

## Research Article

# Incremental Nonnegative Matrix Factorization for Face Recognition

Wen-Sheng Chen,<sup>1</sup> Binbin Pan,<sup>1</sup> Bin Fang,<sup>2</sup> Ming Li,<sup>3</sup> and Jianliang Tang<sup>1</sup>

<sup>1</sup> College of Mathematics and Computational Science, Shenzhen University, Shenzhen 518060, China

<sup>2</sup> College of Computer Science, Chongqing University, Chongqing 400044, China

<sup>3</sup> School of Information Science & Technology, East China Normal University, Shanghai 200241, China

Correspondence should be addressed to Wen-Sheng Chen, chenws@szu.edu.cn

Received 25 May 2008; Accepted 5 June 2008

Recommended by Cristian Toma

Nonnegative matrix factorization (NMF) is a promising approach for local feature extraction in face recognition tasks. However, there are two major drawbacks in almost all existing NMF-based methods. One shortcoming is that the computational cost is expensive for large matrix decomposition. The other is that it must conduct repetitive learning, when the training samples or classes are updated. To overcome these two limitations, this paper proposes a novel incremental nonnegative matrix factorization (INMF) for face representation and recognition. The proposed INMF approach is based on a novel constraint criterion and our previous block strategy. It thus has some good properties, such as low computational complexity, sparse coefficient matrix. Also, the coefficient column vectors between different classes are orthogonal. In particular, it can be applied to incremental learning. Two face databases, namely FERET and CMU PIE face databases, are selected for evaluation. Compared with PCA and some state-of-the-art NMF-based methods, our INMF approach gives the best performance.

Copyright © 2008 Wen-Sheng Chen 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.

## 1. Introduction

Face recognition has been one of the most challenging problems in computer science and information technology since 1990 [1, 2]. The approaches of face recognition can be mainly categorized into two groups, namely geometric feature-based and appearance-based [3]. The geometric features are based on the short range phenomena of face images such as eyes, eyebrows, nose, and mouth. The facial local features are learnt to form a face geometric feature vector for face recognition. The appearance-based approach relies on the global facial features, which generate an entire facial feature vector for face classification. Nonnegative matrix factorization (NMF) [4, 5] belongs to geometric feature-based category, while principle component analysis (PCA) [6] is based on the whole facial features. Both NMF and PCA are unsupervised learning methods for face recognition. The basic ideas of

these two approaches are to find the basis images using different criterions. All face images can be reconstructed by the basis images. The basis images of PCA are called eigenfaces, which are the eigenvectors corresponding to large eigenvalues of total scatter matrix. NMF aims to perform nonnegative matrix decomposition on the training image matrix  $V$  such that  $V \approx WH$ , where  $W$  and  $H$  are the basis image matrix and the coefficient matrix, respectively. The local image features are learnt and contained in  $W$  as column vectors. Follow the success of applying NMF in learning the parts of objects [4], many researchers have conducted in-depth investigation on NMF and different NMF-based approaches have been developed [7–19]. Li et al. proposed a local NMF method [7] by adding some spatial constraints. Wild et al. [8] utilized spherical  $K$ -means clustering to produce a structured initialization for NMF. Buciu and Pitas [9] presented a DNMF method for learning facial expressions in a supervised manner. However, DNMF does not guarantee convergence to a stationary limit point. Kotsia et al. [15] thus presented a modified DNMF method using projected gradients. Some similar supervised methods incorporated into NMF were developed to enhance the classification power of NMF [11–13, 19]. Hoyer [10] added sparseness constraints to NMF to find solutions with desired degrees of sparseness. Lin [16, 17] modified traditional NMF updates using projected gradient method and discussed their convergences. Recently, Zhang et al. [18] proposed a topology structure preservation constraint in NMF to improve the NMF performance.

However, to the best of our knowledge, almost all existing NMF-based approaches encounter two major problems, namely time-consuming problem and incremental learning problem. In most cases, the training image matrix  $V$  is very large and it leads to expensive computational cost for NMF-based schemes. Also, when the training samples or classes are updated, NMF must implement repetitive learning. These drawbacks greatly restrict the practical applications of NMF-based methods to face recognition. To avoid the above two problems, this paper, motivated by our previous work on incremental learning [19], proposes a supervised incremental NMF (INMF) approach under a novel constraint NMF criterion, which aims to cluster within class samples tightly and augment the between-class distance simultaneously. Our incremental strategy utilizes the supervised local features, which are considered as the short-range phenomena of face images, for face classifications. Two public available face databases, namely FERET and CMU PIE face databases, are selected for evaluation. Experimental results show that our INMF method outperforms PCA [6], NMF [4], and BNMF [19] approaches in both nonincremental learning and incremental learning of face recognition.

The rest of this paper is organized as follows: Section 2 briefly reviews the related works. Theoretical analysis and INMF algorithm design are given in Section 3. Experimental results are reported in Section 4. Finally, Section 5 draws the conclusions.

## 2. Related work

This section briefly introduces PCA [6], NMF [4], and BNMF [19] methods. Details are as follows.

### 2.1. PCA

Principal component analysis (PCA), also called eigenface method, is a popular statistic appearance-based linear method for dimensionality reduction in face recognition. The theory used in PCA is based on Karhunen-Loeve transform. It performs the eigenvalue

decomposition on the total scatter matrix  $S_i$  and then selects the large principal components (eigenfaces) to account for most distributions. All face images can be expressed by the linear combinations of these basis images (eigenfaces). However, PCA is not able to exploit all of the feature classification information and how to choose the principal component elements is still a problem. Therefore, PCA cannot give satisfactory performance in pattern recognition tasks.

