Manifold Regularized Discriminative Nonnegative Matrix Factorization With Fast Gradient Descent

Naiyang Guan, Dacheng Tao, Member, IEEE, Zhigang Luo, and Bo Yuan

Abstract—Nonnegative matrix factorization (NMF) has become a popular data-representation method and has been widely used in image processing and pattern-recognition problems. This is because the learned bases can be interpreted as a natural parts-based representation of data and this interpretation is consistent with the psychological intuition of combining parts to form a whole. For practical classification tasks, however, NMF ignores both the local geometry of data and the discriminative information of different classes. In addition, existing research results show that the learned basis is unnecessarily parts-based because there is neither explicit nor implicit constraint to ensure the representation parts-based. In this paper, we introduce the manifold regularization and the margin maximization to NMF and obtain the manifold regularized discriminative NMF (MD-NMF) to overcome the aforementioned problems. The multiplicative update rule (MUR) can be applied to optimizing MD-NMF, but it converges slowly. In this paper, we propose a fast gradient descent (FGD) to optimize MD-NMF. FGD contains a Newton method that searches the optimal step length, and thus, FGD converges much faster than MUR. In addition, FGD in a variant of MD-NMF and experimental results confirm its efficiency. Experimental results on several face image datasets suggest the effectiveness of MD-NMF.

Index Terms—Gradient descent, nonnegative matrix factorization (NMF), manifold regularization.

I. INTRODUCTION

DATA representation plays a fundamental role in many image processing and pattern recognition tasks. A good representation typically reveals the latent structure of a dataset, and thus, can be used to improve the performance and to reduce the redundancy as well as the subsequent computational cost. Data-representation methods are often tailored to the statistics of the data, which can be learned from observations. Some types of data, and in particular all image and video segments, are nonnegative. Data-analysis tools, such as principal component analysis (PCA) [18] and Fisher’s linear discriminant analysis (FLDA) [17] do not, however, maintain this nonnegativity property, i.e., the basis vectors have both positive and negative elements, and the combination coefficients for data representation have negative elements as well. Therefore, the representation obtained by classical data-analysis tools is inconsistent with the psychological intuition of combining parts to form a whole.

Recently, nonnegative matrix factorization (NMF, [24]) has been proposed to incorporate the nonnegativity constraints and obtain a parts-based representation. It has been widely applied to image processing and pattern recognition applications, because the learned bases can be interpreted as a natural parts-based representation of data. In particular, it represents data as a linear combination of a set of basis vectors, in which both the combination coefficients and the basis vectors are nonnegative.

Several NMF’s variants have been developed by introducing additional constraints to the original NMF. By adding the localization constraint, Li et al. [26] presented the local NMF (LNMF) that learns spatially localized, parts-based subspace representation for visual patterns. Hoyer [22] proposed the nonnegative sparse coding (NSC) that incorporates the sparseness constraint with NMF to obtain sparse encoding vectors. To incorporate the data geometric structure, Cai et al. [12] proposed a graph-regularized NMF (GNMF). The data geometric structure is encoded by a nearest-neighbor (NN) graph, which plays an important role in popular dimension reduction algorithms, such as Laplacian eigenmap [5] and locality-preserving projections [21]. GNMF was specifically designed for clustering tasks, so it cannot perform well for classification problems. By introducing the Fisher’s discriminative information to NMF, Zafeiriou et al. [47] developed the discriminant NMF (DNMF) for frontal face verification. Yang et al. [44] proposed the nonnegative graph embedding (NGE) that simultaneously learns two subspaces to incorporate the marginal Fisher discriminative information with
NMF. However, NGE requires the matrix inverse operator in each iteration round. To improve the efficiency, Wang et al. [42] presented a multiplicative algorithm for optimizing NGE. Liu et al. [30] further developed PNGE that learns nonnegative projection matrix for dimension reduction.

NMF and its extensions have been successfully applied to various tasks in biometrics [26], image processing [46], computer vision [24], and document analysis [35]. However, they share some of the following problems for practical applications.

1) Donoho and Stodden [16] theoretically proved that NMF does not necessarily decompose an object into parts, i.e., NMF could fail to obtain the parts-based representation for some datasets. For example, Fig. 2 shows that NMF performs well on the CBCL face dataset but fails to get the parts-based representation on the ORL face dataset [36]. Many NMF’s variants share this problem.

