathbf{m})^{\mathrm{T}}].
\end{aligned}$$

Here, $E[(\mathbf{h} - \mathbf{m})(\mathbf{h} - \mathbf{m})^{\mathrm{T}}]$ is the covariance matrix of the projections and thus maximizing $\mathbf{HLH}^{\mathrm{T}}$ is equivalent to maximizing $\sum_{i=1}^{n} \left\| \mathbf{h}_i - \mathbf{m} \right\|$, which is the core optimization function of PCA. At the same time, minimizing $\sum_{i \neq j} u_{ij}$ will guarantee that all the basis vectors are orthogonal.

Then let $\alpha, \beta > 0$ and $\mathbf{L} = (1/n)\mathbf{I} - (1/n^2)\mathbf{ee}^{\mathrm{T}}$, the optimization function (1) is a PCA constrained NMF.

3.2. Fisher NMF

There are several works [34] and [35] introduce the Fisher's discriminative information to NMF and show their ability on face recognition and classification. The DNMF proposed in [35] can also be unified to our GSC NMF, which we call Fisher NMF.

Linear discriminant analysis (LDA) tries to find a projection direction that can separate the different classes. Define the "between classes scatter matrix" $\mathbf{S_B}$ and the "within classes scatter matrix" $\mathbf{S_W}$ which reveal the original data distribution information. Fisher LDA expects to find a linear transformation matrix $\mathbf{W}$ to maximize the $\tilde{\mathbf{S}}_\mathbf{W}$ and minimize the $\tilde{\mathbf{S}}_\mathbf{B}$ on the projection space, where

$$\tilde{\mathbf{S}}_W = \mathbf{W}^T\mathbf{S_W}\mathbf{W}$$
$$\tilde{\mathbf{S}}_B = \mathbf{W}^T\mathbf{S_B}\mathbf{W}$$

To this purpose, the $\mathbf{W}$ can be solved by maximizing the cost function:

$$J(\mathbf{w}) = \mathrm{Tr}(\tilde{\mathbf{S}}_\mathbf{B} - \tilde{\mathbf{S}}_\mathbf{W}) = \mathrm{Tr}(\mathbf{W^T S_B W} - \mathbf{W^T S_W W})$$

here we use the trace of a matrix as its scalar measurement.

Given l classes of input data with n_i samples in ith class, perform the Fisher LDA on the data set. Denote $\mathbf{y}_j^i$ as the projection of jth sample in ith class, and $\mathbf{m}^i$ is the mean of all projected samples in ith class. Now we can calculate the within classes scatter matrix $\tilde{\mathbf{S}}_\mathbf{W}$ of the projected samples as

$$\begin{aligned}
\tilde{\mathbf{S}}_\mathbf{W} = & \sum_{i=1}^{l}\left(\sum_{j=1}^{n_i}\left(\mathbf{y}_j^i - \mathbf{m}^i\right)\left(\mathbf{y}_j^i - \mathbf{m}^i\right)^T\right) \\
= & \sum_{i=1}^{l}\left(\sum_{j=1}^{n_i}\left(\mathbf{y}_j^i\left(\mathbf{y}_j^i\right)^T - \mathbf{m}^i\left(\mathbf{y}_j^i\right)^T - \mathbf{y}_j^i\left(\mathbf{m}^i\right)^T + \mathbf{m}^i\left(\mathbf{m}^i\right)^T\right)\right) \\
= & \sum_{i=1}^{l}\left(\sum_{j=1}^{n_i}\mathbf{y}_j^i\left(\mathbf{y}_j^i\right)^T - n_i\mathbf{m}^i\left(\mathbf{m}^i\right)^T\right) \\
= & \sum_{i=1}^{l}\left(\mathbf{Y}_i\mathbf{Y}_i^T - \frac{1}{n_i}\left(\mathbf{y}_1^i + \cdots + \mathbf{y}_{n_i}^i\right)\left(\mathbf{y}_1^i + \cdots + \mathbf{y}_{n_i}^i\right)^T\right) \\
= & \sum_{i=1}^{l}\left(\mathbf{Y}_i\mathbf{Y}_i^T - \frac{1}{n_i}\mathbf{Y}_i\left(\mathbf{e}_i\mathbf{e}_i^T\right)\mathbf{Y}_i^T\right) \\
= & \sum_{i=1}^{l}\mathbf{Y}_i\mathbf{L}_i\mathbf{Y}_i^T
\end{aligned} \tag{2}$$

where $\mathbf{Y}_i\mathbf{L}_i\mathbf{Y}_i^T$ is the covariance matrix of the projected samples in ith class with $\mathbf{Y}_i = [\mathbf{y}_1^i, \mathbf{y}_2^i, \ldots, \mathbf{y}_{n_i}^i]$. $\mathbf{L}_i = \mathbf{I} - (1/n_i)\mathbf{e}_i\mathbf{e}_i^T$ is a $n_i \times n_i$ matrix where $\mathbf{I}$ is an identity matrix, and $\mathbf{e}_i = (1, 1, \ldots, 1)^T$ is a n_i dimensional vector. For further simplification, let $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n]$ denote the matrix of all projected samples where $n = \sum_{i=1}^{l} n_i$ and $\mathbf{y}_i$ is the projected vector corresponds to $\mathbf{x}_i$. Besides, define a $n \times n$ matrix which encodes the label information as follows:

$$\mathbf{W}_{ij} = \begin{cases} 1/n_k & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ both belong to the } k\text{th class,} \\ 0 & \text{otherwise.} \end{cases}$$

Let $\mathbf{L_W} = \mathbf{I} - \mathbf{W}$, then $\mathbf{S_W}$ can be rewritten as

$$\tilde{\mathbf{S}}_W = \mathbf{Y L_W Y}^T$$

Apparently, the between classes scatter matrix $\tilde{\mathbf{S}}_\mathbf{B}$ on the projection space can be represented in a similar way. Denote $\mathbf{m}$ as the mean of all projected samples in all classes, we have