## 2.2. NMF

NMF aims to find nonnegative matrices  $W$  and  $H$  such that

$$V_{m \times n} \stackrel{\text{NMF}}{\approx} W_{m \times r} H_{r \times n}, \quad (2.1)$$

where matrix  $V$  is also a nonnegative matrix generated by total  $n$  training images. Each column of  $W$  is called basis image, while  $H$  is the coefficient matrix. The basis number  $r$  is usually chosen less than  $n$  for dimensionality reduction. The divergence between  $V$  and  $WH$  is defined as

$$F = \sum_{ij} \left( V_{ij} \log \frac{V_{ij}}{(WH)_{ij}} - V_{ij} + (WH)_{ij} \right). \quad (2.2)$$

NMF (2.1) is equivalent to the following optimization problem:

$$\min_{W, H} F, \quad \text{s.t. } W \geq 0, H \geq 0, \quad \sum_i W_{ik} = 1, \quad \forall k. \quad (2.3)$$

The minimization problem (2.3) can be solved using the following iterative formulae, which converge to a local minimum:

$$W_{ij} \leftarrow W_{ij} \sum_k \frac{V_{ik}}{(WH)_{ik}} H_{jk}, \quad W_{ij} \leftarrow \frac{W_{ij}}{\sum_k W_{kj}}, \quad H_{ij} \leftarrow H_{ij} \sum_k W_{ki} \frac{V_{kj}}{(WH)_{kj}}. \quad (2.4)$$

## 2.3. BNMF

The basic idea of BNMF is to perform NMF on  $c$  small matrices  $V^{(i)} \in \mathbb{R}^{m \times n_0}$  ( $i = 1, 2, \dots, c$ ), namely

$$(V^{(i)})_{m \times n_0} \stackrel{\text{NMF}}{\approx} (W^{(i)})_{m \times r_0} (H^{(i)})_{r_0 \times n_0}, \quad i = 1, 2, \dots, c, \quad (2.5)$$

where  $V^{(i)}$  contains  $n_0$  training images of the  $i$ th class, and  $c$  is the number of classes. BNMF is yielded from (2.5) as follows:

$$V_{m \times n} \stackrel{\text{BNMF}}{\approx} W_{m \times r} H_{r \times n}, \quad (2.6)$$

where  $r = cr_0$ ,  $V_{m \times n} = [V^{(1)} V^{(2)} \dots V^{(c)}]$ ,  $W_{m \times r} = [W^{(1)} W^{(2)} \dots W^{(c)}]$ ,  $H_{r \times n} = \text{diag}(H^{(1)}, H^{(2)}, \dots, H^{(c)})$ , and  $n (= cn_0)$  is the total number of training images.

### 3. Proposed INMF

To overcome the drawbacks of existing NMF-based methods, this section proposes a novel incremental NMF (INMF) approach, which is based on a new constraint NMF criterion and our previous block technique [19]. Details are discussed below.

#### 3.1. Constraint NMF criterion

The objective of our INMF is to impose supervised class information on NMF such that between-class distances increase, while the within-class distances simultaneously decrease. To this end, we define the within-class scatter matrix  $S_w^{(i)}$  of the  $i$ th coefficient matrix  $H^{(i)} \in \mathbb{R}^{r_0 \times n_0}$  as

$$S_w^{(i)} = \frac{1}{n_0} \sum_{j=1}^{n_0} (H_j^{(i)} - U^{(i)})(H_j^{(i)} - U^{(i)})^T, \quad (3.1)$$

where  $U^{(i)} = (1/n_0) \sum_{j=1}^{n_0} H_j^{(i)}$  is the mean column vector of the  $i$ th class. The within-class samples of the  $k$ th class will cluster tightly as  $\text{tr}(S_w^{(k)})$  becomes small.

Assume  $\tilde{U}^{(i)}$  is an enlarging vector of  $U^{(i)}$ , that is,  $\tilde{U}^{(i)} = (1+t)U^{(i)}$  with  $t > 0$ . Then we have

$$\|U^{(i)} - U^{(j)}\| < (1+t)\|U^{(i)} - U^{(j)}\| = \|\tilde{U}^{(i)} - \tilde{U}^{(j)}\|. \quad (3.2)$$

Inequality (3.2) implies that between-class distances are increased as the mean vectors of classes in  $H$  are enlarged.

Based on above analysis, we define a constraint divergence criterion function for the  $k$ th class as follows:

$$F^{(k)} = \sum_{ij} \left( V_{ij}^{(k)} \log \frac{V_{ij}^{(k)}}{(WH)_{ij}^{(k)}} - V_{ij}^{(k)} + (WH)_{ij}^{(k)} \right) + \alpha \text{tr}(S_w^{(k)}) - \beta \|U^{(k)}\|_2^2, \quad (3.3)$$

where parameters  $\alpha, \beta > 0$  and  $k = 1, 2, \dots, c$ .

Our entire INMF criterion function is then designed as below:

$$F = \sum_{k=1}^c F^{(k)} = \sum_{ijk} \left( V_{ij}^{(k)} \log \frac{V_{ij}^{(k)}}{(WH)_{ij}^{(k)}} - V_{ij}^{(k)} + (WH)_{ij}^{(k)} \right) + \sum_k (\alpha \text{tr}(S_w^{(k)}) - \beta \|U^{(k)}\|_2^2). \quad (3.4)$$

Based on criterion (3.4), the following constraint NMF (CNMF) update rules (3.5)–(3.7) will be derived in the next subsection. We can show that the iterative formulae (3.5)–(3.7) converge to a local minimum as well:

$$W_{ij}^{(k)} \leftarrow W_{ij}^{(k)} \sum_l \frac{V_{il}^{(k)}}{(WH)_{il}^{(k)}} H_{jl}^{(k)}, \quad (3.5)$$

$$W_{ij}^{(k)} \leftarrow \frac{W_{ij}^{(k)}}{\sum_l W_{lj}^{(k)}}, \quad (3.6)$$

$$H_{ij}^{(k)} \leftarrow \frac{-b + \sqrt{b^2 - 4ad}}{2a}, \quad (3.7)$$

where  $a = 2\alpha - \beta/n_k^2$ ,  $b = -(2\alpha + \beta/n_k)U_i^{(k)} + 1$ ,  $d = -H_{ij}^{(k)} \sum_l V_{lj}^{(k)} W_{li}^{(k)} / (W^{(k)} H^{(k)})_{ij}$ , and  $U_i^{(k)}$  is the  $i$ th entry of vector  $U^{(k)}$ ,  $k = 1, 2, \dots, c$ .

