

Research Article

Incremental Graph Regulated Nonnegative Matrix Factorization for Face Recognition

Zhe-Zhou Yu, Yu-Hao Liu, Bin Li, Shu-Chao Pang, and Cheng-Cheng Jia

College of Computer Science and Technology, Jilin University, Changchun 130012, China

Correspondence should be addressed to Zhe-Zhou Yu; yuzz@jlu.edu.cn

Received 25 March 2014; Revised 4 May 2014; Accepted 4 May 2014; Published 19 May 2014

Academic Editor: Jen-Tzung Chien

Copyright © 2014 Zhe-Zhou Yu 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.

In a real world application, we seldom get all images at one time. Considering this case, if a company hired an employee, all his images information needs to be recorded into the system; if we rerun the face recognition algorithm, it will be time consuming. To address this problem, In this paper, firstly, we proposed a novel subspace incremental method called incremental graph regularized nonnegative matrix factorization (IGNMF) algorithm which imposes manifold into incremental nonnegative matrix factorization algorithm (INMF); thus, our new algorithm is able to preserve the geometric structure in the data under incremental study framework; secondly, considering we always get many face images belonging to one person or many different people as a batch, we improved our IGNMF algorithms to Batch-IGNMF algorithms (B-IGNMF), which implements incremental study in batches. Experiments show that (1) the recognition rate of our IGNMF and B-IGNMF algorithms is close to GNMF algorithm while it runs faster than GNMF. (2) The running times of our IGNMF and B-IGNMF algorithms are close to INMF while the recognition rate outperforms INMF. (3) Comparing with other popular NMF-based face recognition incremental algorithms, our IGNMF and B-IGNMF also outperform then both the recognition rate and the running time.

1. Introduction

Nonnegative matrix factorization (NMF) is a widely used method for low-rank approximation of a nonnegative matrix (matrix with only nonnegative entries), where nonnegative constraints are imposed on factor matrices in the decomposition. There are large bodies of past work on NMF [1]. Lee and Seung [2, 3] proposed NMF for learning parts of faces, and in their work the reconstruction error function $F(U, V)$ is introduced: $F(U, V) = \|X - UV\|_F^2$, where X denotes the data matrix, U can be considered as the basic matrix, and V can be considered as the coefficient matrix; all elements of X , U , and V are nonnegative. Sparse coding is a famous parts-based representation method, by minimizing a L_1 regularization-related objective function of NMF-based algorithms, sparse constraints can be achieved. Hoyer [4] proposed a method by keeping L_2 norm unchanged in each iteration, but L_1 norm set to achieve desired sparseness. Li et al. [5] proposed another sparse representation method which focused on sparse NMF algorithm with Kullback-Leibler based cost function. Discrimination method was also

introduced into NMF algorithm. Wang et al. [6] proposed a Fisher nonnegative matrix factorization which introduces Fisher constraint (discrimination) method into NMF algorithm. later, Nikitidis et al. [7] introduced subclass discriminant into NMF algorithm, by separating each class into several subclasses; this method was able to ensure that the underlying data distribution in each subclass is unimodal. Because the convergence of canonical NMF algorithms is slow, gradient descent based methods are introduced to NMF to improve its speed of convergence. Guan et al. [8] applied Nesterov's optimal gradient method to alternatively optimize one factor with another. By introducing fast gradient descent method into search of the optimal step size for gradient descent based NMF algorithm, Guan et al. [9] introduced nonnegative patch alignment framework (NPAF) and nonnegative discriminative locality alignment (NDLA). Canonical NMF algorithm aims to minimize the Euclidean distance or the Kullback-Leibler distance between the data matrix X and its reconstruction matrix $U \times V$. By introducing Manhattan distance into NMF algorithm, Guan et al. [10] introduced Manhattan nonnegative matrix

factorization (MahNMF), then rank-one residual iteration method and Nesterov's smoothing method are both introduced to optimizing MahNMF. The geometrical information of the original data space is an important information for face recognition. Zhang et al. [11] proposed a novel topology preserving nonnegative matrix factorization (TPNMF) which considers the gradient distance instead of the Euclidean distance or the Kullback-Leibler distance. By constructing the intrinsic and penalty graphs, Liu et al. [12] proposed the projective nonnegative graph embedding (PNGE). Cai et al. [13] proposed graph regularized nonnegative matrix factorization (GNMF) algorithm for face recognition. GNMF imposes manifold into NMF, which makes it able to preserve geometric structure of data after it maps the original image into low dimension space.

NMF-based algorithms are notoriously slow to converge. In practice, if new image comes, algorithm needs to rerun which is time consuming. To address this problem, incremental process used for online study has attracted a lot of attention. During an incremental study process of NMF, once a factorization, such as U and V which mentioned above, is obtained for a group of face images, the representation of U should be updated with a small computational cost, but the canonical NMF and many other algorithms are too expensive to compute U and V every time when new face image comes. Bucak and Günsel [14] introduced an Incremental nonnegative matrix factorization algorithm (INMF), which imposes NMF into incremental study. This makes it possible to represent data content online and reduce dimension significantly. Chen et al. [15] proposed another incremental nonnegative matrix factorization (INMF) for face representation and recognition. In order to distinguish Chen's algorithms [15] from Bucak and Günsel's [14], we called Chen's algorithms CINMF. There are two main differences between CINMF and INMF: first, INMF is an unsupervised method while CINMF is a supervised method; second, INMF can only deal with one new image while CINMF can only deal with a batch of images which belong to a new class. Wang and Lu [16] introduced an incremental orthogonal projective nonnegative matrix factorization algorithm (IOPNMF) which introduced orthogonal projective constraint into incremental NMF. Lefèvre et al. [17] introduced incremental study into Itakura-Saito NMF algorithm; the Itakura-Saito divergence was defined as $d_{IS}(y, x) = \sum_i (y_i/x_i - \log y_i/x_i - 1)$. The proposed algorithm is used in the field of audio source separation. Guan et al. [18] proposed an efficient online RSA-NMF algorithm. RSA-NMF is a NMF algorithm based which introduces Robust Stochastic approximation (RSA) to optimize NMF algorithm. Benefiting from the smartly chosen learning rate and averaging technique, online RSA-NMF algorithm converges at the rate of $O(1/\sqrt{k})$ in each update of the bases. later Guan et al. also gave some extensions on the proposed online RSA-NMF algorithm. Cao et al. [19] introduced an online NMF algorithm. Once we get the factorization of X as $U \times V$, by separating the new data matrix X , and the new coefficient matrix V into two corresponding parts, the inequation can be rewritten as $[X \ \mathbf{x}] \approx U' \times [V' \ \mathbf{v}]$; every iteration, $[V' \ \mathbf{v}]$, can be recomputed with

a small computation cost while U' can be recomputed as $U' = U \times (V')^{-1}$; obviously there is an assumption that $X \approx U \times V$ are full-rank decompositions and V' is invertible.