$$\begin{aligned}
\tilde{\mathbf{S}}_B = & \sum_{i=1}^{l}\left(\left(\mathbf{m}^i - \mathbf{m}\right)\left(\mathbf{m}^i - \mathbf{m}\right)^T\right) \\
= & \left(\sum_{i=1}^{l} n_i\mathbf{m}^i\left(\mathbf{m}^i\right)^T\right) - \mathbf{m}\left(\sum_{i=1}^{l} n_i\left(\mathbf{m}^i\right)^T\right) \\
& - \left(\sum_{i=1}^{l} n_i\left(\mathbf{m}^i\right)\right)\mathbf{m}^T + \left(\sum_{i=1}^{l} n_i\right)\mathbf{m}\mathbf{m}^T \\
= & \left(\sum_{i=1}^{l}\frac{1}{n_i}\left(\mathbf{y}_1^i + \cdots + \mathbf{y}_{n_i}^i\right)\left(\mathbf{y}_1^i + \cdots + \mathbf{y}_{n_i}^i\right)^T\right) \\
& - 2n\mathbf{m}\mathbf{m}^T + n\mathbf{m}\mathbf{m}^T \\
= & \left(\sum_{i=1}^{l}\sum_{j,k=1}^{n_i}\frac{1}{n_i}\mathbf{y}_j^i\left(\mathbf{y}_k^i\right)^T\right) - n\mathbf{m}\mathbf{m}^T \\
= & \mathbf{YWY}^T - n\mathbf{m}\mathbf{m}^T \\
= & \mathbf{YWY}^T - \mathbf{Y}(\frac{1}{n}\mathbf{e}\mathbf{e}^T)\mathbf{Y}^T \\
= & \mathbf{Y}(\mathbf{W} - \frac{1}{n}\mathbf{e}\mathbf{e}^T)\mathbf{Y}^T \\
= & \mathbf{Y L_B Y}^T
\end{aligned} \tag{3}$$

where $\mathbf{L_B} = \mathbf{W} - (1/n)\mathbf{e}\mathbf{e}^T$, $\mathbf{e} = (1, 1, \ldots, 1)^T$ is a n dimensional vector.

Recall that in the case of NMF where $\mathbf{X} \approx \mathbf{BH}$, $\mathbf{H}$ can be regarded as the projection of $\mathbf{X}$ in the subspace constructing with the column vectors of $\mathbf{B}$, which means $\mathbf{H}$ corresponds to the projected matrix $\mathbf{Y}$ in formulas (2) and (3). Similarly, $\mathbf{H L_W H}^T$ represents the within classes scatter matrix $\tilde{\mathbf{S}}_\mathbf{W}$ of projected samples, and $\mathbf{H L_B H}^T$ represents the between classes scatter matrix $\tilde{\mathbf{S}}_\mathbf{B}$ at the same time.

Define:

$$\mathbf{L} = \mathbf{L_W} - \mathbf{L_B} = \mathbf{I} - 2\mathbf{W} + \frac{1}{n}\mathbf{e}\mathbf{e}^T$$

Then minimizing $\mathrm{Tr}(\mathbf{HLH^T})$ is equal to optimize the discrimination of projection. It is to say, to maximize the between classes scatter and minimize the within classes scatter simultaneously.

Then let $\mathbf{L} = \mathbf{I} - 2\mathbf{W} + (1/n)\mathbf{e}\mathbf{e}^T$, $\alpha = 0$ and $\beta < 0$, the optimization function (1) is a Fisher NMF with the constraint of discrimination.

3.3. LPP NMF

Locality Preserving Projection (LPP) [12] is a typical subspace learning method which poses local structure, which implements similar idea to the popular nonlinear methods in manifold learning, such as Locally Linear Embedding (LLE) [36] and Laplacian Eigenmap [37]. Introduce the local invariance idea to NMF, one can implement the LPP NMF, which is the same as the GNMF proposed by Cai et al. [38].

The constraint of LPP is minimizing the following formula:

$$\sum_{ij}\|\mathbf{h}_i - \mathbf{h}_j\|^2\mathbf{S}_{ij}$$

Here, $\mathbf{h}_i$ is the r dimensional vector corresponding to the original vector x_i after the matrix factorization of $\mathbf{X} \approx \mathbf{BH}$, $\mathbf{h}_i$ is a projection of x_i in r dimensional subspace. $\|\cdot\|$ is the Euclidean distance operation. $\mathbf{S}$ is the similarity matrix, which represents the local structure of the original data set $\mathbf{X}$. The constraint function will represent that the subspace of $\mathbf{H}$ can keep the same local geometrical structure as $\mathbf{X}$. The $\mathbf{S}$ can be defined as

$$\mathbf{S}^1_{ij} = \begin{cases} \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/t), & \|\mathbf{x}_i - \mathbf{x}_j\|^2 < \varepsilon \\ 0 & \text{otherwise} \end{cases}$$

or

$$\mathbf{S}^2_{ij} = \begin{cases} \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/t) & \mathbf{x}_i \in \mathrm{kNN}(\mathbf{x}_j) | \mathbf{x}_j \in \mathrm{kNN}(\mathbf{x}_i) \\ 0 & \text{otherwise} \end{cases}$$

Here $\varepsilon > 0$ and $kNN(x)$ denote x's k nearest neighbor set. h_i is a vector with r dimensions and $h_i^{(k)}$ is the kth dimensional element of h_i, considering the optimization function of LPP:

$$\begin{aligned}
&\frac{1}{2}\sum_{ij}\|\mathbf{h}_i - \mathbf{h}_j\|^2\mathbf{S}_{ij} \\
&= \frac{1}{2}\sum_{k=1}^{r}\sum_{ij}\left(h_i^{(k)} - h_j^{(k)}\right)^2\mathbf{S}_{ij} \\
&= \sum_{k=1}^{r}\left(\sum_{ij} h_i^{(k)}\mathbf{S}_{ij}h_i^{(k)} - \sum_{ij} h_i^{(k)}\mathbf{S}_{ij}h_j^{(k)}\right) \\
&= \sum_{k=1}^{r}\left(\sum_{i} h_i^{(k)}\mathbf{D}_{ii}h_i^{(k)} - \mathbf{h}^{(k)}_{row}\mathbf{S}\left(\mathbf{h}^{(k)}_{row}\right)^{\mathrm{T}}\right) \\
&= \sum_{k=1}^{r}\left(\mathbf{h}^{(k)}_{row}\mathbf{D}\left(\mathbf{h}^{(k)}_{row}\right)^{\mathrm{T}} - \mathbf{h}^{(k)}_{row}\mathbf{S}\left(\mathbf{h}^{(k)}_{row}\right)^{\mathrm{T}}\right) \\
&= \sum_{k=1}^{r}\mathbf{h}^{(k)}_{row}(\mathbf{D} - \mathbf{S})\left(\mathbf{h}^{(k)}_{row}\right)^{\mathrm{T}} \\
&= \sum_{k=1}^{r}\mathbf{h}^{(k)}_{row}\mathbf{L}\left(\mathbf{h}^{(k)}_{row}\right)^{\mathrm{T}} \\
&= \mathrm{Tr}(\mathbf{HLH}^{\mathrm{T}})
\end{aligned}$$

Here, $\mathbf{h}^{(k)}_{row}$ is the kth row vector of H, that is $\mathbf{h}^{(k)}_{row} = [h_1^{(k)}, h_2^{(k)}, \dots, h_n^{(k)}]$. $\mathbf{D}$ is a diagonal matrix, and $\mathbf{D}_{ii} = \sum_j \mathbf{S}_{ji}$, $\mathbf{L} = \mathbf{D} - \mathbf{S}$, which is called Laplacian matrix [39].

Then let $\alpha = 0$, $\beta < 0$ and $\mathbf{L} = \mathbf{D} - \mathbf{S}$, the optimization function (1) is a subspace NMF with LPP constraint.