So, our entire INMF is performed as follows:

$$[V^{(1)} \ V^{(2)} \ \dots \ V^{(c)}] \stackrel{\text{INMF}}{\approx} [W^{(1)} \ W^{(2)} \ \dots \ W^{(c)}] \begin{bmatrix} H^{(1)} & & & \\ & H^{(2)} & & \\ & & \ddots & \\ & & & H^{(c)} \end{bmatrix}, \quad (3.8)$$

where

$$(V^{(i)})_{m \times n_0} \stackrel{\text{CNMF}}{\approx} (W^{(i)})_{m \times r_0} (H^{(i)})_{r_0 \times n_0}, \quad i = 1, 2, \dots, c. \quad (3.9)$$

### 3.2. Convergence of proposed constraint NMF

This subsection reports how to derive the iterative formulae (3.5)–(3.7) and discusses their convergences under constraint NMF criterion (3.3).

*Definition 3.1* (see [5]).  $J(Q, \tilde{Q})$  is called an auxiliary function for  $E(Q)$ , if  $J(Q, \tilde{Q})$  satisfies

$$J(Q, \tilde{Q}) \geq E(Q), \quad J(Q, Q) = E(Q), \quad (3.10)$$

where  $Q, \tilde{Q}$  are matrices with the same size.

**Lemma 3.2** (see [5]). *If  $J(Q, \tilde{Q})$  is an auxiliary function for  $E(Q)$ , then  $E(Q)$  is a nonincreasing function under the update rule*

$$Q^{i+1} = \arg \min_Q J(Q, Q^i). \quad (3.11)$$

To obtain iterative rule (3.7) and prove its convergence, one first constructs an auxiliary function for  $F$  with fixed  $W$ .

**Theorem 3.3.** *If  $F^{(k)}(H^{(k)})$  is the value of criterion function (3.3) with fixed  $W^{(k)}$ , then  $G^{(k)}(H^{(k)}, \tilde{H}^{(k)})$  is an auxiliary function for  $F^{(k)}(H^{(k)})$ , where*

$$\begin{aligned} G^{(k)}(H^{(k)}, \tilde{H}^{(k)}) &= \sum_{ij} (V_{ij}^{(k)} \log V_{ij}^{(k)} - V_{ij}^{(k)} + (W^{(k)} H^{(k)})_{ij}) \\ &\quad - \sum_{ijl} V_{ij}^{(k)} \frac{W_{il}^{(k)} \tilde{H}_{lj}^{(k)}}{\sum_l W_{il}^{(k)} \tilde{H}_{lj}^{(k)}} \left( \log (W_{il}^{(k)} H_{lj}^{(k)}) - \log \frac{W_{il}^{(k)} \tilde{H}_{lj}^{(k)}}{\sum_l W_{il}^{(k)} \tilde{H}_{lj}^{(k)}} \right) \\ &\quad + \alpha \text{tr}(S_w^{(k)}) - \beta \|U^{(k)}\|_2^2. \end{aligned} \quad (3.12)$$

*Proof.* It can be directly verified that  $G^{(k)}(H^{(k)}, H^{(k)}) = F^{(k)}(H^{(k)})$ . So we just need show the inequality  $G^{(k)}(H^{(k)}, \tilde{H}^{(k)}) \geq F^{(k)}(H^{(k)})$ . To this end, we will use the convex function  $y = \log x$ . For all  $i, j$ , and  $\sum_l \sigma_{ijl} = 1$ , it holds that

$$-\log \left( \sum_l W_{il}^{(k)} H_{lj}^{(k)} \right) \leq -\sum_l \sigma_{ijl} \log \frac{W_{il}^{(k)} H_{lj}^{(k)}}{\sigma_{ijl}}. \quad (3.13)$$

Substituting  $\sigma_{ijl} = W_{il}^{(k)} \widetilde{H}_{lj}^{(k)} / \sum_l W_{il}^{(k)} \widetilde{H}_{lj}^{(k)}$  into the above inequality, we have

$$-\log \left( \sum_l W_{il}^{(k)} H_{lj}^{(k)} \right) \leq -\sum_l \frac{W_{il}^{(k)} \widetilde{H}_{lj}^{(k)}}{\sum_l W_{il}^{(k)} \widetilde{H}_{lj}^{(k)}} \left( \log W_{il}^{(k)} H_{lj}^{(k)} - \log \frac{W_{il}^{(k)} \widetilde{H}_{lj}^{(k)}}{\sum_l W_{il}^{(k)} \widetilde{H}_{lj}^{(k)}} \right). \quad (3.14)$$

Therefore,  $G^{(k)}(H^{(k)}, \widetilde{H}^{(k)}) \geq F^{(k)}(H^{(k)})$ . This concludes the theorem immediately.  $\square$

Obviously, the function  $G(H, \widetilde{H}) = \sum_k G^{(k)}(H^{(k)}, \widetilde{H}^{(k)})$  is also an auxiliary function for the entire constraint NMF criterion  $F(H) = \sum_k F^{(k)}(H^{(k)})$ . Lemma 3.2 indicates that  $F(H)$  is nonincreasing under the update rule (3.11). Let  $\partial G(H, \widetilde{H}) / \partial H_{ij}^{(k)} = 0$  and we have

$$\begin{aligned} \frac{\partial G(H, \widetilde{H})}{\partial H_{ij}^{(k)}} &= \frac{\partial G^{(k)}(H^{(k)}, \widetilde{H}^{(k)})}{\partial H_{ij}^{(k)}} \\ &= -\sum_l V_{lj}^{(k)} \frac{W_{li}^{(k)} \widetilde{H}_{ij}^{(k)}}{\sum_l W_{li}^{(k)} \widetilde{H}_{ij}^{(k)}} \frac{1}{H_{ij}^{(k)}} + \sum_l W_{li}^{(k)} + 2\alpha(H_{ij}^{(k)} - U_i^{(k)}) - \frac{\beta}{n_k} \left( \frac{1}{n_k} H_{ij}^{(k)} + U_i^{(k)} \right) \\ &= 0. \end{aligned} \quad (3.15)$$

