

## Research Article

# Neighborhood Preserving Convex Nonnegative Matrix Factorization

**Jiang Wei, Li Min, and Zhang Yongqing**

*School of Mathematics, Liaoning Normal University, Dalian 116029, China*

Correspondence should be addressed to Jiang Wei; [swxxjw@aliyun.com](mailto:swxxjw@aliyun.com)

Received 27 September 2013; Accepted 6 January 2014; Published 25 February 2014

Academic Editor: Chung-Hao Chen

Copyright © 2014 Jiang Wei et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The convex nonnegative matrix factorization (CNMF) is a variation of nonnegative matrix factorization (NMF) in which each cluster is expressed by a linear combination of the data points and each data point is represented by a linear combination of the cluster centers. When there exists nonlinearity in the manifold structure, both NMF and CNMF are incapable of characterizing the geometric structure of the data. This paper introduces a neighborhood preserving convex nonnegative matrix factorization (NPCNMF), which imposes an additional constraint on CNMF that each data point can be represented as a linear combination of its neighbors. Thus our method is able to reap the benefits of both nonnegative data factorization and the purpose of manifold structure. An efficient multiplicative updating procedure is produced, and its convergence is guaranteed theoretically. The feasibility and effectiveness of NPCNMF are verified on several standard data sets with promising results.

## 1. Introduction

This nonnegative matrix factorization (NMF) [1, 2] has been widely used in information retrieval, computer vision, pattern recognition, and DNA gene expressions [3, 4]. NMF decomposes the data matrix as the product of two matrices that possess only nonnegative elements. It has been stated by many researchers that there are a lot of favorable properties for such a decomposition over other similar decompositions, such as PCA. One of the most useful properties of NMF is that it usually leads to parts-based representation because it allows only additive, not subtractive, combinations. Such a representation encodes much of the data making them easy to interpret. NMF can be traced back to 1970s and has been studied extensively by Paatero and Tapper [5]. The work of Lee and Seung [1] brought much attention to NMF in machine learning and data mining fields. Since then, various extensions and variations of NMF have been proposed. Li et al. [4] proposed local nonnegative matrix factorization (LNMF) algorithm which imposes extra constraints to the cost function to get more localized and parts-based image features. Hoyer [6, 7] employed sparsity constraints to improve local data representation, while nonnegative tensor

factorization was studied in [8, 9] by Hazan et al. to handle the data encoded as high-order tensors. All the methods mentioned above are unsupervised, Wang et al; [10] and Zafeiriou et al. [11] proposed independently the Fisher-NMF, which was further studied by Kotsia et al. [12], by adding an additional constraint seeking to maximize the between-class scatter and minimize the within-class scatter in the subspace spanned by the bases.

One of the most important drawbacks of NMF and its variants is the fact that these methods have to be performed in the original feature space of the data points, so that it can not be kernelized and the powerful idea of the kernel method cannot be applied to NMF. Ding et al. [13] proposed convex nonnegative factorization (CNMF) that strives to address the problems while inheriting all the strengths of the above NMF method, which models each cluster as a linear combination of the data points and each data point as a linear combination of the cluster centers. The major advantage of CNMF over NMF is that it can be performed on any data representations, either in the original space or RKHS.

Recently, there has been a lot of interest in geometrically motivated approaches to data analysis in high dimensional spaces. When the data lives on or close to a nonlinear

low dimensional manifold which is embedded in the high dimensional ambient space [14, 15], Euclidean distance is incapable of charactering the geometric structure of the data and hence traditional methods like NMF and CNMF no longer work well. Both CNMF and NMF do not exploit the geometric structure of the data, which assume that the data points are sampled from a Euclidean space. To address this problem, Cai et al. proposed a graph regularized NMF (GNMF) [16] and locally consistent concept factorization (LCCF) [17], which assume that the nearby data points are likely to be in the same cluster, that is, cluster assumption [18, 19]. The Euclidean and manifold geometry are unified through a regularization framework, which has a better interpretation from manifold perspective.

In this paper, we introduce a novel matrix factorization algorithm, called neighborhood preserving convex nonnegative matrix factorization (NPCNMF) which is based on the assumption that if a data point can be reconstructed from its neighbors in the input space, then it can be reconstructed from its neighbors by the same reconstruction coefficients in the low dimensional subspace, that is, local linear embedding assumption [20]. NPCNMF not only inherits the advantages of CNMF, for example, nonnegativity, but also overcomes the shortcomings of CNMF, that is, Euclidean assumption. We also propose a multiplicative algorithm to efficiently solve the corresponding optimization problem and its convergence is theoretically guaranteed.

The rest of this paper is organized as follows. In Section 2, we briefly review NMF and CNMF. In Section 3, we introduce our algorithm and provide the proof of the convergence of the proposed algorithm. Experiments on three benchmark face recognition data sets are demonstrated in Section 4. Finally, we draw a conclusion and provide suggestions for future work.

## 2. A Review of NMF and CNMF

Nonnegative matrix factorization (NMF) factorizes the data matrix into one nonnegative basis matrix and one nonnegative coefficient matrix. Given a nonnegative data  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N] \in \mathbb{R}_+^{m \times N}$ , each column of  $\mathbf{X}$  is a sample point. NMF aims to find two nonnegative matrices  $\mathbf{U} \in \mathbb{R}_+^{m \times r}$  and  $\mathbf{V} \in \mathbb{R}_+^{N \times r}$  which minimize the following objective function:

$$\mathcal{J}_{\text{NMF}} = \|\mathbf{X} - \mathbf{UV}^T\|_F^2 \quad \text{s.t. } \mathbf{U} \geq 0, \mathbf{V} \geq 0, \quad (1)$$

where  $\|\cdot\|_F$  is Frobenius norm.

The objective function is joint optimization problem of basis matrix  $\mathbf{U}$  and coefficient matrix  $\mathbf{V}$ . Although it is not jointly convex to  $\mathbf{U}$  and  $\mathbf{V}$ , it is convex with respect to each of them when the other one is fixed. Therefore, it is unrealistic to expect an algorithm to find the global minimum of  $\mathcal{J}_{\text{NMF}}$ .

To optimize the objective, Lee and Seung [2] presented an iterative multiplicative updating algorithm as follows:

$$\begin{aligned} u_{ik}^{t+1} &= u_{ik}^t \frac{(\mathbf{XV})_{ik}}{(\mathbf{UV}^T\mathbf{V})_{ik}}, \\ v_{jk}^{t+1} &= v_{jk}^t \frac{(\mathbf{X}^T\mathbf{U})_{jk}}{(\mathbf{VU}^T\mathbf{U})_{jk}}. \end{aligned} \quad (2)$$

It is proved that the above updated steps will find a local minimum of the objective function in (1).