Although the incremental algorithms we mentioned above dramatically reduce the cost of incremental study, they did not consider the geometrical information of the original data space on the original manifold. The geometrical information of the original data space is an important information for face recognition [20, 21]. Data set often lay in a high dimension manifold, so the intrinsic geodesic distance in the manifold between two data points is more suitable than Euclidean distance. Thus, in this paper, By introducing manifold into Incremental nonnegative matrix factorization algorithm (INMF), first, we propose a new incremental study algorithm called incremental graph regularized nonnegative matrix factorization (IGNMF). Second, as human face images always come in batches, we improve our IGNMF algorithms to Batch-IGNMF algorithms (B-IGNMF). The contributions of the research presented in this paper are as follows.

- (1) We propose a novel subspace incremental method called incremental graph regularized nonnegative matrix factorization algorithm (IGNMF), which is able to preserve discrimination information under incremental study framework.
- (2) We further improve our IGNMF algorithms, thus developing Batch-IGNMF (B-IGNMF), which is able to perform incremental study with image batches no matter whether the batch of images belongs to the same class or different classes.

The remainder of this paper is organized as follows. Section 2 introduces the GNMF algorithm and INMF algorithm. Section 3 presents our IGNMF and B-IGNMF algorithms. Experiments on face databases are reported in Section 4. Section 5 concludes the paper.

2. A Brief Review of GNMF and INMF

In this section, we will give a brief review on graph regularized nonnegative matrix factorization (GNMF) which was proposed by Cai et al. [13] and incremental nonnegative matrix factorization algorithm (INMF) which was proposed by Bucak and Günsel [14]; both algorithms are closely related to our new proposed algorithms.

2.1. GNMF. The geometrical information of the original data space is an important information for face recognition [20, 21]; data set often lay in a high dimension manifold, so the intrinsic geodesic distance in the manifold between two data points is more suitable than Euclidean distance. Thus in this section, we will give a brief review on graph regularized NMF (GNMF) [13].

Given a data matrix $X = [x_1, x_2, \dots, x_k] \in \mathbb{R}^{n \times k}$, there are total k face image vectors and x_i ($1 \leq i \leq k$) denotes a vector of n -dimensional facial image, $U \in \mathbb{R}^{n \times r}$ and $V \in \mathbb{R}^{r \times k}$, so the objective function $F(U, V)$ for GNMF is defined as (1);

by minimizing $F(U, V)$ a factorization, U and V in GNMF, is obtained:

$$F = \|X - UV\|_F^2 + \lambda \text{Tr}(VLV^T). \quad (1)$$

In the above equation $F = \|\cdot\|_F^2$ denotes the matrix Frobenius norm [2]; $\text{Tr}(\cdot)$ denotes the trace of a matrix; $\lambda > 0$ is the regularization parameter; L is called graph Laplacian [22]; $L = D - W$, where W is the weight matrix, $W_{ij} = 1$ if $x_i \in N_p(x_j)$, and D is a diagonal matrix and $D_{ii} = \sum_j W_{ij}$. There are two methods to define W [13, 22].

- (1) Supervised method is achieved by defining $x_i \in N_p(x_j)$ if x_i and x_j belong to the same class.
- (2) Unsupervised method is achieved by defining $N_p(x_i)$ as the set of p nearest neighbors of x_i under the Euclidean distance.

The iterative update algorithm proposed by Cai et al. [13] is as follows (2):

$$\begin{aligned} U_{ij}^{t+1} &= U_{ij}^t \cdot \frac{(XV^T)_{ij}}{(UVV^T)_{ij}} \\ V_{ij}^{t+1} &= V_{ij}^t \cdot \frac{(U^T X + \lambda VW)_{ij}}{(V^T U^T U + \lambda VD)_{ij}}, \end{aligned} \quad (2)$$

where U_{ij} and V_{ij} are the elements of U and V , respectively.

2.2. INMF. Incremental nonnegative factorization (INMF) [14] is a popular incremental study algorithm which is NMF-based. Considering the canonical reconstruction inequation of NMF algorithm as $X \approx U \times V$, and each column of both sides is $X_i \approx U \times V_i$; from this inequation we can find that if we consider each column of the matrix U as the building block of the database and V_i can be considered as the reconstruction coefficient matrix, by summing each building block of U under the coefficient matrix V_i , original image X_i will be approximately reconstructed [2, 3].

Every time when new image \mathbf{x} comes, the inequation of the reconstruction function will become $[X \ \mathbf{x}] \approx U \times [V \ \mathbf{v}]$; by assuming that the previous coefficient matrix V would not change during the incremental process, the computation cost will be reduced significantly.

If we defined the F_k as the cost function corresponding to the NMF as (3) where k denotes the total number of samples before incremental study, then

$$F_k = \|X - UV\|^2 = \sum_{i=1}^n \sum_{j=1}^k [(X_k)_{ij} - (U_k V_k)_{ij}]^2. \quad (3)$$

Everytime, when the $(k + 1)$ th sample \mathbf{x} arrives, the corresponding cost function F_{k+1} can be defined as (4)

$$F_{k+1} = \sum_{i=1}^n \sum_{j=1}^{k+1} [(X_{k+1})_{ij} - (U_{k+1} V_{k+1})_{ij}]^2. \quad (4)$$