From the above equation, it directly induces the iterative formula (3.7), and lemma 3.2 demonstrates that (3.7) converges to a local minimum. For update rule (3.5)-(3.6), the proof is similar to that of update rule (3.7) using the following auxiliary function with fixed  $H$ :

$$\begin{aligned} G(W, \widetilde{W}) &= \sum_k G^{(k)}(W^{(k)}, \widetilde{W}^{(k)}) \\ &= \sum_{ijk} \left( V_{ij}^{(k)} \log V_{ij}^{(k)} - V_{ij}^{(k)} + (W^{(k)} H^{(k)})_{ij} \right) \\ &\quad - \sum_{ijkl} V_{ij}^{(k)} \frac{\widetilde{W}_{il}^{(k)} H_{lj}^{(k)}}{\sum_l \widetilde{W}_{il}^{(k)} H_{lj}^{(k)}} \left( \log (W_{il}^{(k)} H_{lj}^{(k)}) - \log \frac{\widetilde{W}_{il}^{(k)} H_{lj}^{(k)}}{\sum_l \widetilde{W}_{il}^{(k)} H_{lj}^{(k)}} \right) \\ &\quad + \sum_k \left( \alpha \operatorname{tr}(S_w^{(k)}) - \beta \|U^{(k)}\|_2^2 \right). \end{aligned} \quad (3.16)$$

### 3.3. Incremental learning

From the above analysis, our incremental learning algorithm is designed as follows:

(i) *Sample incremental learning*. As a new training sample  $x_0$  of the  $i$ th class is added to training set, we denote that  $\widetilde{V}^{(i)} = [V^{(i)}, x_0]$ . Thus the training image matrix becomes

$$\widetilde{V} = [V^{(1)} \ \dots \ \widetilde{V}^{(i)} \ \dots \ V^{(c)}]. \quad (3.17)$$

In this case, it only needs to perform CNMF on matrix  $\tilde{V}^{(i)}$ , that is,  $\tilde{V}^{(i)} \stackrel{\text{CNMF}}{\approx} \tilde{W}^{(i)} \tilde{H}^{(i)}$ . The rest decompositions such as  $V^{(k)} \stackrel{\text{CNMF}}{\approx} W^{(k)} H^{(k)} (k \neq i)$  need not implement repetitive computation. So, sample incremental learning can be performed as follows:

$$\tilde{V} \stackrel{\text{INMF}}{\approx} \tilde{W} \tilde{H} = \begin{bmatrix} W^{(1)} & \dots & \tilde{W}^{(i)} & \dots & W^{(c)} \end{bmatrix} \begin{bmatrix} H^{(1)} & & & & \\ & \ddots & & & \\ & & \tilde{H}^{(i)} & & \\ & & & \ddots & \\ & & & & H^{(c)} \end{bmatrix}. \quad (3.18)$$

(ii) *Class incremental learning.* As a new class, denoted by matrix  $V^{(c+1)}$ , is added to the current training set, it forms a new training image matrix as

$$\tilde{V} = [V^{(1)} \dots V^{(c)} \mid V^{(c+1)}]. \quad (3.19)$$

The incremental learning settings are similar to the first item (i) that all decompositions  $V^{(k)} \stackrel{\text{CNMF}}{\approx} W^{(k)} H^{(k)} (k = 1, 2, \dots, c)$  need not compute again. We only need perform CNMF on the matrix  $V^{(c+1)}$ , that is,  $V^{(c+1)} \stackrel{\text{CNMF}}{\approx} W^{(c+1)} H^{(c+1)}$ . Hence, class incremental learning can be implemented as below:

$$\tilde{V} \stackrel{\text{INMF}}{\approx} \tilde{W} \tilde{H} = \begin{bmatrix} W^{(1)} & \dots & W^{(c)} & \mid & W^{(c+1)} \end{bmatrix} \begin{bmatrix} H^{(1)} & & & & \\ & \ddots & & & \\ & & H^{(c)} & & \\ & & & & H^{(c+1)} \end{bmatrix}. \quad (3.20)$$

### 3.4. INMF algorithm design

Based on the above discussions, this subsection will give a detail design on our INMF algorithm for face recognition. The algorithm involves two stages, namely training stage and testing stage. Details are as follows.

#### Training stage

*Step 1.* Perform CNMF (3.9) on matrices  $(V^{(i)})_{m \times n_0}$ ,  $i = 1, 2, \dots, c$ , namely,

$$(V^{(i)})_{m \times n_0} \stackrel{\text{CNMF}}{\approx} (W^{(i)})_{m \times r_0} (H^{(i)})_{r_0 \times n_0}, \quad i = 1, 2, \dots, c. \quad (3.21)$$

*Step 2.* INMF is obtained as

$$V_{m \times n} \stackrel{\text{INMF}}{\approx} W_{m \times r} H_{r \times n}, \quad (3.22)$$

where  $r = cr_0$ ,  $n = cn_0$ , and

$$W_{m \times r} = [W^{(1)} \ W^{(2)} \ \dots \ W^{(c)}], \quad H_{r \times n} = \text{diag}(H^{(1)}, H^{(2)}, \dots, H^{(c)}). \quad (3.23)$$

If there is a new training sample or class added to current training set, then the incremental learning algorithm presented in Section 3.4 is applied to this stage.

*Recognition stage*

*Step 3.* Calculate the coordinates of a testing sample  $\hat{v}$  in the feature space  $\text{span}\{W_1, W_2, \dots, W_r\}$  by  $\hat{h} = W^+ \hat{v}$ , where  $W^+$  is the Moore-Penrose inverse of  $W$ .