In reality, we have  $r \ll m$  and  $r \ll N$ . Thus, NMF essentially tries to find a compressed approximation of the original matrix,  $\mathbf{X} \approx \mathbf{UV}^T$ . We can view this approximation column as follows:

$$\mathbf{x}_j = \sum_{k=1}^r \mathbf{u}_k v_{jk}, \quad (3)$$

where  $\mathbf{u}_k$  is the  $k$ th column vector of  $\mathbf{U}$ . Thus, each data vector  $\mathbf{x}_j$  is approximated by a linear combination of the columns of  $\mathbf{U}$ , weighted by the components of  $\mathbf{V}$ . One limitation of NMF is that the nonnegative requirement is not applicable to applications where the data involves negative number. The second is that it is not clear how to effectively perform NMF in the transformed data space so that the powerful kernel method can be applied. To overcome the problem, Ding et al. [13] proposed a convex nonnegative matrix factorization (CNMF) algorithm where nonnegative and mixed-sign data matrices are applied. CNMF claims that each base can be characterized by a linear combination of the entire data points while each data point can be approximated by a linear combination of all the bases. Translating the statements into mathematics, we have

$$\mathbf{u}_k = \sum_{j=1}^N \mathbf{x}_j w_{jk}, \quad (4)$$

$$\mathbf{x}_j = \sum_{k=1}^r \mathbf{u}_k v_{jk}, \quad (5)$$

where  $w_{jk}$  is a nonnegative weight in which data point  $\mathbf{x}_j$  is related to  $k$ th base and  $v_{jk}$  is a nonnegative projection value of  $\mathbf{x}_j$ . Replacing  $\mathbf{u}_k$  in (5) with (4), we have

$$\mathbf{x}_j = \sum_{k=1}^r \sum_{j=1}^N \mathbf{x}_j w_{jk} v_{jk}. \quad (6)$$

We form the  $m \times N$  data matrix  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$  using the feature vector of data point  $\mathbf{x}_i$  as the  $i$ th column, the  $N \times r$  matrix  $\mathbf{W} = [w_{ij}]$  using bases  $w_{ij}$ , and  $N \times r$  projection matrix  $\mathbf{V} = [v_{ij}]$  using the projection values  $v_{ij}$ . From (6), we have

$$\mathbf{X} = \mathbf{XWV}^T. \quad (7)$$

Equation (7) can be interpreted as the approximation of the original data set. Minimizing the squared error  $\mathcal{J}$  and its approximation [13]

$$\mathcal{J}_{\text{CNMF}} = \|\mathbf{X} - \mathbf{XWV}^T\|_F^2, \quad (8)$$

where  $\mathbf{X} \in \mathbb{R}^{m \times N}$ ,  $\mathbf{W} \in \mathbb{R}^{N \times r}$ ,  $\mathbf{V} \in \mathbb{R}^{N \times r}$ . The matrices  $\mathbf{W}$  and  $\mathbf{V}$  are updated iteratively until convergence using the following rules:

$$\begin{aligned} w_{ik} &= w_{ik} \sqrt{\frac{(\mathbf{Y}^+ \mathbf{V})_{ik} + (\mathbf{Y}^- \mathbf{W} \mathbf{V}^T \mathbf{V})_{ik}}{(\mathbf{Y}^- \mathbf{V})_{ik} + (\mathbf{Y}^+ \mathbf{W} \mathbf{V}^T \mathbf{V})_{ik}}}, \\ v_{ik} &= v_{ik} \sqrt{\frac{(\mathbf{Y}^+ \mathbf{W})_{ik} + (\mathbf{V} \mathbf{W}^T \mathbf{Y}^- \mathbf{W})_{ik}}{(\mathbf{Y}^- \mathbf{W})_{ik} + (\mathbf{V} \mathbf{W}^T \mathbf{Y}^+ \mathbf{W})_{ik}}}, \end{aligned} \quad (9)$$

where  $\mathbf{Y} = \mathbf{X}^T \mathbf{X}$  and the matrix  $\mathbf{Y}^+$  and  $\mathbf{Y}^-$  are given by

$$\begin{aligned} \mathbf{Y}_{ik}^+ &= \frac{1}{2} (|\mathbf{Y}_{ik}| + \mathbf{Y}_{ik}), \\ \mathbf{Y}_{ik}^- &= \frac{1}{2} (|\mathbf{Y}_{ik}| - \mathbf{Y}_{ik}), \end{aligned} \quad (10)$$

respectively.

### 3. Neighborhood Preserving Convex Nonnegative Matrix Factorization

In this section, we introduce our neighborhood preserving convex nonnegative matrix factorization method, which takes the local linear embedding constraint as an additional requirement. The method presented in this paper is fundamentally motivated from the neighborhood preserving embedding.

*3.1. The Objective Function.* Many real world data are actually sampled from a nonlinear low dimensional manifold which is embedded in the high dimensional ambient space. Both NMF and CF perform the factorization in the Euclidean space. They fail to discover the local geometrical structure of the data space, which is essential to the clustering problem. NPE aims at preserving the local manifold structure. Specifically, for each data point, it is represented as a linear combination of the neighboring data points and the combination coefficients are specified in the weight matrix. We can find an optimal embedding such that the combination coefficients can be preserved in the low dimensional subspace.

For each data point, we find its  $k$  nearest neighbors. And we can characterize the local geometrical structure by linear coefficients that reconstruct each data point from its neighbors. The reconstruction coefficients are computed by the following objective function:

$$\begin{aligned} \min \quad & \left\| \mathbf{x}_i - \sum_{\mathbf{x}_j \in \mathcal{N}_k(\mathbf{x}_i)} w_{ij} \mathbf{x}_j \right\|^2 \\ \text{s.t.} \quad & \sum_{\mathbf{x}_j \in \mathcal{N}_k(\mathbf{x}_i)} w_{ij} = 1, \end{aligned} \quad (11)$$

and  $w_{ij} = 0$  if  $\mathbf{x}_j \notin \mathcal{N}_k(\mathbf{x}_i)$ , where  $\mathcal{N}_k(\mathbf{x}_i)$  denotes the  $k$  nearest neighborhood of  $\mathbf{x}_i$ .

Then  $\mathbf{v}_i$ ,  $1 \leq i \leq n$  in the dimensionality reduced space can be preserved by minimizing

$$\begin{aligned} \min \left\| \mathbf{v}_i - \sum_{\mathbf{x}_j \in \mathcal{N}_k(\mathbf{x}_i)} w_{ij} \mathbf{v}_j \right\|^2 &= \text{tr}(\mathbf{V}^T (\mathbf{I} - \mathbf{W}) (\mathbf{I} - \mathbf{W}) \mathbf{V}) \\ &= \text{tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}), \end{aligned} \quad (12)$$

where  $\text{tr}(\cdot)$  denotes the trace of a matrix,  $\mathbf{I} \in \mathbb{R}^{n \times n}$  is an identity matrix, and  $\mathbf{L} = (\mathbf{I} - \mathbf{W})(\mathbf{I} - \mathbf{W})$ . By minimizing (12), we essentially try to formalize our intuition that if a data point can be represented from its neighbors in the original space, then it can be represented from its neighbors by the same combination coefficients in the dimensionality reduced space.