4. Iterative GSC NMF algorithm

Let $\mathbf{V} = [v_{ij}] = \mathbf{HLH^T} \in \mathcal{R}^{\mathbf{r}\times\mathbf{r}}$, and we use the C_2 cost function, then the target function in formula (1) can be rewritten as

$$O_U = \sum_{i,j}\left(x_{ij}\log\frac{x_{ij}}{y_{ij}} - x_{ij} + y_{ij}\right) + \alpha\sum_{i,j}u_{ij} - \beta\sum_{i}v_{ii} \tag{4}$$

Observing $\mathbf{BH} \triangleq \mathbf{Y} = [y_{ij}]$, $\mathbf{U} = [u_{ij}] = \mathbf{B}^{\mathrm{T}}\mathbf{B}$ and $\mathbf{V} = [v_{ij}] = \mathbf{HLH}^{\mathrm{T}}$, we then have

$$\begin{aligned}
y_{ij} &= \sum_{k=1}^{r} b_{ik}h_{kj} \\
u_{ij} &= \sum_{k=1}^{d} b_{ki}b_{kj} \\
v_{ij} &= \sum_{k=1}^{n}\left(\sum_{l=1}^{n} h_{il}l_{lk}\right)h_{jk}
\end{aligned}$$

To solve the above optimization function, we use the auxiliary function [32] similar to the EM algorithm.

Definition 1. If function $G(h, h')$ satisfied $G(h, h') \ge F(h)$, $G(h, h) = F(h)$, then $G(h, h')$ is an auxiliary function of $F(h)$.

Based on Definition 1, we can obtain the following theorem.

Theorem 1. *If $G(h, h')$ is an auxiliary function of $F(h)$, then when updating $F(h)$ with $h^{t+1} = \arg\min_h G(h, h^t)$, $F(h)$ will be monotonous and non-incremental.*

The proof of Theorem 1 can be obtained from Definition 1 easily.

According to Theorem 1, $F(h^{t_1}) = F(h^t)$, if and only if h^t is a local minima of $G(h, h^t)$. If the target function F is differentiable and continuous at an interval of h^t, which also means $\nabla F(h^t) = 0$, then we can obtain a sequence of estimations when we use the iterative update rule in Theorem 1. Thus the target function F will converge to a local minima $h_{\min} = \arg\min_h F(h)$ with

$$F(h_{\min}) \le \cdots F(h^{t+1}) \le F(h^t) \le \cdots \le F(h^1) \le F(h^0)$$

Considering formula (4), the optimizations for $\mathbf{B}$ and $\mathbf{H}$ are non-convex in function $D(\mathbf{X} \parallel \mathbf{BH})$, we adopt alternate iterative method to solve $\mathbf{B}$ and $\mathbf{H}$. We use $D(\mathbf{H})$ to represent the function $D(\mathbf{X} \parallel \mathbf{BH})$ with respect to $\mathbf{H}$ when $\mathbf{B}$ is fixed, and $D(\mathbf{B})$ to represent the function $D(\mathbf{X} \parallel \mathbf{BH})$ with respect to $\mathbf{B}$ when $\mathbf{H}$ is fixed. Based on Theorem 1, our alternate iterative solution tries to find the corresponding auxiliary function of $D(\mathbf{H})$ and $D(\mathbf{B})$.

Theorem 2. *Function:*

$$G(\mathbf{H}, \mathbf{H}') = \sum_{i,j} x_{ij}\log x_{ij} - \sum_{i,j,k} x_{ij}\frac{b_{ik}h'_{kj}}{\sum_k b_{ik}h'_{kj}}\left(\log\left(b_{ik}h_{kj}\right) - \log\frac{b_{ik}h'_{kj}}{\sum_k b_{ik}h'_{kj}}\right) - \sum_{i,j} x_{ij} + \sum_{i,j} y_{ij} + \alpha\sum_{i,j} u_{ij} - \beta\sum_{i} v_{ii} \tag{5}$$

is an auxiliary function of $D(\mathbf{H})$. The proof is given in Appendix A.

According to Theorem 2, we then can minimize $D(\mathbf{H})$ with respect to $\mathbf{H}$ and use the following updating rule:

$$\mathbf{H}^{t+1} = \arg\min_{\mathbf{H}} G(\mathbf{H}, \mathbf{H}^t)$$

We can update the value of $\mathbf{H}$ by solving $\partial G(\mathbf{H}, \mathbf{H}')/\partial h_{mn} = 0$ for every subscripts of m and n.

The $\partial G(\mathbf{H}, \mathbf{H}')/\partial h_{mn} = 0$ can be decomposed as

$$\frac{\partial\sum_{i,j} x_{ij}\log x_{ij}}{\partial h_{mn}} = 0 \quad \text{and} \quad \frac{\partial\sum_{i,j} x_{ij}}{\partial h_{mn}} = 0,$$

that is

$$\begin{aligned}
\sum_{i,j} y_{ij} = \sum_{i,j,k} b_{ik}h_{kj} &\Rightarrow \frac{\partial\sum_{i,j} y_{ij}}{\partial h_{mn}} = \sum_{i} b_{im} \\
&\frac{\partial\sum_{i,j} u_{ij}}{\partial h_{mn}} = 0 \\
\sum_{i} v_{ii} = \sum_{i,k,l} h_{il}l_{lk}h_{ik} &\Rightarrow \frac{\partial\sum_{i} v_{ii}}{\partial h_{mn}} = \sum_{k} l_{nk}h_{mk} + \sum_{l} h_{ml}l_{\mathrm{ln}}
\end{aligned}$$

Based on the analysis in previous section, the projection matrix $\mathbf{L}$, which represents the constrained relations among the samples, is usually symmetrical, thus

$$\frac{\partial\sum_i v_{ii}}{\partial h_{mn}} = \sum\nolimits_k l_{nk}h_{mk} + \sum\nolimits_l h_{ml}l_{\mathrm{ln}} = 2\sum\nolimits_l l_{nl}h_{ml}$$

then

$$\frac{\partial G(\mathbf{H},\mathbf{H}')}{\partial h_{mn}} = -\sum_i x_{in}\frac{b_{im}\,h'_{mn}}{\sum_k b_{ik}h'_{kn}}\frac{1}{h_{mn}} + \sum_i b_{im} - 2\beta\sum_l l_{nl}h_{ml} \tag{6}$$

Let

$$\sum_i x_{in}\frac{b_{im}\,h'_{mn}}{\sum_k b_{ik}h'_{kn}} \triangleq A_{mn},\ \sum_{l\neq n} l_{nl}h_{ml} \triangleq C_{mn},\ \left(\sum_i b_{im} - 2\beta C_{mn}\right) \triangleq Q,$$

and setting formula (6) to zero,we can then obtain

$$h_{mn} = \begin{cases} \dfrac{Q \pm \sqrt{Q^2 - 8\beta l_{nn}A_{mn}}}{4\beta l_{nn}}, & \beta l_{nn} \neq 0 \\ \dfrac{A_{mn}}{Q}, & l_{nn} = 0,\ \beta \neq 0 \\ \dfrac{A_{mn}}{\sum_i b_{im}}, & \beta = 0 \end{cases}$$