*Step 4.* Compute the mean column vector  $\bar{v}_i$  of class  $i$  and its coordinates vector  $h_i = W^+ \bar{v}_i$  ( $i = 1, 2, \dots, c$ ). The testing image  $\hat{v}$  is classified to class  $k$ , if  $d(\hat{h}, h_k) = \min_{1 \leq i \leq c} d(\hat{h}, h_i)$ , where  $d(\hat{h}, h_i)$  denotes the Euclidean distance between vectors  $\hat{h}$  and  $h_i$ .

### 3.5. Sparseness of coefficient matrix $H$

Let  $h \in \mathbb{R}^n$ , define sparseness function with  $L^1$  and  $L^2$  norms [7] by

$$f_{\text{sparse}}(h) = \frac{\sqrt{n} - \|h\|_1 / \|h\|_2}{\sqrt{n} - 1}. \quad (3.24)$$

It can be seen that sparseness function  $f_{\text{sparse}} : \mathbb{R}^n \rightarrow \mathbb{R}$  with range  $[0, 1]$ .

For INMF method, we have the following theorem for each column  $h_i$  of  $H$ .

**Theorem 3.4.** *Sparseness of each column  $h_i$  of  $H$  in INMF has the following estimation:*

$$\frac{\sqrt{cr_0} - \sqrt{r_0}}{\sqrt{cr_0} - 1} \leq f_{\{\text{sparse}\}}(h_i) \leq 1. \quad (3.25)$$

*Proof.* Let

$$h_i = (0, \dots, 0, h_{i1}^{(j)}, \dots, h_{ir_0}^{(j)}, 0, \dots, 0)^T \in \mathbb{R}^r, \quad \tilde{h}_i = (h_{i1}^{(j)}, \dots, h_{ir_0}^{(j)})^T \in \mathbb{R}^{r_0}, \quad (3.26)$$

where  $h_i$  belongs to class  $i$  in  $H$ .

Obviously,

$$\|h_i\|_1 = \|\tilde{h}_i\|_1, \quad \|h_i\|_2 = \|\tilde{h}_i\|_2. \quad (3.27)$$

Moreover,

$$1 \leq \frac{\|\tilde{h}_i\|_1}{\|\tilde{h}_i\|_2} \leq \sqrt{r_0}. \quad (3.28)$$

So, we have

$$\frac{\sqrt{r} - \sqrt{r_0}}{\sqrt{r} - 1} \leq \frac{\sqrt{r} - \|h_i\|_1 / \|h_i\|_2}{\sqrt{r} - 1} \leq 1. \quad (3.29)$$

It concludes for  $r = cr_0$  that

$$\frac{\sqrt{cr_0} - \sqrt{r_0}}{\sqrt{cr_0} - 1} \leq f_{\text{sparse}}(h_i) \leq 1. \quad (3.30)$$

□

In the experimental section, the parameters are selected as  $r_0 = 4$  and  $c = 120$  using INMF on FERET database. It can be calculated that

$$0.9522 \leq f_{\text{sparse}}(h_i) \leq 1. \quad (3.31)$$

While on CMU PIE database, we select  $r_0 = 4$  and  $c = 68$  and calculate that

$$0.9355 \leq f_{\text{sparse}}(h_i) \leq 1. \quad (3.32)$$

These demonstrate that each column of  $H$  in INMF is highly sparse. Apparently, the coefficient column vectors between different classes in  $H$  are automatically orthogonal.



### 3.6. Computational complexity

This section discusses the computational complexity of our proposed INMF approach. The  $i$ th iterative procedure of proposed INMF includes two parts, namely  $W^{(i)}$  and  $H^{(i)}$ . For each matrix  $V^{(i)}$  the iteration for  $W^{(i)}$  needs  $mr_0(n_0r_0 + 2n_0 + 2)$  multiple times. While for  $H^{(i)}$ , it needs  $n_0r_0(mr_0 + 2m + 10)$  multiple times. Therefore, the total running multiple times of our INMF are

$$T_{\text{INMF}} = (2mn_0r_0^2 + 4mn_0r_0 + 2mr_0 + 10n_0r_0)c = \frac{2mnr^2}{c^2} + \frac{4mnr}{c} + \frac{10nr}{c} + 2mr. \quad (3.33)$$

Similar to INMF, we can obtain the running multiple times of NMF approach as  $T_{\text{NMF}} = 2mnr^2 + 4mnr + 2mr + 2nr$ . It can be seen that the computational complexity of our INMF method is greatly lower than that of NMF.

## 4. Experimental results

In this section, FERET and CMU PIE databases are selected to evaluate the performance of our INMF method along with BNMF, NMF, and PCA methods. All images in two databases are aligned by the centers of eyes and mouth and then normalized with resolution  $112 \times 92$ . The original images with resolution  $112 \times 92$  are reduced to wavelet feature face with resolution  $30 \times 25$  after two-level D4 wavelet decomposition. If there are negative pixels in the wavelet faces, we will transform them into nonnegative faces with simple translations. The nearest neighbor classifier using Euclidean distance is exploited here. In the following experiments, the parameters are set to  $r = 120$  for NMF,  $r_0 = 4$  for BNMF and INMF,  $\alpha = 10^{-4}$ ,  $\beta = 10^{-3}$  for INMF. The stopping condition of iterative update is

$$\frac{F^{(n-1)} - F^{(n)}}{F^{(n)}} \leq \delta, \quad (4.1)$$

where  $F^{(n)}$  is the  $n$ th update criterion function defined in (3.3), the threshold  $\delta$  is set to  $10^{-12}$ . We stop the iteration if stopping condition (4.1) is met or if exceeding 1000 times iteration.

### 4.1. Face databases

In FERET database, we select 120 people, 6 images for each individual. The six images are extracted from 4 different sets, namely Fa, Fb, Fc, and duplicate. Fa and Fb are sets of images taken with the same camera at the same day but with different facial expressions. Fc is a set of images taken with different camera at the same day. Duplicate is a set of images taken around 6–12 months after the day taking the Fa and Fb photos. Details of the characteristics of each set can be found in [3]. Images from one individual are shown in Figure 1.