Note F_{k+1} can be separated into two parts as follows:

$$\begin{aligned} F_{k+1} &= \sum_{i=1}^n \sum_{j=1}^{k+1} [(X_{k+1})_{ij} - (U_{k+1} V_{k+1})_{ij}]^2 \\ &= \sum_{i=1}^n \sum_{j=1}^k [(X_{k+1})_{ij} - (U_{k+1} V_{k+1})_{ij}]^2 \\ &\quad + \sum_{i=1}^n [\mathbf{x} - (U_{k+1} \mathbf{v})_{i,k+1}]^2, \end{aligned} \quad (5)$$

where \mathbf{x} and \mathbf{v} can be considered as the new coming image and its corresponding coefficient vector. By assuming that the first k columns of V_{k+1} would not change after the incremental study, the first part of (5) can be rewritten as

$$\begin{aligned} &\sum_{i=1}^n \sum_{j=1}^k [(X_{k+1})_{ij} - (U_{k+1} V_{k+1})_{ij}]^2 \\ &\approx F_k = \sum_{i=1}^n \sum_{j=1}^k [(X_k)_{ij} - (U_k V_k)_{ij}]^2. \end{aligned} \quad (6)$$

So F_{k+1} can be rewritten as

$$F_{k+1} = F_k + \sum_{i=1}^n [\mathbf{x} - (U_{k+1} \mathbf{v})_{i,k+1}]^2 = F_k + f. \quad (7)$$

Considering that F_k is the function which is independent of v_i , the partial derivative of $\partial F_k / v_i = 0$. Thus the partial derivative of $\partial F_{k+1} / v_i (= \partial f / v_i)$ and $\partial F_{k+1} / U_{ij}$ can be deduced. Then the update rule of \mathbf{v}_i and U_{ij} can be formulated within the framework of gradient descent algorithm.

In order to save space, here we just list the iterative update algorithms of INMF in (8):

$$\begin{aligned} \mathbf{v}'_i &\leftarrow \mathbf{v}_i \cdot \frac{(U_{k+1}^T \mathbf{x})_i}{(U_{k+1}^T U_{k+1} \mathbf{v})_i}, \\ (U'_{k+1})_{ia} &\leftarrow (U_{k+1})_{ia} \cdot \frac{(X_k V_k^T + X_{k+1} \mathbf{v}^T)_{ia}}{(U_{k+1} U_k V_k^T + U_{k+1} \mathbf{v} \mathbf{v}^T)_{ia}}. \end{aligned} \quad (8)$$

3. IGNMF and B-IGNMF

Bucak and Günsel [14] introduced an incremental nonnegative matrix factorization algorithm (INMF) which imposed the NMF algorithm into incremental study, so INMF inherits the disadvantage of the NMF algorithm; that is, it does not consider the geometric structure in the data. In this section, we introduced an incremental graph regularized nonnegative matrix factorization algorithm (IGNMF), in which manifold is introduced to overcome this limitation.

3.1. IGNMF. Let X_k , U_k , V_k , L_k , and F_k denote the corresponding X , U , V , L , and $F(U, V)$ in (1) under the initial k samples, so the objective function F_k can be rewritten as

$$F_k = \sum_{i=1}^n \sum_{j=1}^k [(X_k)_{ij} - (U_k V_k)_{ij}]^2 + \lambda \text{Tr}(V_k L_k V_k^T). \quad (9)$$

Let X_{k+1} , U_{k+1} , V_{k+1} , L_{k+1} , and F_{k+1} denote the corresponding X , U , V , L , and $F(U, V)$ in (1) when the $(k + 1)$ th sample arrives. So objective function F_{k+1} can be rewritten as

$$F_{k+1} = \sum_{i=1}^n \sum_{j=1}^{k+1} [(X_{k+1})_{ij} - (U_{k+1}V_{k+1})_{ij}]^2 + \lambda \text{Tr}(V_{k+1}L_{k+1}V_{k+1}^T). \quad (10)$$

U can be considered as the basic matrix and V can be considered as the coefficient matrix, so the reconstruction process can be thought as adding columns of matrix U under the coefficient matrix V_i , just as $X_i \approx U \times V_i$ [3]. Thus we make the assumption that during the incremental process, when the $(k + 1)$ th sample arrives, the first k columns of V_{k+1} does not change, which means the first k columns of V_{k+1} is approximately equal to V_k . This assumption would reduce the computation cost significantly. Experiments show that IGDMF would iterate less than 5 times and then the objective function converges to its minimal value, because we just need to update the last column of V_{k+1} meanwhile U_{k+1} needs to be updated completely which will dramatically reduce the cost of incremental study. For more details about this assumption, please refer to [14].

Assuming F_{k+1}^k refers to the objective function corresponding to GDMF representation of the first k sample when the $(k + 1)$ th sample arrives, L_{k+1}^k refers to the $k \times k$ dimensional matrix which equals the first k rows and first k columns of L_{k+1} , and r is a predefined parameter which indicates that the n -dimensional facial image vector maps in to an r -dimensional vector. So F_{k+1}^k can be rewritten as

$$F_{k+1}^k = \sum_{i=1}^n \sum_{j=1}^k [X_{ij} - (U_{k+1}V_{k+1})_{ij}]^2 + \lambda \sum_{l=1}^r \sum_{i=1}^k \sum_{j=1}^k (V_{k+1})_{li} (L_{k+1}^k)_{ij} (V_{k+1}^T)_{jl} \approx \sum_{i=1}^n \sum_{j=1}^k [X_{ij} - (U_{k+1}V_k)_{ij}]^2 + \lambda \sum_{l=1}^r \sum_{i=1}^k \sum_{j=1}^k (V_k)_{li} (L_{k+1}^k)_{ij} (V_k^T)_{jl}. \quad (11)$$

Consequently, objective function F_{k+1} can be rewritten as follows:

$$F_{k+1} = \sum_{i=1}^n \sum_{j=1}^{k+1} [X_{ij} - (U_{k+1}V_{k+1})_{ij}]^2 + \lambda \sum_{l=1}^r \sum_{i=1}^{k+1} \sum_{j=1}^{k+1} (V_{k+1})_{li} (L_{k+1})_{ij} (V_{k+1}^T)_{jl} \approx \sum_{i=1}^n \sum_{j=1}^k [X_{ij} - (U_{k+1}V_k)_{ij}]^2 + \lambda \sum_{l=1}^r \sum_{i=1}^k \sum_{j=1}^k (V_k)_{li} (L_{k+1}^k)_{ij} (V_k^T)_{jl} + \sum_{i=1}^n [\mathbf{x} - (U_{k+1}\mathbf{v})_i]^2 + \lambda \sum_{l=1}^r \left\{ \mathbf{v}^2 (L_{k+1})_{k+1,k+1} + \sum_{j=1}^k [2\mathbf{v}(V_k)_{lj} (L_{k+1})_{j,k+1}] \right\} = F_{k+1}^k + f_{k+1}, \quad (12)$$

where \mathbf{x} means the new coming facial vector and \mathbf{v} means the coefficient vector corresponding to \mathbf{x} .