The iterative update rule can be formulated as follows:
For **H** If β=0,

$$\mathbf{H} \leftarrow \frac{[\mathbf{H}]}{\overline{\left[\mathbf{B}^{\mathrm{T}}\mathbf{1}_{d\times n}\right]}}\bigcirc\left(\mathbf{B}^{\mathrm{T}}\frac{[\mathbf{X}]}{\overline{[\mathbf{BH}]}}\right) \tag{7}$$

If $\beta l_{nn} \neq 0$

$$\mathbf{C} \leftarrow \gamma\mathbf{B}^{\mathrm{T}}\mathbf{1}_{d\times n} - \frac{2}{\gamma}\beta\mathbf{H}\mathbf{F}^{\mathrm{T}} \tag{8}$$

$$\mathbf{A} \leftarrow 8\beta\left(\left[\mathbf{H}\right]\bigcirc\left(\mathbf{B}^{\mathrm{T}}\frac{[\mathbf{X}]}{\overline{[\mathbf{BH}]}}\right)\right)\mathbf{D} \tag{9}$$

$$\mathbf{H} \leftarrow \frac{\left[\mathbf{C} - \left[\,[\mathbf{C}]^2 - \mathbf{A}\right]^{1/2}\right]}{\overline{\left[4\beta\mathbf{1}_{r\times 1}\mathbf{m}\right]}} \tag{10}$$

If $l_{nn} = 0,\ \beta \neq 0$

$$\mathbf{A} \leftarrow [\mathbf{H}]\bigcirc\left(\mathbf{B}^{\mathrm{T}}\frac{[\mathbf{X}]}{\overline{[\mathbf{BH}]}}\right) \tag{11}$$

$$\mathbf{H} \leftarrow \frac{\mathbf{A}}{\gamma\mathbf{B}^{\mathrm{T}}\mathbf{1}_{d\times n} - \dfrac{2}{\gamma}\beta\mathbf{H}\mathbf{F}^{\mathrm{T}}} \tag{12}$$

For **B**

$$\mathbf{B} \leftarrow \frac{\gamma\,[\mathbf{B}]}{\overline{\left[\mathbf{1}_{d\times d}\mathbf{B}\right]}} \tag{13}$$

Here $\mathbf{X}\bigcirc\mathbf{Y}$ denotes the operation of Hadamard product, and $[\mathbf{X}]/\overline{[\mathbf{Y}]}$ denotes the operation of Hadamard division. $[\mathbf{X}]^a$ denotes the operation to a power for each element in matrix, $\mathbf{1}_{m\times n}$ denotes the $m \times n$ matrix with all the elements equalling to 1. $\mathbf{D} = [d_{ij}]$, if i=j, $d_{ii} = l_{ii}$ otherwise $d_{ij} = 0$. $\mathbf{F} = \mathbf{L} - \mathbf{D}$, and $m = [m_i]$ is a vector of $1 \times n$, $m_i = l_{ii}$.

When $\beta l_{nn} \neq 0$ in formula (8), γ is a non-negative value, which satisfy that $\forall\ m, n$, $(\gamma\sum_i b_{im} - (2/\gamma)\beta C_{mn}) > 0$ and , and $\gamma \geq 1$.

When $l_{mm} = 0, \beta \neq 0$ in formula (12), γ is a non-negative satisfying $\forall\ m, n, (\gamma\sum_i b_{im} - (2/\gamma)\beta C_{mn}) > 0$. The γ in formula (13) is set according to the formula (8)–(12).

In each iterative update of **B** and **H**, the value of γ remains constant for all the elements of **H** and **B**. We use γ to guarantee that the iterative update of formula (7)–(12) will always output non-negative values. At the beginning of each iteration for all the elements in **H**, we set γ=1 and then test whether γ can satisfy the constraints ($(\gamma\sum_i b_{im} - (2/\gamma)\beta C_{mn}) > 0$ and). If the current value of γ cannot satisfy the constraints, we multiply a factor larger than 1, e.g. 3, with γ until the new value of γ can satisfy the constraints. And then updating the elements in **H** and **B** with that γ. Normally, the value of γ will decrease after several iterations, and then the remainder iterations will keep using γ=1. With formula (13), we actually add a constraint of $\sum_i b_{im} = \gamma$ (γ=1 when β=0), and this constraint can guarantee: (1) there will not be some basis vectors (column vector in **B**) tending to all zero, while corresponding elements in **H** tend to infinite; (2) the scales between each basis vector will tend to similar.

To minimize $D(\mathbf{B})$ with respect to **B**, we can also construct $D(\mathbf{B})$'s auxiliary function $G(\mathbf{B},\mathbf{B}')$ and iterate with , we also solve $\partial G(\mathbf{B},\mathbf{B}')/\partial b_{mn} = 0$ and can obtain

$$b_{mn} = \frac{b'_{mn}\sum_j x_{mj}\dfrac{h_{nj}}{\sum_k b'_{mk}h_{kj}}}{\sum_j h_{nj} + \alpha\sum_j b_{mj}}$$

Thus the updating rule for **B** can be given as follows:

$$\mathbf{B} \leftarrow \frac{[\mathbf{B}]}{\overline{\left[\mathbf{1}_{d\times n}\mathbf{H}^{\mathrm{T}} + \alpha\mathbf{B}\mathbf{1}_{r\times r}\right]}}\bigcirc\left(\frac{[\mathbf{X}]}{\overline{[\mathbf{BH}]}}\mathbf{H}^{\mathrm{T}}\right) \tag{14}$$

Then the solution for the target function can be obtained by iterative executing formula (7)–(14), and if β=0, then set γ as 1, otherwise calculate γ with the method mentioned above.

In solution, our approach first randomly initializes the **B** and **H** as nonsingular matrixes, whose elements hold uniform distributions within [0, 1]. Let $\mathbf{BH} \triangleq \mathbf{Y} = [y_{ij}]$, the terminating condition for the iteration is

$$\max_{i,j}(\left|y_{ij}^{new} - y_{ij}^{old}\right|/y_{ij}^{old}) < \epsilon$$

Here, ϵ is a non-negative constant, ${y_{ij}}^{new}$ denotes the value of the current iteration, and ${y_{ij}}^{old}$ denotes the value of the previous iteration.

5. Experiments

In this section, we carry out experiments to evaluate the different NMF methods. Firstly, we show the results of the classification experiments to evaluate their discriminative ability. Then more specific analyses are presented to demonstrate the specific characteristic of different NMF methods. For PCA NMF, we show its superiority on the sparse basis and low reconstruction error, where LPP NMF shows the ability to capture the data manifold.