CMU PIE database includes totally 68 people. There are 13 pose variations ranging from full right-profile image to full left-profile image and 43 different lighting conditions, 21 flashes with ambient light on or off. In our experiment, for each person, we select 56 images including 13 poses with neutral expression and 43 different lighting conditions in frontal view. Part images of one person are shown in Figure 2.



Figure 1: Images of one person from FERET database.



Figure 2: Part images of one person from CMU PIE database.

#### 4.2. Basis face images

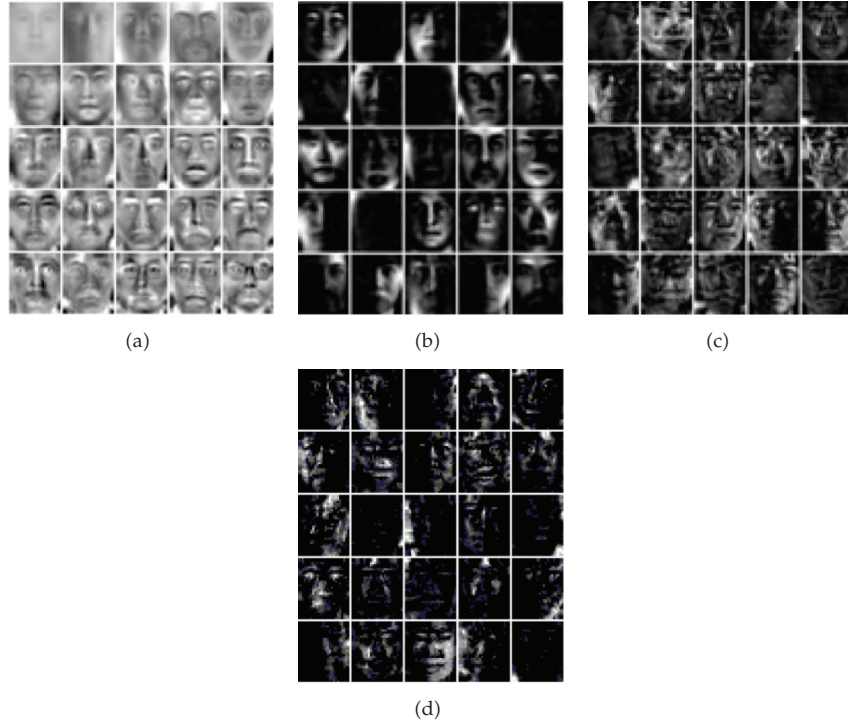
This section shows the basis images of the training set learnt by PCA, NMF, BNMF, and INMF approaches. Figure 3 shows 25 basis images of each approach on CMU PIE database. It can be seen that the bases of all methods are additive except for PCA. PCA extracts the holistic facial features. INMF learns more local features than NMF and BNMF. Moreover, the greater number of basis image is, the more localization is learnt in all NMF-based approaches.

#### 4.3. Results on FERET database

This section reports the experimental results with nonincremental learning and incremental learning on FERET database. All methods use the same training and testing face images. The experiments are repeated 10 times; and the average accuracies under different training number, along with the mean running times, are recorded.

##### 4.3.1. Nonincremental learning

We randomly select  $n$  ( $n = 2, 3, 4, 5$ ) images from each person for training, while the rest of  $(6-n)$  images of each individual for testing. The average accuracies of training samples ranging from 2 to 5 are recorded in Table 1 and plotted in Figure 4(a). The recognition accuracies of INMF, BNMF, NMF, and PCA are 66.73%, 66.07%, 64.44%, and 34.33%, respectively, with 2 training images. The performance for each method is improved when the number of training images increases. When the number of training images is equal to 5, the recognition accuracies of INMF, BNMF, NMF, and PCA are 83.08%, 81.67%, 80.25%, and 37.58%, respectively.



**Figure 3:** Comparisons on basis images of PCA, NMF, BNMF, and INMF (from left to right), respectively, on CMU PIE database results.

In addition, Table 2 gives the comparisons on average time-consuming in three NMF-based approaches. It can be seen that our INMF method gives the best performance for all cases of nonincremental learning on FERET database.

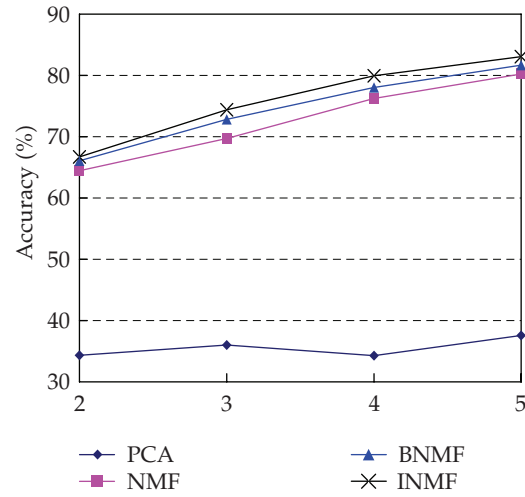
#### 4.3.2. Class incremental learning

For 119 people, we randomly select 3 images from each individual for training and then add a new class to the training set. NMF must conduct repeated learning while BNMF and INMF need merely perform incremental training on the new added class. The average accuracies and the mean running times are recorded in Table 3 (plotted in Figure 6(a)) and Table 4, respectively.

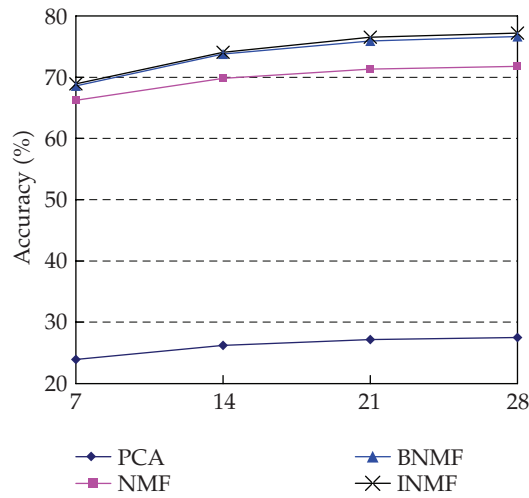
Compared with the NMF and BNMF approaches, the proposed method gives around 5% and 1.5% accuracy improvements, respectively. The running time of INMF is around 2 times and 219 times faster than that of NMF with 119 and 120 individuals for training and class-incremental learning, respectively. Above all, our INMF gives the best performance on FERET database.