With the neighborhood preserving constraint, CNMF incorporates (8) and minimizes the objective function as follows:

$$\mathcal{F} = \left\| \mathbf{X} - \mathbf{X} \mathbf{W} \mathbf{V}^T \right\|^2 + \lambda \text{tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}), \quad (13)$$

where  $\lambda$  is a positive regularization parameter controlling the contribution of the additional constraint. We call (13) neighborhood preserving convex nonnegative matrix factorization (NPCNMF). Let  $\lambda = 0$ ; (13) degenerates to the original CNMF.

*3.2. The Algorithm.* We introduce an iterative algorithm to find a local minimum for the optimization problem. By defining  $\mathbf{K} = \mathbf{X}^T \mathbf{X}$  and using the matrix properties  $\|\mathbf{A}\|^2 = \text{tr}(\mathbf{A}^T \mathbf{A})$ ,  $\text{tr}(\mathbf{A} \mathbf{B}) = \text{tr}(\mathbf{B} \mathbf{A})$ , and  $\text{tr}(\mathbf{A}) = \text{tr}(\mathbf{A}^T)$ , we can rewrite the objective function  $\mathcal{F}$  as follows:

$$\begin{aligned} \mathcal{F} &= \text{tr} \left( (\mathbf{X} - \mathbf{X} \mathbf{W} \mathbf{V}^T)^T (\mathbf{X} - \mathbf{X} \mathbf{W} \mathbf{V}^T) \right) + \lambda \text{tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}) \\ &= \text{tr} \left( (\mathbf{I} - \mathbf{W} \mathbf{V}^T)^T \mathbf{K} (\mathbf{I} - \mathbf{W} \mathbf{V}^T) \right) + \lambda \text{tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}) \\ &= \text{tr}(\mathbf{K}) - 2 \text{tr}(\mathbf{V} \mathbf{W}^T \mathbf{K}) + \text{tr}(\mathbf{V} \mathbf{W}^T \mathbf{K} \mathbf{W} \mathbf{V}^T) \\ &\quad + \lambda \text{tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}). \end{aligned} \quad (14)$$

This is a typical constrained optimization problem and can be solved using the Lagrange multiplier method. Let  $\psi_{ij}$  and  $\phi_{ij}$  be the Lagrange multiplier for constraint  $w_{ij} \geq 0$  and  $v_{ij} \geq 0$ , respectively, and let  $\Psi = [\psi_{ij}]$  and  $\Phi = [\phi_{ij}]$ . The Lagrangian function is

$$\begin{aligned} \mathcal{L} &= \text{tr}(\mathbf{K}) - 2 \text{tr}(\mathbf{V} \mathbf{W}^T \mathbf{K}) + \text{tr}(\mathbf{V} \mathbf{W}^T \mathbf{K} \mathbf{W} \mathbf{V}^T) \\ &\quad + \lambda \text{tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}) + \text{tr}(\Psi \mathbf{W}^T) + \text{tr}(\Phi \mathbf{V}^T). \end{aligned} \quad (15)$$

The partial derivatives of  $\mathcal{L}$  with respect to  $\mathbf{W}$  and  $\mathbf{V}$  are

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{W}} &= -2 \mathbf{K} \mathbf{V} + 2 \mathbf{K} \mathbf{W} \mathbf{V}^T \mathbf{V} + \Psi, \\ \frac{\partial \mathcal{L}}{\partial \mathbf{V}} &= -2 \mathbf{K} \mathbf{W} + 2 \mathbf{V} \mathbf{W}^T \mathbf{K} \mathbf{W} + 2 \lambda \mathbf{L} \mathbf{V} + \Phi. \end{aligned} \quad (16)$$

Using the Karush-Kuhn-Tucker conditions  $\psi_{ik}w_{ik} = 0$  and  $\phi_{jk}v_{jk} = 0$ , we get the following equations for  $w_{ik}$  and  $v_{jk}$ :

$$\begin{aligned} (-\mathbf{KV} + \mathbf{KWV}^T\mathbf{V})_{ik}w_{ik} &= 0, \\ (-\mathbf{KW} + \mathbf{VW}^T\mathbf{KW} + \lambda\mathbf{LV})_{jk}v_{jk} &= 0. \end{aligned} \quad (17)$$

The corresponding equivalent formulas are as follows:

$$\begin{aligned} (-\mathbf{KV} + \mathbf{KWV}^T\mathbf{V})_{ik}w_{ik}^2 &= 0, \\ (-\mathbf{KW} + \mathbf{VW}^T\mathbf{KW} + \lambda\mathbf{LV})_{jk}v_{jk}^2 &= 0. \end{aligned} \quad (18)$$

Introduce

$$\mathbf{A} = \mathbf{A}^+ - \mathbf{A}^-, \quad (19)$$

where  $\mathbf{A}_{ik}^+ = (|\mathbf{A}_{ik}| + \mathbf{A}_{ik})/2$  and  $\mathbf{A}_{ik}^- = (|\mathbf{A}_{ik}| - \mathbf{A}_{ik})/2$ .

The equations lead to the following updating formulas:

$$w_{ik} = w_{ik} \sqrt{\frac{(\mathbf{K}^+\mathbf{V})_{ik} + (\mathbf{K}^-\mathbf{WV}^T\mathbf{V})_{ik}}{(\mathbf{K}^-\mathbf{V})_{ik} + (\mathbf{K}^+\mathbf{WV}^T\mathbf{V})_{ik}}}, \quad (20)$$

$$v_{jk} = v_{jk} \sqrt{\frac{(\mathbf{K}^+\mathbf{W})_{jk} + (\mathbf{VW}^T\mathbf{K}^-\mathbf{W})_{jk} + \lambda(\mathbf{L}^-\mathbf{V})_{jk}}{(\mathbf{K}^-\mathbf{W})_{jk} + (\mathbf{VW}^T\mathbf{K}^+\mathbf{W})_{jk} + \lambda(\mathbf{L}^+\mathbf{V})_{jk}}}. \quad (21)$$

Note that the solution to minimizing the criterion function  $\mathcal{F}$  is not unique. If  $\mathbf{W}$  and  $\mathbf{V}$  are the solution, then,  $\mathbf{WD}$ ,  $\mathbf{WD}^{-1}$  will also form a solution for any positive diagonal matrix  $\mathbf{D}$ . To make the solution unique, we will further require that  $\mathbf{w}^T\mathbf{K}\mathbf{w} = 1$ , where  $\mathbf{w}$  is the column vector of  $\mathbf{W}$ . The matrix  $\mathbf{V}$  will be adjusted accordingly so that  $\mathbf{WV}^T$  does not change. This can be achieved by

$$\begin{aligned} \mathbf{V} &\leftarrow \mathbf{V}[\text{diag}(\mathbf{W}^T\mathbf{KW})]^{1/2}, \\ \mathbf{W} &\leftarrow \mathbf{W}[\text{diag}(\mathbf{W}^T\mathbf{KW})]^{1/2}. \end{aligned} \quad (22)$$

**3.3. Convergence Analysis.** In this section, we will investigate the convergence of the updating formula in (14). We use the auxiliary function approach [16] to prove the convergence. Here we first introduce the definition of auxiliary function [16].