5.1. Discriminative analysis

To evaluate the different NMF methods, we carry out classification experiments to study their discriminative ability. The input data is the ORL face database [40], which contains 40 persons and 400 face images totally, each person corresponds to 10 grayscale face images with a resolution of 112 × 92. In our experiment, we adjust the resolution of images into 56 × 46 and normalize the grayscale into [0, 1]. For classification, divide the 400 face images into training set and test set. For each person, 5 images are used for training and the others are for test.

The comparison are taken between the PCA, Fisher LDA, LPP and their corresponding NMF methods. An additional method to compare is the ANMF. The implementation of ANMF is given by [22], which does not set α and β explicitly and can be found in

Statistics Toolbox of MATLAB.[1] Notice that Fisher LDA and Fisher NMF are supervised feature extraction methods, where the others are unsupervised. The parameter setting of all the NMF methods are shown in Table 2, where the parameters of the PCA NMF, Fisher NMF and LPP NMF are selected by the optimal grid search.

Here all the methods are regarded as feature extraction approaches, which means we project the training set and test set to the lower-dimensional space and then use the projected training samples to train a classifier for testing. For all NMF methods, firstly we learn the basis matrix **B** from the training set. Define:

$$\mathbf{B}^{+} = \left(\mathbf{B}^{\mathrm{T}}\mathbf{B}\right)^{-1}\mathbf{B}^{\mathrm{T}} \tag{15}$$

Then we can get the projection of each test sample x_{test} with $h_{test} = \mathbf{B}^{+}x_{test}$. Finally, we train the classifier with the projected training set and test on the projected test set. Here we use Softmax regression for multi-class classification. The classification results are shown in Table 3. To randomize the experiment, we extract different number of classes of images from the whole database to train and test. In each case, we set the reduced dimension r the same as the number of classes k for all the compared approaches except for Fisher LDA, where the reduced dimension of Fisher LDA is $k-1$. The showing results are all average value of 10 times of random experiment. Results tell that the proposed NMF methods give comparative results on the classification, where LPP NMF performs best.

5.2. Reconstruction study

We carry out quantitative experiments to compare the reconstruction errors on test data. Comparisons are made between the PCA and the different NMF methods. The reconstruction error is calculated as $\| x_{\text{reconstruct}} - x_{\text{original}} \|_F$, $\|\cdot\|_F$ is Frobenius norm. The results under different numbers of basis r are shown in Table 4. Denote the trained basis matrix of PCA as **W**. As its column basis vectors are orthogonal, the PCA's reconstruction instance can be calculated as $y = \mathbf{W}\mathbf{W}^{\mathbf{T}}x$, x is an instance test set. When reconstructing the test data using ANMF, NMF and PCA NMF approaches, we reconstruct x_{test} with $y_{test} = \mathbf{B}\mathbf{B}^{+}x_{test}$, where $\mathbf{B}^{+}$ is calculated by function (15).

Table 4 shows that PCA performs best on the low dimension of basis. It is obvious since PCA is optimal for reconstruction. However, high dimensional basis leads to overfitting on PCA. It means that the proposed NMF methods are more insensitive to the dimension of basis, since it performs better than PCA and NMF when the reduction dimension is high.

Considering the influence of the parameters, we compare the performance of different NMF methods under a fixed reduced dimension (r=100) as shown in Fig. 1. Here we only consider the influence of β since all of the NMF methods have a nonzero β. For PCA NMF which has another parameter α, we set α=0.1 in this experiment. Notice that the abscissa axis is the absolute value of β because the sign of β is different for different NMF methods. Fig. 1 shows the reconstruction error of different NMF methods when β varies from $1e-2$ to $1e+2$. We can find that the NMF methods outperform than NMF and PCA in most cases. All of the NMF methods get the lowest reconstruction error at a medium value of β. This is rational since the large β corresponding to stronger constraint. Then the performance of the reconstruction will be influenced, whether on training set or test set.

[1] http://www.mathworks.com/help/stats/nnmf.html

Table 2
Parameters setting for different NMF methods.

Parameter	NMF	ANMF	PCA NMF	Fisher NMF	LPP NMF
α	0	/	1	0	0
β	0	/	1	1	−1

Table 3
Classification results on ORL face database.

Class	5	10	15	20	25	30	Average
PCA	0.9160	0.8960	0.9053	0.9096	0.9144	0.9127	0.9090
Fisher LDA	0.9160	0.9420	**0.9413**	0.9230	0.8928	0.8860	0.9169
LPP	0.9000	0.8720	0.8800	0.8780	0.8400	0.7831	0.8589
ANMF	0.9230	0.9160	0.9120	0.8770	0.8888	0.8680	0.8975
PCA NMF	0.9400	0.9220	0.9307	0.9150	0.8984	0.8880	0.9157
Fisher NMF	0.9560	**0.9440**	0.9240	0.8990	0.9032	0.8870	0.9189
LPP NMF	**0.9680**	0.9420	0.9213	**0.9200**	**0.9088**	**0.9007**	**0.9268**

Table 4
Reconstruction error under different number of basis in test data set.

# Basis	40	60	80	100	120	140	Average
PCA	**10.1649**	**9.0989**	8.8058	8.5776	8.3821	8.1978	8.8712
NMF	10.2665	9.3003	9.0135	8.5381	8.2878	8.1993	8.9343
PCA NMF	10.2765	9.2718	8.7390	8.3750	8.0934	8.1571	8.8188
Fisher NMF	10.1801	9.2704	9.0105	**8.2530**	8.0730	8.0104	8.7996
LPP NMF	10.2497	9.4847	**8.6515**	8.4089	**8.0551**	**7.6079**	**8.7560**

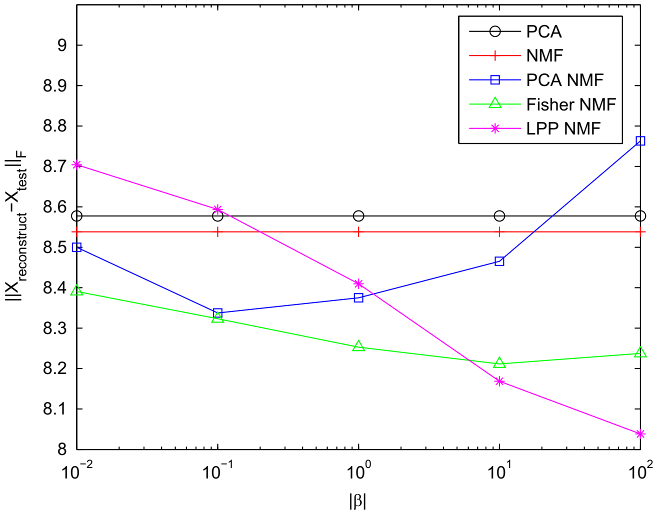


Fig. 1. Reconstruction error under different β in test data set.