After constructing the objective function given by (12), gradient descent optimization that yields IGDMF can be performed. The update rule of \mathbf{v} can be formulated within the framework of gradient descent algorithm as follows:

$$\mathbf{v}'_a \leftarrow \mathbf{v}_a - \mu_a \cdot \frac{\partial F_{k+1}}{\partial \mathbf{v}_a}. \quad (13)$$

In (13), $\partial F_{k+1}/\partial \mathbf{v}_a$ is the partial derivative of F_{k+1} with respect to \mathbf{v}_a and $\partial F_{k+1}/\partial \mathbf{v}_a = \partial F_{k+1}^k/\partial \mathbf{v}_a + \partial f_{k+1}/\partial \mathbf{v}_a$, because F_{k+1}^k is the function which is not relevant to \mathbf{v}_a , so $\partial F_{k+1}^k/\partial \mathbf{v}_a = 0$, and we just need to compute the partial derivative of f_{k+1} :

$$\frac{\partial F_{k+1}}{\partial \mathbf{v}_a} = \frac{\partial F_{k+1}^k}{\partial \mathbf{v}_a} + \frac{\partial f_{k+1}}{\partial \mathbf{v}_a} = \frac{\partial}{\partial \mathbf{v}_a} \left\{ \sum_{i=1}^n [\mathbf{x} - (U_{k+1}\mathbf{v})_i]^2 + \lambda \sum_{l=1}^r \left[\sum_{j=1}^k [2\mathbf{v}(V_k)_{lj} (L_{k+1})_{j,k+1} + (V_k)_{j,k+1}^2 (W_{k+1})_{l,k+1}] \right] \right\} = 2[-U_{k+1}^T \mathbf{x} + U_{k+1}^T U_{k+1} \mathbf{v} + \lambda \mathbf{v} (L_{k+1})_{k+1,:}]_a. \quad (14)$$

μ_a is the step size and is calculated by

$$\mu_a = \frac{\mathbf{v}_a}{2[U_{k+1}^T U_{k+1} \mathbf{v} + \lambda \cdot \mathbf{v}(D_{k+1})_{k+1,:}]_a}, \quad (15)$$

where $(L_{k+1})_{k+1,:}$, $(D_{k+1})_{k+1,:}$, and $(W_{k+1})_{k+1,:}$ are vectors which are equal to the $(k+1)$ th row of L_{k+1} , D_{k+1} and D_{k+1} , respectively.

After substituting (14) and (15) into (13), The update rule equation for \mathbf{v}_a yields as

$$\mathbf{v}'_a \leftarrow \mathbf{v}_a \cdot \frac{[U_{k+1}^T \mathbf{x} + \lambda \cdot \mathbf{v}(W_{k+1})_{k+1,:}]_a}{[U_{k+1}^T U_{k+1} \mathbf{v} + \lambda \cdot \mathbf{v}(D_{k+1})_{k+1,:}]_a}. \quad (16)$$

The update rule of $(U_{k+1})_{ia}$ can also be formulated within the framework of gradient descent algorithm as

$$(U'_{k+1})_{ia} \leftarrow (U_{k+1})_{ia} - \eta_{ia} \cdot \frac{\partial F_{k+1}}{\partial (U_{k+1})_{ia}}. \quad (17)$$

In (17), $\partial F_{k+1}/\partial (U_{k+1})_{ia}$ is the partial derivative of F_{k+1} with respect to $(U_{k+1})_{ia}$ and is given as

$$\begin{aligned} & \frac{\partial F_{k+1}}{\partial (U_{k+1})_{ia}} \\ &= \frac{\partial}{\partial (U_{k+1})_{ia}} \left\{ \sum_{i=1}^n \sum_{j=1}^{k+1} [X_{ij} - (U_{k+1} V_{k+1})_{ij}]^2 \right. \\ & \quad \left. + \lambda \sum_{l=1}^r \sum_{i=1}^{k+1} \sum_{j=1}^{k+1} (V_{k+1})_{li} (L_{k+1})_{ij} (V_{k+1})_{jl} \right\} \\ &= \frac{\partial}{\partial (U_{k+1})_{ia}} \sum_{i=1}^n \sum_{j=1}^{k+1} [X_{ij} - (U_{k+1} V_{k+1})_{ij}]^2 \\ &= 2 \sum_{j=1}^{k+1} [-X_{ij} (U_{k+1})_{aj} + (V_{k+1})_{aj} (U_{k+1} V_{k+1})_{ij}]. \end{aligned} \quad (18)$$

η is the step size and is calculated by

$$\eta_{ia} = \frac{(U_{k+1})_{ia}}{2 \sum_{j=1}^{k+1} (U_{k+1} V_{k+1})_{ij} V_{ja}^T}. \quad (19)$$

After substituting (18) and (19) into (17), The update rule equation for $(U_{k+1})_{ia}$ yields as

$$(U'_{k+1})_{ia} \leftarrow (U_{k+1})_{ia} \cdot \frac{(X_k V_k^T + X_{k+1} \mathbf{v}^T)_{ia}}{(U_{k+1} U_k V_k^T + U_{k+1} \mathbf{v} \mathbf{v}^T)_{ia}}. \quad (20)$$

We omit the proof of its convergence as it is similar to that of GNMF while IGNNMF just assumes the first k columns of V would not be updated during iterations.

3.2. Batch-IGNMF. In practice, images often come in batches. For example, a face recognition system needs to record a batch of images belonging to a new class or belonging to many different classes.

Once several images come, our IGNNMF needs to run one time for each image, which is time consuming if there are too many images. In this section, we propose an improved version of IGNNMF algorithm which can deal with a batch of images. This improved version is named Batch-IGNMF (B-IGNMF), which is able to perform incremental study in batch of images no matter whether the batch of images belongs to the same class or different classes.