*Definition 1.*  $Z(h, h')$  is an auxiliary function of  $F(h)$  if the conditions

$$Z(h, h') \geq F(h), \quad Z(h, h) = F(h) \quad (23)$$

are satisfied.

**Lemma 2.** If  $Z$  is an auxiliary function for  $F$ , then  $F$  is nonincreasing under the update

$$h^{(t+1)} = \arg \min_h Z(h, h^{(t)}). \quad (24)$$

*Proof.* Consider

$$F(h^{(t+1)}) \leq Z(h^{(t+1)}, h^{(t)}) \leq Z(h^{(t)}, h^{(t)}) = F(h^{(t)}). \quad (25)$$

□

**Lemma 3.** For any nonnegative matrices  $\mathbf{A} \in \mathbb{R}^{n \times n}$ ,  $\mathbf{B} \in \mathbb{R}^{k \times k}$ ,  $\mathbf{S} \in \mathbb{R}^{n \times k}$ , and  $\mathbf{S}' \in \mathbb{R}^{n \times k}$ , and  $\mathbf{A}$ ,  $\mathbf{B}$  are symmetric, then the following inequality holds:

$$\sum_{i=1}^n \sum_{p=1}^k \frac{(\mathbf{AS}'\mathbf{B})_{ip}^2}{\mathbf{S}'_{ip}} \geq \text{tr}(\mathbf{S}^T\mathbf{ASB}). \quad (26)$$

The correctness and convergence of the algorithm are addressed in the following.

For given  $\mathbf{K}$ , fixing  $\mathbf{V}$ , considering any element  $w_{ij}$  in  $\mathbf{W}$ , we use  $\mathcal{F}(\mathbf{W})$  to denote the part of  $\mathcal{F}$ , which is only relevant to  $w_{ij}$ . We get

$$\mathcal{F}(\mathbf{W}) = -2 \text{tr}(\mathbf{VW}^T\mathbf{K}) + \text{tr}(\mathbf{VW}^T\mathbf{KWV}^T). \quad (27)$$

**Theorem 4.** One rewrites  $\mathcal{F}(\mathbf{W})$  as follows:

$$\begin{aligned} \mathcal{F}(\mathbf{H}) &= \text{tr}(-2\mathbf{H}^T\mathbf{B}^+ + 2\mathbf{H}^T\mathbf{B}^- + \mathbf{H}^T\mathbf{A}^+\mathbf{HC} \\ &\quad - \mathbf{H}^T\mathbf{A}^-\mathbf{HC}), \end{aligned} \quad (28)$$

where  $\mathbf{B} = \mathbf{KV}$ ,  $\mathbf{A} = \mathbf{K}$ ,  $\mathbf{C} = \mathbf{V}^T\mathbf{V}$ , and  $\mathbf{H} = \mathbf{W}$ .

Then the following function

$$\begin{aligned} Z(\mathbf{H}, \mathbf{H}') &= - \sum_{ik} 2\mathbf{B}_{ik}^+\mathbf{H}'_{ik} \left(1 + \log \frac{\mathbf{H}_{ik}}{\mathbf{H}'_{ik}}\right) \\ &\quad + \sum_{ik} \mathbf{B}_{ik}^- \frac{\mathbf{H}_{ik}^2 + (\mathbf{H}')_{ik}^2}{\mathbf{H}'_{ik}} + \sum_{ik} \frac{(\mathbf{A}^+\mathbf{H}'\mathbf{C})_{ik}\mathbf{H}_{ik}^2}{\mathbf{H}'_{ik}} \\ &\quad - \sum_{ijkl} \mathbf{A}_{ij}^-\mathbf{H}'_{jk}\mathbf{C}_{kl}\mathbf{H}'_{il} \left(1 + \log \frac{\mathbf{H}_{jk}\mathbf{H}_{il}}{\mathbf{H}'_{jk}\mathbf{H}'_{il}}\right) \end{aligned} \quad (29)$$

is an auxiliary function of  $\mathcal{F}(\mathbf{H})$ : that is, it satisfies the requirements  $\mathcal{F}(\mathbf{H}) \leq Z(\mathbf{H}, \mathbf{H}')$  and  $\mathcal{F}(\mathbf{H}) = Z(\mathbf{H}, \mathbf{H})$ . Furthermore, it is a convex function of  $\mathbf{H}$  and its global minimum is

$$\mathbf{H}_{ik} = \underset{\mathbf{H}}{\text{argmin}} \mathbf{H}'_{ik} \sqrt{\frac{\mathbf{B}_{ik}^+ + (\mathbf{H}'\mathbf{A}^-\mathbf{C})_{ik}}{\mathbf{B}_{ik}^- + (\mathbf{A}^+\mathbf{H}'\mathbf{C})_{ik}}}. \quad (30)$$

From its minima and setting  $\mathbf{H}^{(t+1)} \leftarrow \mathbf{H}$  and  $\mathbf{H}^{(t)} \leftarrow \mathbf{H}'$ , one recovers (20), letting  $\mathbf{B}^+ = (\mathbf{K}^+)\mathbf{V}$ ,  $\mathbf{B}^- = (\mathbf{K}^-\mathbf{V})$ ,  $\mathbf{A} = \mathbf{K}$ ,  $\mathbf{C} = \mathbf{V}^T\mathbf{V}$ , and  $\mathbf{H} = \mathbf{W}$ .

*Proof.* The function  $\mathcal{F}(\mathbf{H})$  is

$$\begin{aligned} \mathcal{F}(\mathbf{H}) &= \text{tr}(-2\mathbf{H}^T\mathbf{B}^+ + 2\mathbf{H}^T\mathbf{B}^- + \mathbf{H}^T\mathbf{A}^+\mathbf{HC} \\ &\quad - \mathbf{H}^T\mathbf{A}^-\mathbf{HC}). \end{aligned} \quad (31)$$

We find upper bounds for each of the two positive terms and lower bounds for each of the two negative terms. For the third term in  $\mathcal{F}(\mathbf{H})$ , by applying Lemma 3, we obtain an upper bound

$$\text{tr}(\mathbf{H}^T \mathbf{A}^+ \mathbf{H} \mathbf{C}) \leq \sum_{ik} \frac{(\mathbf{A}^+ \mathbf{H}' \mathbf{C})_{ik} \mathbf{H}_{ik}^2}{\mathbf{H}'_{ik}}. \quad (32)$$

The second term of  $\mathcal{F}(\mathbf{H})$  is bounded by

$$\text{tr}(\mathbf{H}^T \mathbf{B}^-) = \sum_{ik} \mathbf{H}_{ik} \mathbf{B}_{ik}^- \leq \sum_{ik} \mathbf{B}_{ik}^- \frac{\mathbf{H}_{ik}^2 + (\mathbf{H}')_{ik}^2}{2\mathbf{H}'_{ik}}. \quad (33)$$