5.3. Analysis on basis matrix

The visualization of basis matrix can give an intuitive sense about what the methods learned the input data. Fig. 2 shows the 64 dimension basis obtained from the PCA, ANMF, NMF and PCA NMF. We reshape the column basis vector as a 56 × 46 grayscale image and use darker gray levels to represent the larger values in the basis vectors. In the basis vectors of ANMF, NMF and PCA NMF, the basis images are sorted descended with respect to v_{ii} ($[v_{ii}] = \mathbf{H}\mathbf{H}^{\mathbf{T}}$) from left to right, up to down in Fig. 2. The value of v_{ii} can represent the importance of the basis vector, a larger value of v_{ii} will indicate a more important basis vector. The basis vector images of PCA are also

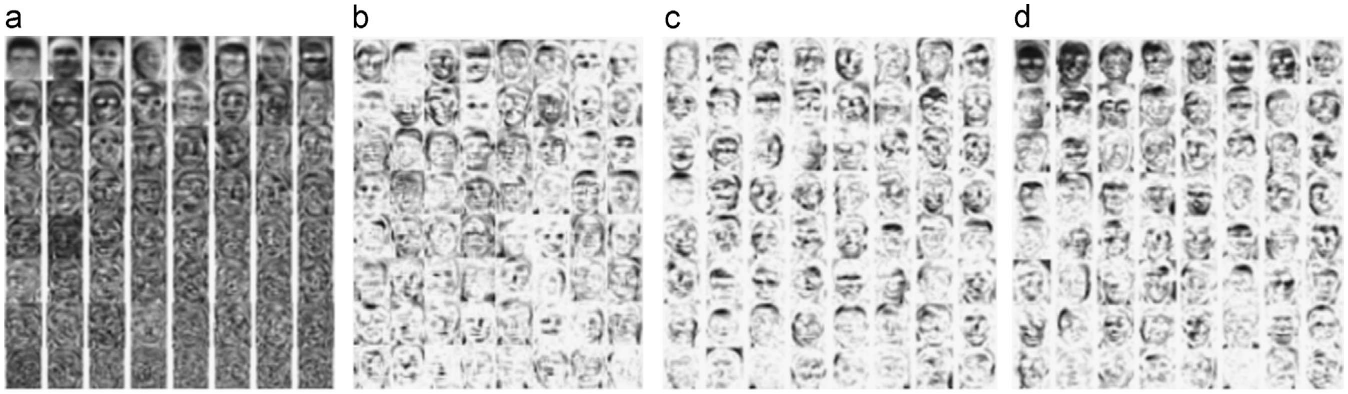


Fig. 2. Basis vectors represented as images for the training samples of ORL data set. (a) PCA. (b) ANMF. (c) NMF. (d) PCA NMF.

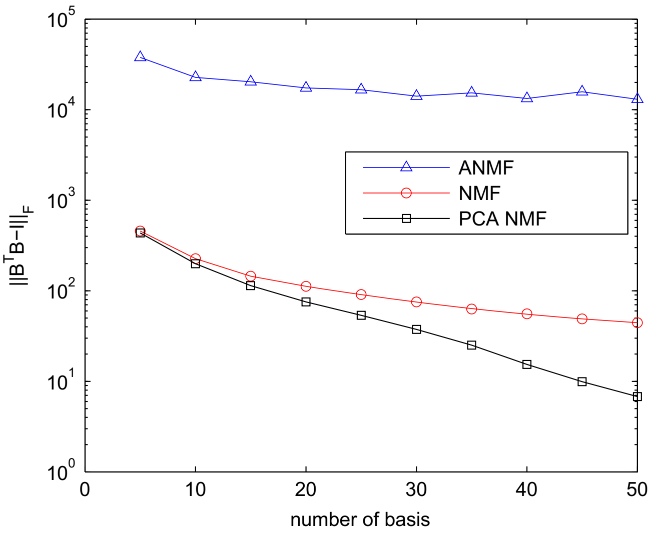


Fig. 3. Orthogonality of different NMF methods, logarithmic coordinates is used in Y-axis.

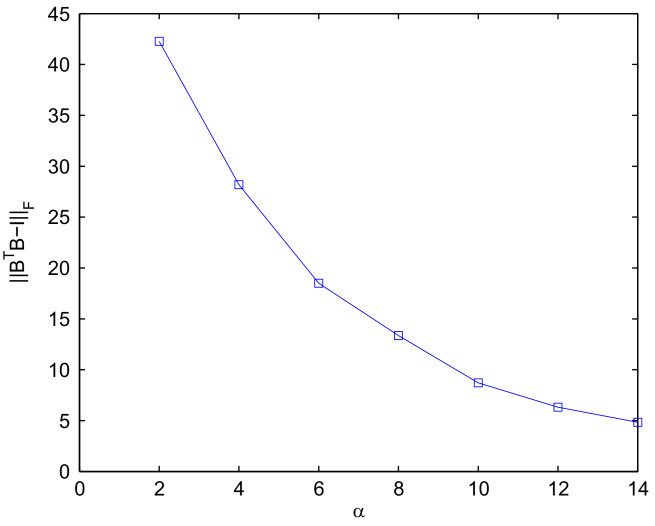


Fig. 4. Orthogonality of PCA NMF under different α, basis number is 25.

sorted descended based on their eigenvalues. We can observe that the basis images of both NMF and PCA NMF contain more elements close to zero (denoted with light color in the figures), and thus more sparse than the basis vectors of PCA. The PCA NMF is adding non-negative constraint to the basic PCA, and it can enforce the sparsity while still being able to represent the most important basis vectors capturing the largest variance, which means that PCA NMF is better to learn the part-based representation. That is the top lines of the basis images in both PCA and PCA NMF appear to have large areas with dark pixels.

We also consider the orthogonality of the basis obtained by each method, we use $\| \mathbf{B}^T\mathbf{B} - \mathbf{I} \|_F$ to measure the orthogonality. The results are shown in Fig. 3, as PCA can always satisfy $\| \mathbf{B}^T\mathbf{B} - \mathbf{I} \|_F = 0$, we do not draw the curve of PCA in this figure. The results show PCA NMF can achieve much smaller value of $\| \mathbf{B}^{\mathbf{TB}} - \mathbf{I} \|_F$, which demonstrates the orthogonality of basis.

As the orthogonality of basis are controlled by the term of $\sum_{ij} u_{ij}$, the value of α will affect the orthogonality of basis directly. We then consider the relation between α and $\| \mathbf{B}^{\mathbf{TB}} - \mathbf{I} \|_F$ in Fig. 4. As can be observed the orthogonality of the basis increases with α.