Let X_l denote the new coming l images. So X_{k+l} is the total sample with its first k columns equal to X_k and the rest l columns equal to X_l . U_{k+l} and V_{k+l} denote the optimized factor matrices of X_{k+l} , assuming V_k as the first k columns of V_{k+l} which would not change during incremental study, V_l is the last l columns of V_{k+l} denoting the corresponding coefficient matrix of X_l , and $(W_{k+l})_{k+1 \sim k+l,:}$ denotes the rows of W_{k+l} from $k+1$ to $k+l$.

So the objective function F_{k+l} can be rewritten as

$$F_{k+l} = \sum_{i=1}^n \sum_{j=1}^{k+l} [X_{ij} - (U_{k+l} V_{k+l})_{ij}]^2 + \lambda \text{Tr}(V_{k+l} L_{k+l} V_{k+l}^T). \quad (21)$$

We omit the detailed derivation process due to its similarity to the derivation process of IGNNMF. We just list the multiplicative update rules as

$$\begin{aligned} (V'_l)_{ia} &\leftarrow (V_l)_{ia} \cdot \frac{[U_{k+1}^T X_l + \lambda \cdot V_{k+l} (W_{k+l})_{k+1 \sim k+l,:}]_{ia}}{[U_{k+l}^T U_{k+l} V_l + \lambda \cdot V_{k+l} (D_{k+l})_{k+1 \sim k+l,:}]_{ia}}, \\ (U'_{k+l})_{ia} &\leftarrow (U_{k+l})_{ia} \cdot \frac{(X_k V_k^T + X_l V_l^T)_{ia}}{(U_{k+l} U_k V_k^T + U_{k+l} V_l V_l^T)_{ia}}. \end{aligned} \quad (22)$$

4. Experiments

In this section, the FERET database [23, 24] and CMU-PIE database [25] are selected to evaluate the performance of our IGNNMF and B-IGNMF algorithms, along with two canonical face recognition algorithms: supervised GNMF (GNMF-S) and unsupervised GNMF (GNMF-U) and three incremental algorithms: INMF, CINMF, and IOPNNMF. NMF is selected as the baseline. The stopping condition of iterative update is defined as (23) in all experiments:

$$\frac{F^{(n-1)} - F^{(n)}}{F^{(n)}} < \delta, \quad (23)$$

where $F^{(n)}$ is the n th iteration of the update criterion function defined in (9) under the n th iteration.

In the FERET database [23, 24], we select 200 individual faces (7 images for each). All images in the database are aligned by the centers of eyes and mouth and then resized into 25×25 . The CMU-PIE database includes total of 68 individuals [25] and we select all these 68 individuals, each of which



FIGURE 1: All images of one person from FERET database.



FIGURE 2: All images of one person from CMU-PIE database.

has 56 images, including 7 poses with neural expression in different views and 38 different lighting conditions and 11 talking images in frontal view. All images in the database are aligned by the centers of eyes and mouth and then resized into 28×23 . Examples of these two datasets are shown in Figures 1 and 2. Each experiment is run for 10 trials, and the parameters for our IGNMF and B-IGNMF are set as follows: $\lambda = 0.01$ and $\delta = 0.01$ for both databases and $r = 200$ for the FERET database and $r = 68$ for the CMU-PIE database. Then the average accuracies and the mean running time are recorded.

4.1. Incremental Study for Single Image. In these experiments, we choose 5 images for each individual as the training set from the FERET database and 42 images for each individual from the CMU-PIE database. Our experiments are performed as follows: first, GNMF and NMF are chosen for initialization; second, IGNMF and INMF are performed to incremental study; GNMF and NMF are also performed by rerunning GNMF and NMF every time when new image comes. We choose one image of each individual from the rest of the database to incremental study, so all algorithms were run at a total of 200 times for the FERET database and 68 times for the CMU-PIE database. Notice INMF is an unsupervised learning algorithm, so unsupervised IGNMF is performed.

Figures 3 and 4 illustrate the curves of recognition rates for INMF, IGNMF, NMF, and GNMF during incremental study. We can see that when a small number of new images come, incremental methods (INMF and IGNMF) perform better than canonical methods (NMF and GNMF). We also observe that incremental methods can keep the recognition rate increasing smoothly with the number of images increasing; while canonical methods would make the recognition rate fluctuating. This is because every time when we rerun the canonical methods, the initial values for U and V are randomly assigned; different initial values would lead to different local minimum value, which in turn affect the recognition rate significantly. It is also noted that the recognition rates of IGNMF and GNMF are better than those of NMF and INMF; this is because the use of the manifold method, which preserves the geometrical information of

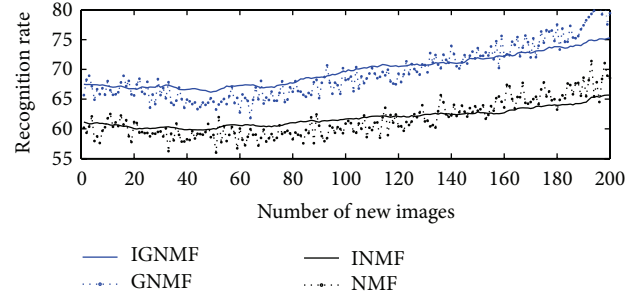


FIGURE 3: The average recognition rate during incremental study for the FERET database.

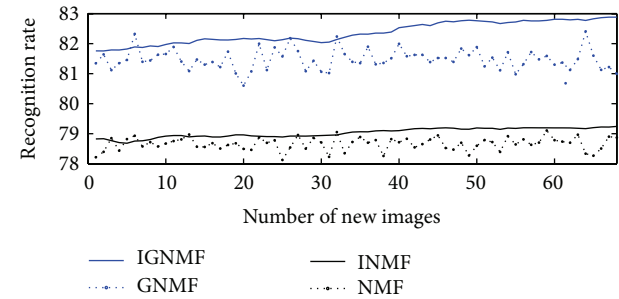


FIGURE 4: The average recognition rate during incremental study for the CMU-PIE database.

the original data, contributes to the improvement of the recognition rate.

Figures 5 and 6 illustrate the running time of the selected algorithms in both databases, from the figure we can see that the mean running time of IGNMF for single image is close then INMF, while faster than GNMF and NMF, which means IGNMF achieved better recognition rate than INMF while faster than it.