To obtain lower bounds for the two remaining terms, we use the inequality  $z \geq 1 + \log z$ , which holds for any  $z > 0$ , and the first term in  $\mathcal{F}(\mathbf{H})$  is bounded by

$$\text{tr}(\mathbf{H}^T \mathbf{B}^+) = \sum_{ik} \mathbf{B}_{ik}^+ \mathbf{H}_{ik} \geq \sum_{ik} \mathbf{B}_{ik}^+ \mathbf{H}'_{ik} \left(1 + \log \frac{\mathbf{H}_{ik}}{\mathbf{H}'_{ik}}\right). \quad (34)$$

The last term in  $\mathcal{F}(\mathbf{H})$  is bounded by

$$\text{tr}(\mathbf{H}^T \mathbf{A}^- \mathbf{H} \mathbf{C}) \geq \sum_{ijkl} \mathbf{A}_{ij}^- \mathbf{H}'_{jk} \mathbf{C}_{kl} \mathbf{H}'_{il} \left(1 + \log \frac{\mathbf{H}_{jk} \mathbf{H}_{il}}{\mathbf{H}'_{jk} \mathbf{H}'_{il}}\right). \quad (35)$$

Collecting all bounds, we obtain  $Z(\mathbf{H}, \mathbf{H}')$  as in (29). Obviously,  $\mathcal{F}(\mathbf{H}) \leq Z(\mathbf{H}, \mathbf{H}')$  and  $\mathcal{F}(\mathbf{H}) = Z(\mathbf{H}, \mathbf{H})$ .

To find the minimum of  $Z(\mathbf{H}, \mathbf{H}')$ , we take

$$\begin{aligned} \frac{\partial Z(\mathbf{H}, \mathbf{H}')}{\partial \mathbf{H}_{ik}} &= -2\mathbf{B}_{ik}^+ \frac{\mathbf{H}'_{ik}}{\mathbf{H}_{ik}} + 2\mathbf{B}_{ik}^- \frac{\mathbf{H}_{ik}}{\mathbf{H}'_{ik}} \\ &+ \frac{2(\mathbf{H}' \mathbf{A}^+ \mathbf{C})_{ik} \mathbf{H}_{ik}}{\mathbf{H}'_{ik}} - \frac{2(\mathbf{H}' \mathbf{A}^- \mathbf{C})_{ik} \mathbf{H}'_{ik}}{\mathbf{H}_{ik}}. \end{aligned} \quad (36)$$

To find the minimum of  $Z(\mathbf{H}, \mathbf{H}')$ , we take the Hessian matrix of  $Z(\mathbf{H}, \mathbf{H}')$

$$\frac{\partial^2 Z(\mathbf{H}, \mathbf{H}')}{\partial \mathbf{H}_{ik} \partial \mathbf{H}_{jl}} = \delta_{ij} \delta_{kl} \mathbf{Y}_{ik} \quad (37)$$

To be a diagonal matrix with positive entries

$$\mathbf{Y}_{ik} = \frac{4[(\mathbf{B}^+)_{ik} + (\mathbf{H}' \mathbf{A}^- \mathbf{C})_{ik}] \mathbf{H}'_{ik}}{\mathbf{H}_{ik}^2} + 2 \frac{\mathbf{B}_{ik}^- + (\mathbf{H}' \mathbf{A}^+ \mathbf{C})_{ik}}{\mathbf{H}'_{ik}}. \quad (38)$$

Thus,  $Z(\mathbf{H}, \mathbf{H}')$  is a convex function of  $\mathbf{H}$ . Therefore, we obtain the global minimum by setting  $\partial Z(\mathbf{H}, \mathbf{H}') / \partial \mathbf{H}_{ik} = 0$  in (36) and solving for  $\mathbf{H}$ . Rearranging, we obtain (30).  $\square$

**Theorem 5.** Updating  $\mathbf{W}$  using (20) will monotonically decrease the value of the objective in (13); hence it converges.

*Proof.* By Lemma 2 and Theorem 4, we can get that  $\mathcal{F}(\mathbf{W}^t) = Z(\mathbf{W}^{(t)}, \mathbf{W}^{(t)}) \geq Z(\mathbf{W}^{(t+1)}, \mathbf{W}^{(t)}) \geq \mathcal{F}(\mathbf{W}^{(t+1)})$ , so  $\mathcal{F}(\mathbf{W})$  is monotonically decreasing. Since  $\mathcal{F}(\mathbf{W})$  is obviously bounded below, we prove this theorem.

For given  $\mathbf{K}$ , fixing  $\mathbf{W}$ , considering any element  $v_{ij}$  in  $\mathbf{V}$ , we use  $\mathcal{F}(\mathbf{V})$  to denote the part of  $\mathcal{F}$ , which is only relevant to  $v_{ij}$ . We get

$$\mathcal{F}(\mathbf{V}) = -2 \text{tr}(\mathbf{V} \mathbf{W}^T \mathbf{K}) + \text{tr}(\mathbf{V} \mathbf{W}^T \mathbf{K} \mathbf{W} \mathbf{V}^T) + \lambda \text{tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}). \quad (39)$$

$\square$

**Theorem 6.** One rewrites  $\mathcal{F}(\mathbf{V})$  as follows:

$$\begin{aligned} \mathcal{F}(\mathbf{H}) &= \text{tr}(-2\mathbf{H}^T \mathbf{B}^+ + 2\mathbf{H}^T \mathbf{B}^- + \mathbf{H} \mathbf{A}^+ \mathbf{H}^T \\ &- \mathbf{H} \mathbf{A}^- \mathbf{H}^T + \lambda \mathbf{H}^T \mathbf{L}^+ \mathbf{H} - \lambda \mathbf{H}^T \mathbf{L}^- \mathbf{H}), \end{aligned} \quad (40)$$

where  $\mathbf{B} = \mathbf{K} \mathbf{W}$ ,  $\mathbf{A} = \mathbf{W}^T \mathbf{X}^T \mathbf{X} \mathbf{W}$ , and  $\mathbf{H} = \mathbf{V}$ .

Then the following function

$$\begin{aligned} Z(\mathbf{H}, \mathbf{H}') &= - \sum_{ik} 2\mathbf{B}_{ik}^+ \mathbf{H}'_{ik} \left(1 + \log \frac{\mathbf{H}_{ik}}{\mathbf{H}'_{ik}}\right) \\ &+ \sum_{ik} \mathbf{B}_{ik}^- \frac{\mathbf{H}_{ik}^2 + (\mathbf{H}')_{ik}^2}{\mathbf{H}'_{ik}} + \sum_{ik} \frac{(\mathbf{A}^+ \mathbf{H}')_{ik} \mathbf{H}_{ik}^2}{\mathbf{H}'_{ik}} \\ &- \sum_{ikl} \mathbf{A}_{kl}^- \mathbf{H}'_{ik} \mathbf{H}'_{li} \left(1 + \log \frac{\mathbf{H}_{ik} \mathbf{H}_{li}}{\mathbf{H}'_{ik} \mathbf{H}'_{li}}\right) \\ &+ \lambda \sum_{ik} \frac{(\mathbf{L}^+ \mathbf{H}')_{ik} \mathbf{H}_{ik}^2}{\mathbf{H}'_{ik}} \\ &- \lambda \sum_{ikl} \mathbf{L}_{kl}^- \mathbf{H}'_{ik} \mathbf{H}'_{li} \left(1 + \log \frac{\mathbf{H}_{ik} \mathbf{H}_{li}}{\mathbf{H}'_{ik} \mathbf{H}'_{li}}\right) \end{aligned} \quad (41)$$