#### 4.4. Results on CMU PIE database

The experimental setting on CMU PIE database is similar to that of FERET database. It also includes two parts, namely nonincremental training and incremental learning. The



(a) Training number



(b) Training number

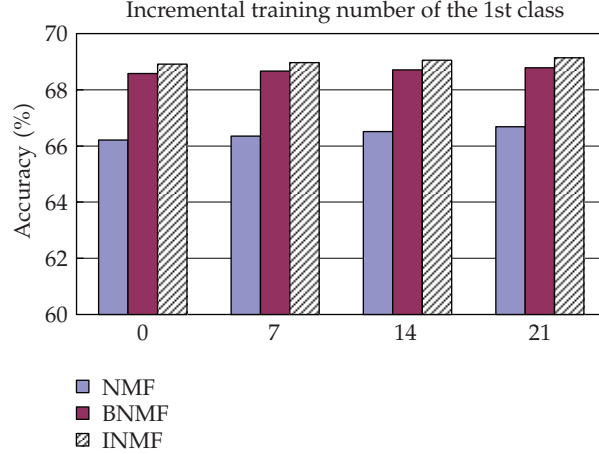
**Figure 4:** Accuracy comparisons on (a) FERET and (b) CMU PIE databases.**Table 1:** Accuracy comparisons on FERET database.

Training number	PCA	NMF	BNMF	INMF
2	34.33%	64.44%	66.07%	66.73%
3	36.00%	69.72%	72.81%	74.39%
4	34.29%	76.25%	78.04%	78.92%
5	37.58%	80.25%	81.67%	83.08%

experiments are repeated 10 times and the average accuracies under different training number, along with the mean running times, are recorded for comparisons. Details are as follows.

**Table 2:** Running times (s) on FERET database.

Training number	NMF(s)	BNMF(s)	INMF(s)
2	51.14	30.69	21.12
3	74.58	57.02	33.68
4	100.12	75.81	45.34
5	122.61	89.34	56.30

**Figure 5:** Comparisons on sample incremental learning.**Table 3:** Accuracy comparisons on class incremental learning.

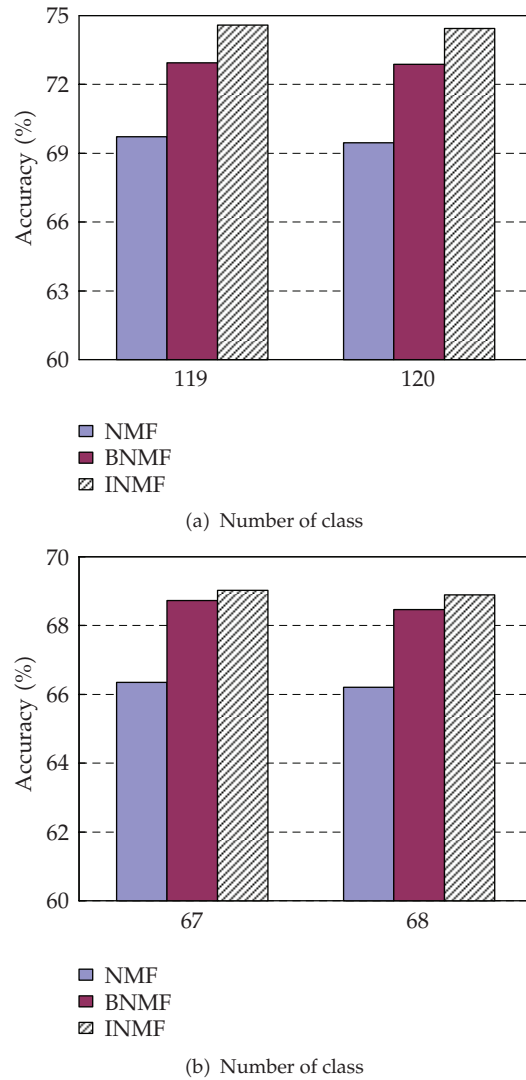
Number of class	NMF	BNMF	INMF
119	69.72%	72.94%	74.59%
120	69.45%	72.88%	74.44%

#### 4.4.1. Nonincremental learning

For each individual,  $n$  ( $n = 7, 14, 21, 28$ ) images are randomly selected for training, while the rest ( $56 - n$ ) images for testing. The average recognition rates and mean running times are tabulated in Table 5 (plotted in Figure 4(b)) and Table 6, respectively. It can be seen that the recognition accuracies of INMF, BNMF, NMF, and PCA are 68.91%, 68.58%, 66.21%, and 23.94%, respectively with training number 7. When the number of training images is equal to 28, the recognition accuracies of INMF, BNMF, NMF, and PCA are 77.18%, 76.64%, 71.77%, and 27.51%, respectively. Compared with the PCA and NMF methods, the proposed method gives around 49% and 5% accuracy improvements, respectively. The performance of INMF is slightly better than that of BNMF. However, the computational efficiency of INMF greatly outperforms BNMF.

#### 4.4.2. Sample incremental learning

We randomly select 7 images from each person for training, and the rest 49 images for testing. In the sample-incremental learning stage, 7, 14, and 21 images of the first individual are added



**Figure 6:** Class incremental learning comparisons on (a) FERET and (b) CUM PIE databases.

to the training set, respectively, while the training images from the rest individuals are kept unchanged. Table 7 (Figure 5) and Table 8 show the average recognition accuracies and the mean running times, respectively. Experimental results show that our INMF method gives the best performance for all cases.