4.2. Incremental Study for One Batch of Images Belonging to One Class. In this section, our B-IGNMF, and the other two typical incremental algorithms (CINMF and IOPNMF) are performed along with supervised GNMF, unsupervised GNMF, and NMF. Because CINMF is a supervised learning algorithm while IOPNMF is an unsupervised one, supervised B-IGNMF (BIGNMF-S) and unsupervised B-IGNMF (BIGNMF-U) are both performed.

We choose 5 images for each individual as the training set from the FERET database and 42 images for each individual from the CMU-PIE database. First, we choose 195 individuals from the FERET database and 63 individuals from the CMU-PIE database for study. Second, one of the rest of five individuals is chosen to add for incremental study, total 5 times. The parameters of CINMF are set to $\alpha = 10$, $\beta = 10^{-7}$, and $r_0 = 6$ for the FERET database and $\alpha = 0.1$, $\beta = 10^{-4}$, and $r_0 = 6$ for the CMU-PIE database; the parameters of IOPNMF are set to $r = 200$ for the FERET database and $r = 136$ for the CMU-PIE database. Tables 1 and 2 showed the recognition rates for two databases; “Add 1” means the recognition rate of adding one batch of images into “Start,”

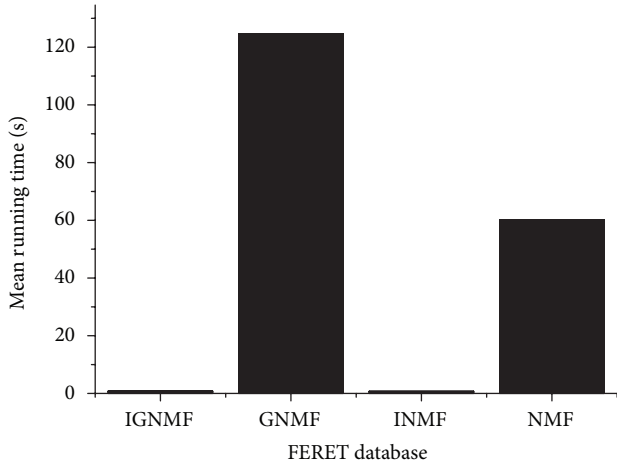


FIGURE 5: The mean running time during incremental study with single image for the FERET database.

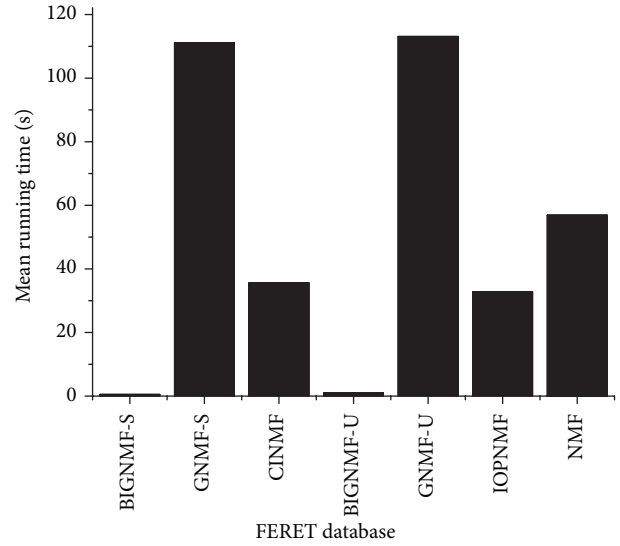


FIGURE 7: The mean running time during incremental study with images belonging to one class for the FERET database.

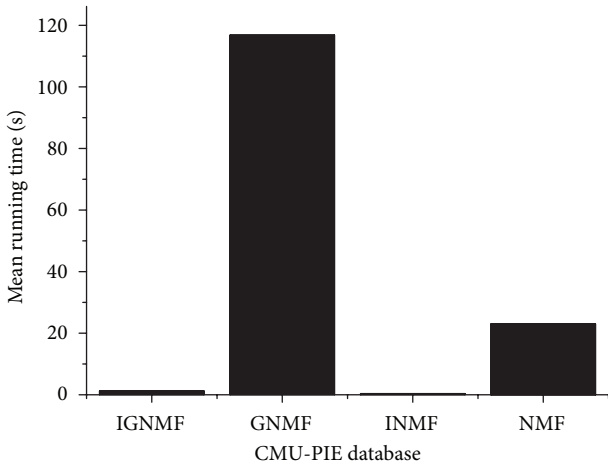


FIGURE 6: The mean running time during incremental study with single image for the CMU-PIE database.

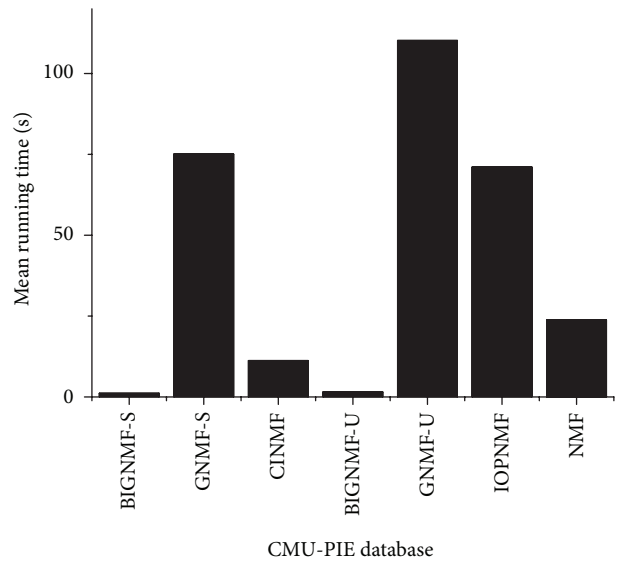


FIGURE 8: The mean running time during incremental study with images belonging to one class for the CMU-PIE database.

and “Add 2” means the recognition rate of adding another one batch of images into “Add 1,” and so on.

Tables 1 and 2 showed the recognition rates for two databases. We can see that the recognition rate of our BIGNMF is close to GNMF, both for the supervised one and the unsupervised one. In some cases BIGNMF is slightly better than GNMF. The recognition rates of GNMF-U, GNMF-S, and NMF are fluctuating during incremental study. In the experiments we found that CINMF was the fastest one to start (study without incremental); it needs less than 40 s for the FERET database and less than 15 s for CMU-PIE database, but during the incremental process, the recognition rate of our BIGNMF outperforms CINMF and is much more faster than CINMF. It is noted that the recognition rate of IOPNMF performs close to NMF for the FERET database while the worst is for the CMU-PIE database; the reason does not concern us.