5.4. *Analysis on dimensionality reduced projection*

As we mentioned before, the **H** can be regarded as the projection of the original data which has a lower dimension. When we project the input data to a two-dimensional space, we can plot the projection samples to visualize the dimension reduction. In the dimensional reduction experiment, we use the face sculpture data set from [41] as our input data. The high dimensional images intrinsically lie on a low dimensional manifold corresponds to the viewpoints and lighting angles. We use LPP NMF to reduce the dimensionality, which keeps the local invariance of the input data at the same time. For the convenient of visualization, the face images are mapped to a two-dimensional space as shown in Fig. 5. The result shows that the projected points reveal the intrinsic manifold of the face images. More specifically, the left points reveal the continuous change of the images with right cheek, while the bottom points reveal the change of the images with left cheek. The middle points correspond to the front view of face. It means that LPP NMF can capture the underlying low dimensional characters of the high dimensional face data with its consistency of local neighborhood.

6. Conclusion

We have presented a general subspace constrained non-negative factorization (GSC NMF) framework, which can induce almost most of the subspace constraints into a unified NMF optimization function. An iterative optimization algorithm to solve the GSC NMF is also proposed. The experimental results show that GSC NMF framework and its iterative optimization algorithm can achieve

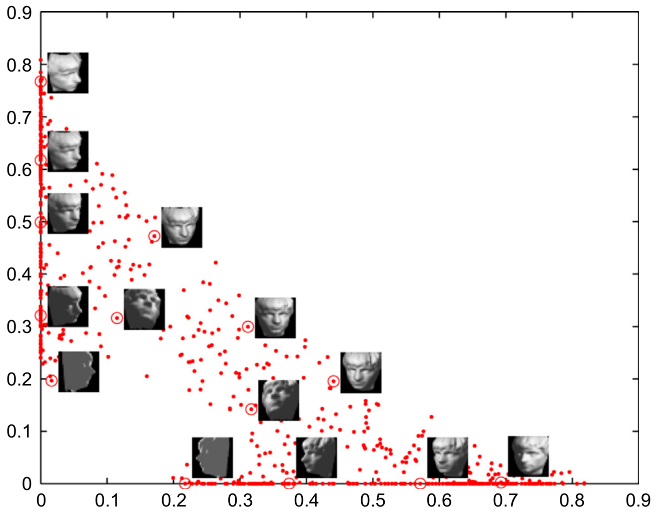


Fig. 5. Dimension reduction results of face sculpture data set with LPP NMF. In this experiment, $\mathbf{S}_{ij}^2$ is used as the similarity matrix and k=6. The data set used in experiment is a set of vectors with 1024 dimensions, and contains 698 columns. Each column represents a face sculpture image with a resolution of 32 × 32. The face sculpture images in this figure are corresponding to their nearby data points with circles.

better performance in data representation than the ordinary NMF approach and PCA approach.

Acknowledgment

This research is based upon work supported in part by National Science Foundation of China (61173123, 61203324), Natural Science Foundation of Zhejiang (LR13F030003), and the Open Research Project of the State Key Laboratory of Industrial Control Technology, Zhejiang University, China (No. ICT1502).

Appendix A. Proof of Theorem 2

Obviously, $G(\mathbf{H},\mathbf{H}) = D(\mathbf{H})$, we then need to proof $G(\mathbf{H},\mathbf{H}') \geq D(\mathbf{H})$. As function $-\log(\sum_k b_{ik}h_{kj})$ is convex, assuming $\sum_k \mu_{ijk} = 1,\ \forall\, i, j$, then

$$-\log\left(\sum_k b_{ik}h_{kj}\right) = -\log\left(\sum_k \mu_{ijk}\frac{b_{ik}h_{kj}}{\mu_{ijk}}\right)$$

$$\leq -\sum_k \mu_{ijk}\log\frac{b_{ik}h_{kj}}{\mu_{ijk}} \tag{A.1}$$

let $\mu_{ijk} = b_{ik}h'_{kj}/\sum_k b_{ik}h'_{kj}$, then formula (A.1) can be rewritten as

$$-\log(\sum_k b_{ik}h_{kj}) \leq -\sum_k \frac{b_{ik}h'_{kj}}{\sum_k b_{ik}h'_{kj}}(\log\left(b_{ik}h_{kj}\right) - \log\frac{b_{ik}}{h'_{kj}}\sum_k b_{ik}h'_{kj}) \tag{A.2}$$

Comparing formulas (4), (5) and (A.2), we can obtain $G(\mathbf{H},\mathbf{H}') \geq D(\mathbf{H})$, thus $G(\mathbf{H},\mathbf{H}')$ is an auxiliary function of $D(\mathbf{H})$.

References

[1] M.A. Turk, A.P. Pentland, Face recognition using eigenfaces, Maui, HI, 1991, pp. 586–591.

[2] W. Zhao, R. Chellappa, P.J. Phillips, A. Rosenfeld, Face recognition: a literature survey, ACM Comput. Surv. 35 (4) (2003) 399–458.

[3] X. Li, Y. Pang, Deterministic column-based matrix decomposition, IEEE Trans. Knowl. Data Eng. 22 (1) (2010) 145–149.

[4] H. Yan, J. Lu, X. Zhou, Y. Shang, Multi-feature multi-manifold learning for single-sample face recognition, Neurocomputing 143 (2014) 134–143.

[5] Y. Pang, X. Jiang, X. Li, J. Pan, Efficient object detection by prediction in 3D space http://dx.doi.org/10.1016/j.sigpro.2014.08.039, September 2014.

[6] Y. Pang, K. Zhang, Y. Yuan, K. Wang, Distributed object detection with linear svms http://dx.doi.org/10.1109/TCYB.2014.2301453, 2014.

[7] X. Jiang, Y. Pang, J. Pan, X. Li, Flexible sliding windows with adaptive pixel strides http://dx.doi.org/10.1016/j.sigpro.2014.08.004, August 2014.

[8] B.C. Moore, Principal component analysis in linear systems – controllability, observability, and model reduction, IEEE Trans. Autom. Control AC 26 (1) (1981) 17–32.

[9] M. Kirby, L. Sirovich, Application of the Karhunen–Loeve procedure for the characterization of human faces, IEEE Trans. Pattern Anal. Mach. Intell. 12 (1) (1990) 103–108.

[10] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, Eigenfaces vs. fisherfaces: recognition using class specific linear projection, IEEE Trans. Pattern Anal. Mach. Intell. 19 (7) (1997) 711–720.

[11] A.M. Martinez, A.C. Kak, Pca versus lda, IEEE Trans. Pattern Anal. Mach. Intell. 23 (2) (2001) 228–233.

[12] X. He, P. Niyogi, Locality preserving projections, in: NIPS, 2003.

[13] Y. Choi, S. Ozawa, M. Lee, Incremental two-dimensional kernel principal component analysis, Neurocomputing 134 (0) (2014) 280–288.

[14] Y. Pang, X. Li, Y. Yuan, Robust tensor analysis with l1-norm, IEEE Trans. Circuits Syst. Video Technol. (2010) 172–178.

[15] X. Li, Y. Pang, Y. Yuan, L1-norm-based 2dpca, IEEE Trans. Syst. Man Cybern. Part B: Cybern. (2010) 1170–1175.