2) In many practical applications, samples lie on a low-dimensional manifold embedded in a high-dimensional ambient space, and thus, it is necessary to consider the data geometric structure to obtain the parts-based representation. However, NMF and its popular extensions do not consider it. Although GNMF uses the NN graph to encode the data geometric structure, it is specifically designed for clustering tasks and cannot perform well on classification tasks.

3) NMF as well as most of its variants are unsupervised, i.e., they ignore the discriminative information of different classes. Although DNMF applies Fisher’s discriminative information for extracting discriminative basis for data representation, it would be more effective to consider the margin-based discriminative information for subsequent data representation. Though NGE outperforms NMF, it is difficult to determine the dimensionality of the two subspaces, and thus, it is not easy to apply it in practice.

4) NMF and its variants use the multiplicative update rule (MUR) for optimization. Although MUR is simple and easy to implement, it converges slowly. In recent years, many algorithms [28], [32] have been developed to accelerate MUR, but they were specifically designed for the least square problem with the nonnegativity constraint. Therefore, it is not easy to apply them to NMF’s variants and NMF of the form that minimizes the Kullback–Leibler (KL) divergence between original samples and their approximations.

In this paper, we propose a manifold regularized discriminative NMF (MD-NMF) to overcome the aforementioned problems. Recently, manifold regularization [3] has been introduced to encode the data geometric structure in learning classification/regression functions [9], [11] and dimension reduction [51]. We apply it to preserving the data geometric structure in learning an NMF subspace. In addition, discriminative dimension reduction has been popularly used in pattern classification [39], [40], [10]. To incorporate the discriminative information in the learned subspace, we maximize the margins between different classes, and thus, improve the performance of NMF in classification tasks. It is related to marginal Fisher analysis (MFA) [45], [43] and discriminative locality alignment (DLA) [49], [50]. We further constrain the learned bases to be orthogonal to make the learned basis parts-based. This is inspired by the intuition that different facial parts appear in different positions on a face, and this indicates the orthogonality of bases. Although MUR can obtain a local solution of MD-NMF and its convergence can be proved by introducing a new auxiliary function, it converges slowly. We thus propose a fast gradient descent (FGD) to optimize MD-NMF. FGD uses the Newton method to search the optimal step length, and thus, it converges much faster than MUR. In addition, FGD includes MUR as a special case and can be applied to optimizing NMF and its variants. For example, FGD converges within 28 s to optimize MD-NMF for a problem with 165 samples in $\mathbb{R}^{1000}$, while MUR costs 282 s. By replacing the KL divergence with Frobenius norm for measuring the distance between the original matrix and its approximation, we present a variant of MD-NMF named Frobenius norm MD-NMF (FMD-NMF). In addition, we apply MUR, FGD, and the projected gradient descent (PGD) to FMD-NMF, experimental results on several problems confirm the efficiency of FGD. Finally, we apply MD-NMF to face recognition on several face image datasets and experimental results suggest the effectiveness of MD-NMF comparing with NMF and the representative variants.

The rest of this paper is outlined as follows. Section II briefly reviews NMF and its variants. Section III presents the proposed MD-NMF, the corresponding MUR, and FGD. Section IV presents a variant of MD-NMF. Section V shows the experimental results on popular face image datasets. We conclude this
paper in Section VI. Detailed proofs of lemmas and theorems are given in Appendices A–C.

II. RELATED WORK

This section briefly reviews NMF and its representation variants. Although these algorithms have been successfully applied to various applications, they share several problems.

A. Nonnegative Matrix Factorization

Given $n$ samples in $\mathbb{R}^m$, whose elements are all nonnegative, arranged in columns of a matrix $V \in \mathbb{R}_{+}^{m \times n}$. NMF finds two low-rank nonnegative matrices $W \in \mathbb{R}_{+}^{m \times r}$ and $H \in \mathbb{R}_{+}^{n \times r}$, with $r < \min\{m, n\}$, to approximate the original matrix, i.e., $V \approx WH$.