Figures 7 and 8 illustrate the running time of the selected algorithms in both databases; from the figure we can see

that the mean running time of BIGNMFs for one batch of images belonging to one class is faster than other NMF-based incremental algorithms, both for the supervised one and the unsupervised one, which means our proposed algorithms achieved better recognition rate than other NMF-based incremental algorithms, while faster than then.

4.3. Incremental Study for one Batch of Images Belonging to Different Classes. In this section, incremental study for one batch of images belonging to different classes is performed. Our B-IGNMF, both supervised method and unsupervised method, is selected, along with supervised GNMF, unsupervised GNMF, IOPNMF, and NMF.

TABLE 1: The recognition rates (%) for the FERET database during adding 5 batches of images belonging to one class.

	Supervised methods			Unsupervised methods			
	BIGNMF-S	GNNMF-S	CINMF	BIGNMF-U	GNNMF-U	IOPNMF	NMF
Start	70.77	70.77	67.18	68.72	68.72	61.67	62.41
Add 1	70.28	69.47	66.84	68.37	65.5	61.52	61.12
Add 2	69.86	69.04	67.89	67.89	66.24	61.42	60.58
Add 3	69.44	70.54	67.3	67.3	67	61.4	61.89
Add 4	69.22	69.1	67.09	67	69	61.58	61.98
Add 5	70.28	69.47	66.84	68.37	65.5	61.52	61.12

TABLE 2: The recognition rates (%) for the CMU-PIE database during adding 5 batches of images belonging to one class.

	Supervised methods			Unsupervised methods			
	BIGNMF-S	GNNMF-S	CINMF	BIGNMF-U	GNNMF-U	IOPNMF	NMF
Start	85.05	85.05	82.36	83.64	83.64	50.34	79.20
Add 1	84.92	83.48	82.8	83.47	82.25	50.33	78.16
Add 2	84.79	83.77	82.66	83.3	83.02	50.66	78.54
Add 3	84.6	84.24	82.7	83.17	82.36	50.54	78.8
Add 4	84.62	84.5	82.8	83.13	82.46	50.75	78.88
Add 5	84.1	83.58	82.5	82.17	82.62	50.63	78.98

Experiments are designed as follows: we choose 4 images for each individual (totally 7 images) as the starting training set from the FERET database; the rest of the images are considered as the testing set. For each incremental study, one image for each individual from the testing set is selected as one batch of images to incremental study, totally 200 images for each batch. The incremental study for FERET is performed twice. For the CMU-PIE database, we choose 42 images for each individual (totally 56 images) as the starting training set, then the rest of the images are considered as the testing set. For each incremental study, 2 images for each individual from the testing set are selected as one batch of images to incremental study, total 68×2 images for each batch. The incremental study for FERET is performed 5 times. Tables 3 and 4 showed the recognition rates for two databases; “Add 1” means the recognition rate of adding one batch of images into “Start,” and “Add 2” means the recognition rate of adding another one batch of images into “Add 1,” and so on.

Tables 3 and 4 showed the recognition rates for two databases; we can see that the recognition rate of our BIGNMF, both the supervised one and the unsupervised one, is close to that of GNNMF and better than that of NMF. Also, we can see that the recognition rate for both BIGNMFs is increasing while the new batch of images comes, which illustrate the effectiveness of the new incremental study algorithms. Note in the experiment of the FERET database, in the starting, there are 4 images for each individual, totally 800 images; after adding the second batch of images, we added totally 400 images (2 images for each individual). Which means during the 50% amount of new images to incremental study, our BIGNMF still works. It is noted that the recognition rate of IOPNMF performs close to NMF

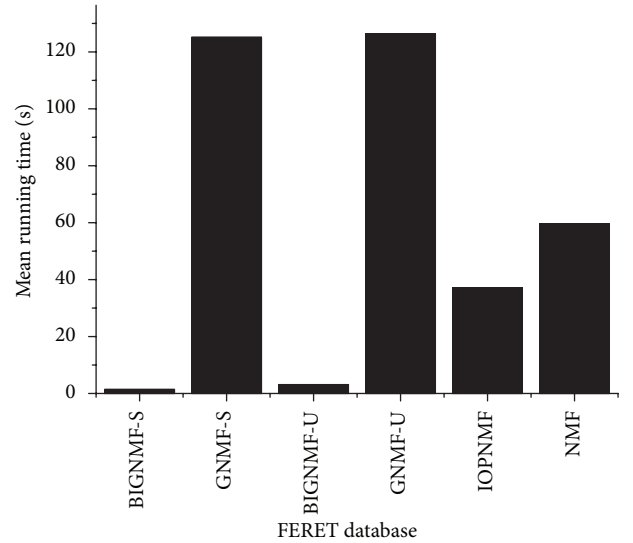


FIGURE 9: The mean running time during incremental study with images belonging to different classes for the FERET database.

for the FERET database while the worst for the CMU-PIE database; the reason is unknown.

Figures 9 and 10 illustrate the mean running times for all the selected algorithms in both database. From the figures we can see that the mean running time of our BIGNMFs for one batch of images belonging to different classes is faster than other NMF-based incremental algorithms, both for the supervised one and the unsupervised one, which means the recognition rate of our proposed algorithms is close to

TABLE 3: The recognition rates (%) for the FERET database during adding 2 batches of images belonging to different classes.

	Supervised methods			Unsupervised methods		
	BIGNMF-S	GNNMF-S	BIGNMF-U	GNNMF-U	IOPNMF	NMF
Start	58.53	58.53	57.24	57.24	49.54	55.06
Add 1	63.41	60.36	60.46	61.93	59.63	59.28
Add 2	69.14	70.94	69.08	71.43	67	67.75

TABLE 4: The recognition rates (%) for the CMU-PIE database during adding 5 batches of images belonging to different classes.