[16] Y. Pang, Y. Yuan, K. Wang, Learning optimal spatial filters by discriminant analysis for brain–computer-interface, Neurocomputing 77 (1) (2012) 20–27.

[17] D. Tao, X. Li, X. Wu, S.J. Maybank, General tensor discriminant analysis and gabor features for gait recognition, IEEE Trans. Pattern Anal. Mach. Intell. 29 (10) (2007) 1700–1715.

[18] Y. Pang, Z. Ji, P. Jing, X. Li, Ranking graph embedding for learning to rerank, IEEE Trans. Neural Netw. Learn. Syst. 24 (8) (2013) 1292–1303.

[19] Y. Pang, S. Wang, Y. Yuan, Learning regularized LDA by clustering http://dx.doi.org/10.1109/TNNLS.2014.2306844, 2014.

[20] W. Xu, X. Liu, 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 information retrieval, SIGIR '03, ACM, New York, NY, USA, 2003, pp. 267–273.

[21] V.P. Pauca, F. Shahnaz, M.W. Berry, R.J. Plemmons, Text mining using non-negative matrix factorizations., in: SDM, SIAM, 2004.

[22] M.W. Berry, M. Browne, A.N. Langville, V.P. Pauca, R.J. Plemmons, Algorithms and applications for approximate nonnegative matrix factorization, Comput. Stat. Data Anal. 52 (1) (2007) 155–173.

[23] Y. Chen, M. Rege, M. Dong, J. Hua, Non-negative matrix factorization for semi-supervised data clustering, Knowl. Inf. Syst. 17 (3) (2008) 355–379.

[24] S. Zafeiriou, M. Petrou, Nonlinear non-negative component analysis algorithms, IEEE Trans. Image Process. 19 (4) (2010) 1050–1066.

[25] A. Holzapfel, Y. Stylianou, Musical genre classification using nonnegative matrix factorization-based features, IEEE Trans. Audio Speech Lang. Process. 16 (2) (2008) 424–434.

[26] B. Schuller, F. Weninger, M. Wollmer, Y. Sun, G. Rigoll, Non-negative matrix factorization as noise-robust feature extractor for speech recognition, in: 2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), IEEE, Dallas, Texas, USA, 2010, pp. 4562–4565.

[27] H.-Y. Shum, H. Zhang, Tao Feng, Stan Z. Li, Local non-negative matrix factorization as a visual representation, in: Proceedings of the 2nd International Conference on Development and Learning, ICDL '02, IEEE Computer Society, Washington, DC, USA, 2002, pp. 178–183.

[28] C. Ding, T. Li, W. Peng, H. Park, Orthogonal nonnegative matrix t-factorizations for clustering, in: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2006, pp. 126–135.

[29] Z. Li, X. Wu, H. Peng, Nonnegative matrix factorization on orthogonal subspace, Pattern Recognit. Lett. 31 (9) (2010) 905–911.

[30] P.O. Hoyer, P. Dayan, Non-negative matrix factorization with sparseness constraints, J. Mach. Learn. Res. 5 (2004) 1457–1469.

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

[32] D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, in: In NIPS, MIT Press, Vancouver, British Columbia, Canada, 2001, pp. 556–562.

[33] S.Z. Li, X. Hou, H. Zhang, Q. Cheng, Learning spatially localized, parts-based representation, in: CVPR2001, 2001, pp. 207–212.

[34] Y. Wang, Y. Jia, Fisher non-negative matrix factorization for learning local features, in: Proceedings of Asian Conference on Computer Vision, 2004.

[35] S. Zafeiriou, A. Tefas, I. Buciu, I. Pitas, Exploiting discriminant information in nonnegative matrix factorization with application to frontal face verification, IEEE Trans. Neural Netw./A Publ. IEEE Neural Netw. Counc. 17 (3) (2006) 683–695.

[36] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.

[37] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, in: NIPS, 2001, pp. 585–591.

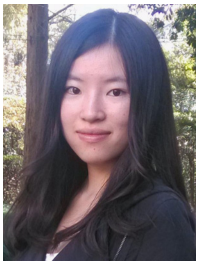
[38] D. Cai, X. He, J. Han, T.S. Huang, Graph regularized nonnegative matrix factorization for data representation, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1548–1560.

[39] F.R.K. Chung, Spectral Graph Theory, Regional Conference Series in Mathematics. 92. Providence, RI: American Mathematical Society (AMS), xi, 1997.

[40] F.S. Samaria, A.C. Harter, Parameterisation of a stochastic model for human face identification, in: 1994 Proceedings of the Second IEEE Workshop on Applications of Computer Vision, IEEE, Sarasota, FL, USA, 1994, pp. 138–142.

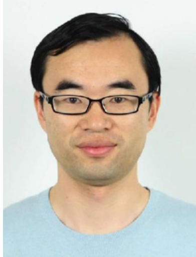
[41] J.B. Tenenbaum, V. de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.

Yong Liu received his B.S. degree in computer science and engineering from Zhejiang University in 2001, and the Ph.D. degree in computer science from Zhejiang University in 2007. He is currently an associate professor in the Institute of Cyber-Systems and Control, Department of Control Science and Engineering, Zhejiang University. He has published more than 30 research papers in machine learning, computer vision, information fusion, robotics. His latest research interests include machine learning, robotics vision, information processing and granular computing. He is the corresponding author of this paper.

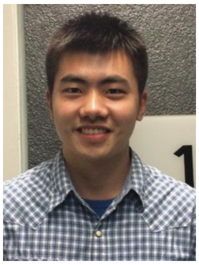


Yiyi Liao received her B.S. degree in automation from Xi'an Jiaotong University in 2013. She is currently a master student in the Institute of Cyber-Systems and Control, Department of Control Science and Engineering, Zhejiang University. Her research interests include machine learning and computer vision.

Liang Tang graduated in 2009 from Department of Automation, Zhejiang University, China, with a Ph.D. degree in Control Theory & Control Engineering. He is currently an engineer in China Ship Development and Design Center (CSDDC), focusing on practical application of artificial intelligence to ship information engineering.



Feng Tang is a senior researcher in Hewlett-Packard Laboratories, Palo Alto since 2009. He obtained Ph.D. degree in University of California, Santa Cruz in December 2008, master's degree and bachelor's degree in State Key Lab of CAD&CG, Zhejiang University in 2004 and 2001 respectively. He has published more than 40 papers in computer vision, machine learning and multimedia. He received the best paper award for Multimedia Modeling Conference in 2011 and best paper run up in the International Conference on Internet Multimedia Computing and Service in 2012.



Weicong Liu received the B.S. degree in control science and engineering from Zhejiang University, Hangzhou, in 2013. He is currently a Ph.D. student in the computer science and engineering department, in the Chinese University of Hong Kong, Shatin, Hong Kong. His research interests include machine learning and financial engineering.