is an auxiliary function of  $\mathcal{F}(\mathbf{H})$ : that is, it satisfies the requirements  $\mathcal{F}(\mathbf{H}) \leq Z(\mathbf{H}, \mathbf{H})$  and  $\mathcal{F}(\mathbf{H}) = Z(\mathbf{H}, \mathbf{H}')$ . Furthermore, it is a convex function of  $\mathbf{H}$  and its global minimum is

$$\mathbf{H}_{ik} = \mathbf{H}'_{ik} \sqrt{\frac{\mathbf{B}_{ik}^+ + (\mathbf{H}' \mathbf{A}^-)_{ik} + \lambda(\mathbf{L}^- \mathbf{H}')_{ik}}{\mathbf{B}_{ik}^- + (\mathbf{H}' \mathbf{A}^+)_{ik} + \lambda(\mathbf{L}^+ \mathbf{H}')_{ik}}}. \quad (42)$$

From its minima and setting  $\mathbf{H}^{(t+1)} \leftarrow \mathbf{H}$  and  $\mathbf{H}^{(t)} \leftarrow \mathbf{H}'$ , one recovers (21), letting  $\mathbf{B}^+ = (\mathbf{K})^+ \mathbf{U}$ ,  $\mathbf{B}^- = (\mathbf{K})^- \mathbf{U}$ ,  $\mathbf{A} = \mathbf{W}^T \mathbf{K} \mathbf{W}$ , and  $\mathbf{H} = \mathbf{V}$ .

*Proof.* The function  $\mathcal{F}(\mathbf{H})$  is

$$\begin{aligned} \mathcal{F}(\mathbf{H}) &= \text{tr}(-2\mathbf{H}^T \mathbf{B}^+ + 2\mathbf{H}^T \mathbf{B}^- + \mathbf{H} \mathbf{A}^+ \mathbf{H}^T \\ &- \mathbf{H} \mathbf{A}^- \mathbf{H}^T + \lambda \mathbf{H}^T \mathbf{L}^+ \mathbf{H} - \lambda \mathbf{H}^T \mathbf{L}^- \mathbf{H}). \end{aligned} \quad (43)$$

We find upper bounds for each of the three positive terms and lower bounds for each of the three negative terms. For the

third term in  $\mathcal{F}(\mathbf{H})$ , by applying Lemma 3 and setting  $\mathbf{A} \leftarrow \mathbf{I}$ ,  $\mathbf{B} \leftarrow \mathbf{A}^+$ , we obtain an upper bound

$$\text{tr}(\mathbf{H}\mathbf{A}^+\mathbf{H}^T) \leq \sum_{ik} \frac{(\mathbf{H}'\mathbf{A}^+)_{ik} \mathbf{H}'_{ik}}{\mathbf{H}'_{ik}}. \quad (44)$$

The second term of  $\mathcal{F}(\mathbf{H})$  is bounded by

$$\text{tr}(\mathbf{H}^T\mathbf{B}^-) = \sum_{ik} \mathbf{H}_{ik}\mathbf{B}_{ik}^- \leq \sum_{ik} \mathbf{B}_{ik}^- \frac{\mathbf{H}_{ik}^2 + (\mathbf{H}')_{ik}^2}{2\mathbf{H}'_{ik}}, \quad (45)$$

using the inequality  $a \leq (a^2 + b^2)/2b$ , which holds for any  $a, b > 0$ .

For the fifth term in  $\mathcal{F}(\mathbf{H})$ , setting  $\mathbf{A} \leftarrow \mathbf{L}^+$ ,  $\mathbf{B} \leftarrow \mathbf{I}$ , and  $\mathbf{S} \leftarrow \mathbf{H}$ , we obtain an upper bound

$$\lambda \text{tr}(\mathbf{H}^T\mathbf{L}^+\mathbf{H}) \leq \lambda \sum_{ik} \frac{(\mathbf{L}^+\mathbf{H}')_{ik} \mathbf{H}'_{ik}}{\mathbf{H}'_{ik}}. \quad (46)$$

To obtain lower bounds for the three remaining terms, we use the inequality  $z \geq 1 + \log z$ , which holds for any  $z > 0$ , and the first term in  $J(\mathbf{H})$  is bounded by

$$\text{tr}(\mathbf{H}^T\mathbf{B}^+) = \sum_{ik} \mathbf{B}_{ik}^+ \mathbf{H}_{ik} \geq \sum_{ik} \mathbf{B}_{ik}^+ \mathbf{H}'_{ik} \left(1 + \log \frac{\mathbf{H}_{ik}}{\mathbf{H}'_{ik}}\right). \quad (47)$$

The fourth term in  $\mathcal{F}(\mathbf{H})$  is bounded by

$$\text{tr}(\mathbf{H}\mathbf{A}^-\mathbf{H}^T) \geq \sum_{ikl} \mathbf{A}_{kl}^- \mathbf{H}'_{ik} \mathbf{H}'_{li} \left(1 + \log \frac{\mathbf{H}_{ik}\mathbf{H}_{li}}{\mathbf{H}'_{ik}\mathbf{H}'_{li}}\right). \quad (48)$$

The last term in  $\mathcal{F}(\mathbf{H})$  is bounded by

$$\lambda \text{tr}(\mathbf{H}^T\mathbf{L}^-\mathbf{H}) \geq \lambda \sum_{ikl} \mathbf{L}_{kl}^- \mathbf{H}'_{ik} \mathbf{H}'_{li} \left(1 + \log \frac{\mathbf{H}_{ik}\mathbf{H}_{li}}{\mathbf{H}'_{ik}\mathbf{H}'_{li}}\right). \quad (49)$$

Collecting all bounds, we obtain  $Z(\mathbf{H}, \mathbf{H}')$  as in (41). Obviously,  $\mathcal{F}(\mathbf{H}) \leq Z(\mathbf{H}, \mathbf{H}')$  and  $\mathcal{F}(\mathbf{H}) = Z(\mathbf{H}, \mathbf{H})$ .