#### 4.4.3. Class incremental learning

For 67 people, we randomly select 7 images from each individual for training and then add a new class to the training set. NMF should conduct repetitive learning. BNMF and INMF need merely to perform incremental learning on the new added class. The average recognition rates and the mean running times are recorded in Table 9 (plotted in Figure 6(b)) and Table 10,

**Table 4:** Running times (s) on class incremental learning.

Number of class	NMF(s)	BNMF(s)	INMF(s)
119	72.57	56.12	32.34
120	74.58	0.86	0.34

**Table 5:** Accuracy comparisons on CMU PIE database.

Training number	PCA	NMF	BNMF	INMF
7	23.94%	66.21%	68.58%	68.91%
14	26.24%	69.82%	73.79%	74.07%
21	27.15%	71.33%	75.93%	76.55%
28	27.51%	71.77%	76.64%	77.18%

**Table 6:** Running times (s) on CMU PIE database.

Training number	NMF(s)	BNMF(s)	INMF(s)
7	79.58	57.35	37.24
14	144.52	114.61	89.50
21	208.20	164.18	124.45
28	267.97	215.94	176.23

**Table 7:** Accuracy comparisons on sample incremental learning.

Incremental training number	NMF	BNMF	INMF
0	66.21%	68.58%	68.91%
7	66.35%	68.66%	68.97%
14	66.51%	68.71%	69.05%
21	66.68%	68.78%	69.14%

**Table 8:** Running times on sample incremental learning.

Incremental training number	NMF (s)	BNMF (s)	INMF (s)
0	79.58	57.35	37.24
7	79.77	2.34	1.20
14	80.02	3.12	2.36
21	80.23	4.53	3.21

**Table 9:** Comparisons on class incremental learning.

Number of class	NMF	BNMF	INMF
67	66.35%	68.73%	69.02%
68	66.21%	68.46%	68.89%

respectively. Experimental results show that INMF outperforms BNMF and NMF in both recognition rates and computational efficiency.

**Table 10:** Running time (s) on class incremental learning.

Number of class	NMF(s)	BNMF(s)	INMF(s)
67	72.45	55.19	34.28
68	79.58	1.36	0.70

## 5. Conclusions

This paper proposed a novel constraint INMF method to address the time-consuming problem and incremental learning problem of existing NMF-based approaches for face recognition. INMF has some good properties, such as low computational complexity; sparse coefficient matrix; orthogonal coefficient column vectors between different classes in coefficient matrix  $H$ ; especially for incremental learning, and so on. Experimental results on FERET and CMU PIE face database show that INMF outperforms PCA, NMF, and BNMF approaches in nonincremental learning and incremental learning.

## Acknowledgments

This work is supported by NSF of China (60573125, 60603028) and in part by the Program for New Century Excellent Talents of Educational Ministry of China (NCET-06-0762), NSF of Chongqing CSTC (CSTC2007BA2003). The authors would like to thank the US Army Research Laboratory for contribution of the FERET database and CMU for the CMU PIE database.

## References

- [1] R. Chellappa, C. L. Wilson, and S. Sirohey, "Human and machine recognition of faces: a survey," *Proceedings of the IEEE*, vol. 83, no. 5, pp. 705–741, 1995.
- [2] W. Zhao, R. Chellappa, A. Rosenfeld, and J. Phillips, "Face recognition: a literature survey," Tech. Rep. CFAR-TR00-948, University of Maryland, College Park, Md, USA, 2000.
- [3] R. Brunelli and T. Poggio, "Face recognition: features versus templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 10, pp. 1042–1052, 1993.
- [4] 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.
- [5] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS '01)*, vol. 13, pp. 556–562, Vancouver, Canada, December 2001.
- [6] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.
- [7] S. Z. Li, X. W. Hou, H. J. Zhang, and Q. S. Cheng, "Learning spatially localized, parts-based representation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '01)*, vol. 1, pp. 207–212, Kauai, Hawaii, USA, December 2001.
- [8] S. Wild, J. Curry, and A. Dougherty, "Improving non-negative matrix factorizations through structured initialization," *Pattern Recognition*, vol. 37, no. 11, pp. 2217–2232, 2004.
- [9] I. Buciu and I. Pitas, "A new sparse image representation algorithm applied to facial expression recognition," in *Proceedings of the 14th IEEE Workshop on Machine Learning for Signal Processing (MLSP '04)*, pp. 539–548, Sao Luis, Brazil, September–October 2004.
- [10] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.
- [11] Y. Xue, C. S. Tong, W.-S. Chen, and W. Zhang, "A modified non-negative matrix factorization algorithm for face recognition," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR '06)*, vol. 3, pp. 495–498, Hong Kong, August 2006.



- [12] 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.
- [13] I. Buciu and I. Pitas, "NMF, LNMF, and DNMF modeling of neural receptive fields involved in human facial expression perception," *Journal of Visual Communication and Image Representation*, vol. 17, no. 5, pp. 958–969, 2006.
- [14] A. Pascual-Montano, J. M. Carazo, K. Kochi, D. Lehmann, and R. D. Pascual-Marqui, "Nonsmooth nonnegative matrix factorization (nsNMF)," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 3, pp. 403–415, 2006.
- [15] I. Kotsia, S. Zafeiriou, and I. Pitas, "A 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.
- [16] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural Computation*, vol. 19, no. 10, pp. 2756–2779, 2007.
- [17] C.-J. Lin, "On the convergence of multiplicative update algorithms for nonnegative matrix factorization," *IEEE Transactions on Neural Networks*, vol. 18, no. 6, pp. 1589–1596, 2007.
- [18] T. Zhang, B. Fang, Y. Y. Tang, G. He, and J. Wen, "Topology preserving non-negative matrix factorization for face recognition," *IEEE Transactions on Image Processing*, vol. 17, no. 4, pp. 574–584, 2008.
- [19] B. B. Pan, W. S. Chen, and C. Xu, "Incremental learning of face recognition based on block non-negative matrix factorization," 2008 (Chinese), to appear in *Computer Application Research*.