	Supervised methods			Unsupervised methods		
	BIGNMF-S	GNNMF-S	BIGNMF-U	GNNMF-U	IOPNMF	NMF
Start	81.37	81.37	79.15	79.64	46.32	79.01
Add 1	81.46	80.78	79.21	78.77	51.23	79.14
Add 2	81.26	81.98	81.29	81.75	54.85	79.85
Add 3	83.28	82.45	81.36	81.03	53.86	79.38
Add 4	83.3	83.45	83.61	83.59	51.96	80.28
Add 5	84.75	84.49	84.73	84.58	51.47	81.97

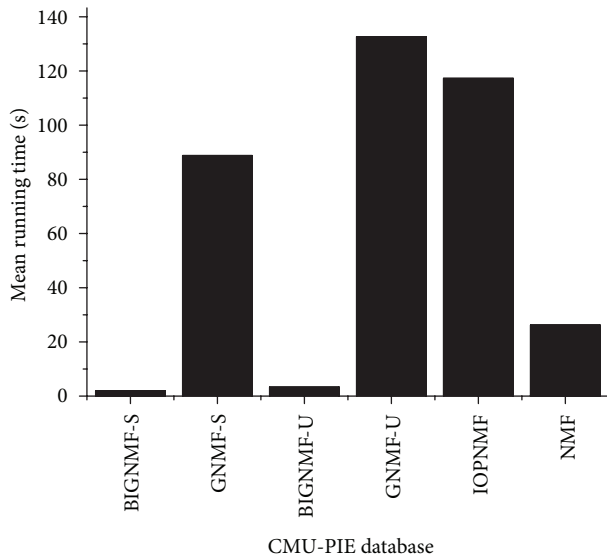


FIGURE 10: The mean running time during incremental study with images belonging to different classes for the CMU-PIE database.

rerunning GNNMF algorithms (sometimes slightly better), but faster.

5. Conclusions

In this paper, a novel incremental study method named incremental graph regularized nonnegative matrix factorization (IGNMF) has been proposed for face recognition. IGNMF introduces the graph regularized nonnegative matrix factorization algorithm into incremental study. The proposed IGNMF is able to preserve discrimination information under incremental study framework. In addition, we adapted our IGNMF algorithm to deal with learning from image batches, resulting in another new learning method called Batch

IGNMF (B-IGNMF). Experiments show that the recognition rates of our IGNMF and B-IGNMF algorithms are close to GNNMF algorithms while they run faster than GNNMF algorithms. The running times of our IGNMF and B-IGNMF algorithms are close to INMF, faster than other popular NMF-based face recognition incremental algorithms, while the recognition rate of our IGNMF and B-IGNMF algorithms outperforms them.

Finally we point out the fact that if IGNMF and BIGNMF run too many times, the recognition rate would worsen than that of rerun GNNMF. The reason for this is the assumption that V_k remains unchanged during the iterations and cannot hold if algorithms run too many times. So our future work will be focusing on how to resolve this issue.

Conflict of Interests

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

Acknowledgments

This work is supported by the Ph.D. Programs Foundation of Ministry of Education of China under Grant no. 20120061110045, the Natural Science Foundation of Jilin province of China under Grant no. 201115022, and Graduate Innovation Fund of Jilin University Project no. 20121103. The corresponding author is Zhe-Zhou Yu.

References

- [1] Y.-X. Wang and Y.-J. Zhang, "Non-negative matrix factorization: a comprehensive review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 6, pp. 1336–1353, 2013.
- [2] D. D. Lee and H. S. Seung, "Algorithms for nonnegative matrix factorization," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS '01)*, pp. 556–562, 2001.

- [3] 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.
- [4] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.
- [5] 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*, pp. I207–I212, December 2001.
- [6] 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*, pp. 27–30, 2004.
- [7] S. Nikitidis, A. Tefas, N. Nikolaidis, and I. Pitas, "Subclass discriminant nonnegative matrix factorization for facial image analysis," *Pattern Recognition*, vol. 45, no. 12, pp. 4080–4091, 2012.
- [8] N. Guan, D. Tao, Z. Luo, and B. Yuan, "NeNMF: an optimal gradient method for nonnegative matrix factorization," *IEEE Transactions on Signal Processing*, vol. 60, no. 6, pp. 2882–2898, 2012.
- [9] N. Guan, D. Tao, Z. Luo, and B. Yuan, "Non-negative patch alignment framework," *IEEE Transactions on Neural Networks*, vol. 22, no. 8, pp. 1218–1230, 2011.
- [10] N. Guan, D. Tao, Z. Luo, and J. Shawe-Taylor, "MahNMF: manhattan non-negative matrix factorization," *Journal of Machine Learning Research*, 2012.
- [11] 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.
- [12] X. Liu, S. Yan, and H. Jin, "Projective nonnegative graph embedding," *IEEE Transactions on Image Processing*, vol. 19, no. 5, pp. 1126–1137, 2010.
- [13] D. Cai, X. He, X. Wu, and J. Han, "Non-negative matrix factorization on manifold," in *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM '08)*, pp. 63–72, December 2008.
- [14] S. S. Bucak and B. Günsel, "Incremental subspace learning via non-negative matrix factorization," *Pattern Recognition*, vol. 42, no. 5, pp. 788–797, 2009.
- [15] W.-S. Chen, B. Pan, B. Fang, M. Li, and J. Tang, "Incremental nonnegative matrix factorization for face recognition," *Mathematical Problems in Engineering*, vol. 2008, Article ID 410674, 17 pages, 2008.
- [16] D. Wang and H. Lu, "Incremental orthogonal projective non-negative matrix factorization and its applications," in *Proceedings of the 18th IEEE International Conference on Image Processing*, Brussels, Belgium, 2011.
- [17] A. Lefèvre, F. Bach, and C. Févotte, "Online algorithms for nonnegative matrix factorization with the Itakura-Saito divergence," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA '11)*, pp. 313–316, October 2011.
- [18] N. Guan, D. Tao, Z. Luo, and B. Yuan, "Online nonnegative matrix factorization with robust stochastic approximation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 7, pp. 1087–1099, 2012.
- [19] B. Cao, D. Shen, J. -T. Sun, X. Wang, Q. Yang, and Z. Chen, "Detect and track latent factors with online nonnegative matrix factorization," in *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pp. 2689–2694, 2007.
- [20] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
- [21] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
- [22] X. He and P. Niyogi, "Locality preserving projections," in *Proceedings of the Conference of Advances in Neural Information Processing Systems*, vol. 16, Vancouver, Canada, 2003.
- [23] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.
- [24] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: a literature survey," *ACM Computing Surveys*, vol. 35, no. 4, pp. 399–458, 2003.
- [25] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression (PIE) database," in *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, May 2002.