To find the minimum of  $Z(\mathbf{H}, \mathbf{H}')$ , we take

$$\begin{aligned} \frac{\partial Z(\mathbf{H}, \mathbf{H}')}{\partial \mathbf{H}_{ik}} &= -2\mathbf{B}_{ik}^+ \frac{\mathbf{H}'_{ik}}{\mathbf{H}_{ik}} + 2\mathbf{B}_{ik}^- \frac{\mathbf{H}_{ik}}{\mathbf{H}'_{ik}} \\ &+ \frac{2(\mathbf{H}'\mathbf{A}^+)_{ik}\mathbf{H}_{ik}}{\mathbf{H}'_{ik}} - \frac{2(\mathbf{H}'\mathbf{A}^-)_{ik}\mathbf{H}'_{ik}}{\mathbf{H}_{ik}} \\ &+ \frac{2\lambda(\mathbf{L}^+\mathbf{H}')_{ik}\mathbf{H}_{ik}}{\mathbf{H}'_{ik}} - \frac{2\lambda(\mathbf{L}^-\mathbf{H}')_{ik}\mathbf{H}'_{ik}}{\mathbf{H}_{ik}} = 0. \end{aligned} \quad (50)$$

We have

$$\begin{aligned} &-2\mathbf{B}_{ik}^+(\mathbf{H}')_{ik}^2 + 2\mathbf{B}_{ik}^-\mathbf{H}_{ik}^2 + 2(\mathbf{H}'\mathbf{A}^+)_{ik}\mathbf{H}_{ik}^2 \\ &-2(\mathbf{H}'\mathbf{A}^-)_{ik}(\mathbf{H}')_{ik}^2 + 2\lambda(\mathbf{L}^+\mathbf{H}')_{ik}\mathbf{H}_{ik}^2 \\ &-2\lambda(\mathbf{L}^-\mathbf{H}')_{ik}(\mathbf{H}')_{ik}^2 = 0. \end{aligned} \quad (51)$$

Therefore

$$\mathbf{H}_{ik} = \mathbf{H}'_{ik} \sqrt{\frac{\mathbf{B}_{ik}^+ + (\mathbf{H}'\mathbf{A}^-)_{ik} + \lambda(\mathbf{L}^-\mathbf{H}')_{ik}}{\mathbf{B}_{ik}^- + (\mathbf{H}'\mathbf{A}^+)_{ik} + \lambda(\mathbf{L}^+\mathbf{H}')_{ik}}}. \quad (52)$$

The Hessian matrix containing the second derivatives

$$\frac{\partial^2 Z(\mathbf{H}, \mathbf{H}')}{\partial \mathbf{H}_{ik} \partial \mathbf{H}_{jl}} = \delta_{ij} \delta_{kl} \mathbf{Y}_{ik} \quad (53)$$

is a diagonal matrix with positive entries

$$\begin{aligned} \mathbf{Y}_{ik} &= \frac{4[(\mathbf{B}^+)_{ik} + (\mathbf{H}'\mathbf{A}^-)_{ik} + \lambda(\mathbf{L}^-\mathbf{H}')_{ik}]\mathbf{H}'_{ik}}{\mathbf{H}_{ik}^2} \\ &+ 2\frac{\mathbf{B}_{ik}^- + (\mathbf{H}'\mathbf{A}^+)_{ik} + \lambda(\mathbf{L}^+\mathbf{H}')_{ik}}{\mathbf{H}'_{ik}}. \end{aligned} \quad (54)$$

Thus,  $Z(\mathbf{H}, \mathbf{H}')$  is a convex function of  $\mathbf{H}$ . Therefore, we obtain the global minimum by setting  $\partial Z(\mathbf{H}, \mathbf{H}')/\partial \mathbf{H}_{ik} = 0$  in (41) and solving for  $\mathbf{H}$ . Rearranging, we obtain (21).  $\square$

**Theorem 7.** *Updating  $\mathbf{V}$  using (21) will monotonically decrease the value of the objective in (13); hence it converges.*

*Proof.* By Lemma 2 and Theorem 4, we can get that  $\mathcal{F}(\mathbf{V}^t) = Z(\mathbf{V}^{(t)}, \mathbf{V}^{(t)}) \geq Z(\mathbf{V}^{(t+1)}, \mathbf{V}^{(t)}) \geq \mathcal{F}(\mathbf{V}^{t+1})$ , so  $\mathcal{F}(\mathbf{V})$  is monotonically decreasing. Since  $\mathcal{F}(\mathbf{V})$  is obviously bounded below, we prove this theorem.  $\square$

## 4. Experimental Results

In this section, we show the performance of the proposed method on face recognition and compare our proposed method with the popular subspace learning algorithms: four unsupervised ones which are principal component analysis [21] (PCA), neighborhood preserving embedding (NPE) [20], local nonnegative matrix factorization (LNMF) [4], and convex nonnegative factorization (CNMF) [13] the one supervised algorithm and which is linear discriminant analysis (LDA) [21]. We use the nearest neighbor (NN) classifier as baseline in original space. We apply different algorithms to obtain new representations for each chosen data set, and then the NN method is applied in the new representation spaces.

**4.1. Data Preparation.** The experiments are used on three data sets. One is Cambridge ORL face database, the other is the Yale database, and the third one is the CMU PIE face database. The important statistics of these data sets are described below.

The Yale database contains 165 gray scale images of 15 individuals. All images demonstrate variations in lighting condition (left-light, center-light, right-light), facial expression (normal, happy, sad, sleepy, surprised, and wink), and with/without glasses.

The ORL database contains ten different images of each of 40 distinct subjects, thus 400 images in total. For some subjects, the images were taken at different times, varying the lighting, facial expressions (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement).

TABLE 1: Face recognition accuracy on the ORL data set. The number in brackets is the corresponding projection dimensionality.

Method	2 Train	3 Train	4 Train
Baseline	69.32%	77.56%	83.48%
PCA	69.32% (79)	77.56% (118)	83.48% (152)
LDA	72.80% (25)	83.79% (39)	90.13% (39)
NPE	73.19% (36)	84.29% (54)	91.06% (73)
NMF	70.87% (97)	78.98% (81)	84.48% (95)
LNMF	71.73% (178)	81.09% (168)	86.31% (195)
CNMF	72.23% (138)	83.58% (143)	89.56% (111)
NPCNMF	77.31% (143)	86.73% (153)	93.35% (145)

The CMU PIE face database contains more than 40 000 facial images of 68 people. The images were acquired over different poses, under variable illumination conditions, and with different facial expressions. In our experiment, we choose the images from the frontal pose (C27) and each subject has around 49 images from varying illuminations and facial expressions.

In all the experiments, images are preprocessed so that faces are located. Original images are first normalized in scale and orientation such that the two eyes are aligned at the same position. Then the facial areas were cropped into the final images for clustering. Each image is of  $32 \times 32$  pixels with 256 gray levels per pixel.

**4.2. Parameter Settings.** For each data set, we randomly divide it into training and testing sets, and evaluate the recognition accuracy on the testing set. In detail, for each individual in the ORL and Yale data sets, we randomly select 2, 3, and 4 images per individual, respectively, for training samples, and the remaining for test samples, while for each individual in the PIE data set, we randomly select 5, 10, and 20 images per individual for training samples. For each partition, we repeated each experiment 20 times and calculated the average recognition accuracy. In general, the recognition rate varies with the dimension of the face subspace. The best result obtained in the optimal subspace and the corresponding dimensionality for each method are shown.

For the face recognition experiments, several parameters need to be decided beforehand. For LDA, we use PCA as a first step dimensionality reduction algorithm to avoid the singularity problem. The dimension of the PCA step is fixed as  $N - c$  and then performs LDA. There are two parameters in our NPCNMF and NPE approach: the number of nearest neighbors  $k$  and the regularization parameter  $\lambda$ . Throughout our experiments, we empirically set the number of nearest neighbors  $k$  to 5, the value of the regularization parameter  $\lambda$  to 100.

Each testing sample  $y$  is projected into the linear subspace spanned by the column vectors of the basis matrix  $U$ , namely,  $\mathbf{h}_y \approx \mathbf{W}^\dagger y$ , where  $\mathbf{W}^\dagger$  indicates the pseudoinverse of matrix  $\mathbf{W}$ .

**4.3. Classification Results.** Tables 1, 2, and 3 show the evaluation results of all the methods on the three data sets,

TABLE 2: Face recognition accuracy on the Yale data set. The number in brackets is the corresponding projection dimensionality.

Method	2 Train	3 Train	4 Train
Baseline	46.04%	49.96%	55.62%
PCA	46.04% (29)	49.96% (44)	55.62% (58)
LDA	42.81% (11)	60.33% (14)	68.10% (13)
NPE	48.19% (13)	62.00% (19)	69.00% (73)
NMF	44.11% (112)	49.00% (195)	52.19% (164)
LNMF	44.00% (157)	48.84% (198)	53.57% (197)
CNMF	49.72% (125)	59.50% (168)	65.77% (129)
NPCNMF	63.45% (124)	71.83% (148)	81.38% (153)

TABLE 3: Face recognition accuracy on the PIE data set. The number in brackets is the corresponding projection dimensionality.

Method	5 Train	10 Train	20 Train
Baseline	43.02%	62.90%	83.19%
PCA	42.87% (199)	62.51% (195)	82.84% (200)
LDA	84.39% (67)	90.47% (67)	93.98% (67)
NPE	84.71% (166)	91.48% (200)	94.33% (200)
NMF	78.66% (200)	88.98% (200)	92.52% (200)
LNMF	76.47% (200)	87.91% (200)	92.61% (196)
CNMF	83.72% (176)	90.89% (187)	93.78% (159)
NPCNMF	88.43% (147)	94.86% (158)	98.58% (133)

respectively, where the value in each entry represents the average recognition accuracy of 20 independent trials, and the number in brackets is the corresponding projection dimensionality. These experiments reveal a number of interesting points.

- (1) It is clear that the use of dimensionality reduction is beneficial in face recognition. There is a significant increase in performance from using LDA, NPE, NMF, LNMF, and CNMF. However, PCA fails to gain improvement over the baseline. This is because that PCA does not encode the discriminative information.
- (2) The performances of nonnegative algorithms NMF, LNMF, and CNMF are much worse than supervised algorithms LDA, which shows that without considering the labeled data, nonnegative algorithms could not guarantee good discriminating power.
- (3) Our NPCNMF algorithm outperforms all other five methods. The reason lies in the fact that NPCNMF considers the geometrical structure of the data and achieves better performance than the other algorithms. This shows that by leveraging the power of both the parts-based representation and the intrinsic geometrical structure of the data, NPCNMF can learn a better compact representation in the sense of semantic structure.

## 5. Conclusion and Future Work

In this paper, we have presented a novel matrix factorization method called NPCNMF for dimensionality reduction,

which respects the local geometric structure. As a result, NPCNMF can discriminate power more than the ordinary NMF and CNMF approaches which only consider the Euclidean structure of the data. Experimental results on face datasets show that NPCNMF provides better representation in the sense of semantic structure.

Several challenges remain to be investigated in our future work.

- (1) A suitable value of  $\lambda$  is important to our algorithm. It remains unknown how to do model selection theoretically and efficiently.
- (2) NPCNMF is currently limited to the linear projections, and those nonlinear techniques (e.g., kernel tricks) may further boost the algorithmic performance. We will investigate it in our future work.
- (3) Another further research direction is how to extend the current framework for tensor-based nonnegative data decomposition.
- (4) NPCNMF algorithm is iterative and sensitive to the initialization of  $\mathbf{W}$  and  $\mathbf{H}$ . It is unclear how to choose optimal initialization parameters in a principled manner.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## References

- [1] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.
- [2] D. D. Lee and H. S. Seung, "Algorithms for nonnegative matrix factorization," in *Proceedings of the Conference on Neural Information Processing Systems (NIPS '00)*, 2000.
- [3] M. Cooper and J. Foote, "Summarizing video using nonnegative similarity matrix factorization," in *Proceedings of the IEEE Workshop on Multimedia Signal Processing*, 2002.
- [4] S. Z. Li, X. Hou, H. Zhang, and Q. Cheng, "Learning spatially localized, parts-based representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CPRV '01)*, pp. I207–I212, December 2001.
- [5] P. Paatero and U. Tapper, "Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111–126, 1994.
- [6] P. O. Hoyer, "Non-negative sparse coding," in *Proceedings of IEEE Workshop on Neural Networks for Signal Processing*, 2002.
- [7] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2003.
- [8] T. Hazan, S. Polak, and A. Shashua, "Sparse image coding using a 3D non-negative tensor factorization," in *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV '05)*, pp. 50–57, October 2005.
- [9] A. Shashua and T. Hazan, "Non-negative tensor factorization with applications to statistics and computer vision," in *Proceedings of the 22nd International Conference on Machine Learning (ICML '05)*, pp. 793–800, August 2005.
- [10] Y. Wang, Y. Jia, C. Hu, and M. Turk, "Fisher non-negative matrix factorization for learning local features," in *Proceedings of the 6th Asian Conference on Computer Vision (ACCV '04)*, 2004.
- [11] S. Zafeiriou, A. Tefas, I. Buciu, and I. Pitas, "Exploiting discriminant information in nonnegative matrix factorization with application to frontal face verification," *IEEE Transactions on Neural Networks*, vol. 17, no. 3, pp. 683–695, 2006.
- [12] I. Kotsia, S. Zafeiriou, and I. Pitas, "Novel discriminant non-negative matrix factorization algorithm with applications to facial image characterization problems," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 588–595, 2007.
- [13] C. H. Ding, T. Li, and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 45–55, 2010.
- [14] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.
- [15] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
- [16] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1548–1560, 2011.
- [17] D. Cai, X. He, and J. Han, "Locally consistent concept factorization for document clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 6, pp. 902–913, 2011.
- [18] L. Zhang, L. Qiao, and S. Chen, "Graph-optimized locality preserving projections," *Pattern Recognition*, vol. 43, no. 6, pp. 1993–2002, 2010.
- [19] X. He and P. Niyogi, "Locality preserving projections," in *Proceedings of the NIPS*, 2003.
- [20] X. He, D. Cai, S. Yan, and H.-J. Zhang, "Neighborhood preserving embedding," in *Proceedings 10th IEEE International Conference on Computer Vision (ICCV '05)*, pp. 1208–1213, October 2005.
- [21] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