The columns of $W$ signify bases and the columns of $H$ are encoding vectors, i.e., each column of $H$ is the projection of a sample in the low-dimensional subspace. NMF algorithm minimizes the KL divergence between $V$ and $WH$, subject to $W, H \geq 0$.

$$\min_{W,H} \sum_{i,j} \left( V_{ij} \log \frac{V_{ij}}{Z_{ij}} - V_{ij} + Z_{ij} \right), \quad \text{s.t. } W, H \geq 0$$

where $Z = WH$, log refers to the natural logarithmic function, and the KL divergence vanishes iff $V = WH$. The objective function of NMF is nonconvex with respect to both $W$ and $H$, so it is impractical to find the global optimum. Fortunately, it is convex with respect to either $W$ or $H$, so iterative optimization algorithms can be used to find a local optimum. To obtain $W$ and $H$, Lee and Seung [24] developed the MUR

$$W_{ij} \leftarrow W_{ij} \sum_k \frac{V_{ik}}{(WH)_{ik}} H_{jk} \quad \text{(1)}$$

$$W_{ij} \leftarrow \frac{W_{ij}}{\sum_k W_{kj}} \quad \text{(2)}$$

$$H_{ij} \leftarrow H_{ij} \sum_k W_{ki} \frac{V_{kj}}{(WH)_{kj}} \quad \text{(3)}$$

where (1) and (3) are used to update $W$ and $H$, respectively, and (2) is used to normalize the basis matrix $W$.

B. Variants of NMF

With the nonnegativity constraint, usually NMF can get intuitively parts-based representation for a given dataset, but this is unnecessary (see Fig. 2). Li et al. [26] proposed LNMF by adding penalties to NMF to robustly obtain a parts-based representation.

$$D_{\text{LNMF}} = D(V, WH) + \alpha \sum_{ij} U_{ij} - \beta \sum_i U_{ii}'$$

where $D(V, WH)$ is the KL divergence between $V$ and $WH$, $U = WH^TW$, and $U' = HHT$. According to Li et al. [26], minimizing $\sum_{ij} U_{ij}$ suppresses over decomposition of the bases matrix $W$, while maximizing $\sum_i U_{ii}'$ encourages retaining components with important information.

Hoyer [22] proposed NSC to ensure sparsity of the encoding matrix

$$D_{\text{NSC}} = \frac{1}{2} \| V - WH \|_F^2 + \lambda \sum_{ij} H_{ij}$$

Both LNMF and NSC can learn a more robust parts-based representation than NMF, but they ignore the discriminative information and the data geometric structure.

To use the discriminative information, Zafeiriou et al. [47] proposed DNMF that combines Fisher’s criterion with NMF

$$D_{\text{DNMF}} = D(V, WH) + \gamma S_w - \delta S_b$$

where $S_w$ and $S_b$ are within-class scatter and between-class scatter of $H$, respectively. Since NMF itself does not assume samples are Gaussian distributed, it is improper to use Fisher’s criterion to retain the discriminative information for subsequent classification.

Recently, Cai et al. [12] proposed GNMF to encode the data geometric structure in an NN graph

$$D_{\text{GNMF}} = \frac{1}{2} \| V - WH \|_F^2 + \lambda t(x(HLHT)^T)$$

where $\| \cdot \|_F$ is the Frobenius norm and $L$ is the graph Laplacian matrix. GNMF was specifically designed for clustering, and it does not perform well for classification problems.

Yang et al. [44] replaced the graph regularization criterion in GNMF with the marginal Fisher discriminant criterion and proposed nonnegative graph embedding (NGE). NGE simultaneously learns two subspaces, i.e., $W = [W^1, W^2]$. The objective function of NGE is

$$D_{\text{NGE}} = \text{tr}(H^3LH^T) + \text{tr}(H^2LP^T) + \lambda \| V - WH \|_F^2$$

where $L$ and $P$ are the Laplacian matrices of intrinsic and penalty graphs, respectively, and $H^3$ and $H^2$ are the coefficients of samples in the subspaces $W^1$ and $W^2$, respectively, i.e., $H = [H^1, H^2]^T$.

Liu et al. [30] developed projective NGE (PNGE) by directly learning the nonnegative projection matrix for dimension reduction

$$D_{\text{PNGE}} = \| V - WPV\|_F^2 + \lambda (\text{tr}(\hat{P}VLVT\hat{P}^T) + \text{tr}(\hat{P}VL^VP^T\hat{P}^T))$$

where $\hat{P}$ and $\hat{P}$ are the projection matrices for subspaces $W^1$ and $W^2$, i.e., $P = [\hat{P}^T, \hat{P}^T]^T$. However, it is difficult to determine the dimensionality of the two subspaces in practice, respectively.

III. MANIFOLD REGULARIZED DISCRIMINATIVE NMF

Here, we introduce the proposed MD-NMF, which encodes the data geometric structure and the discriminative information of different classes in the parts-based representation. It is direct to propose an MUR to optimize MD-NMF, but this algorithm converges slowly. We thus further propose an FGD to obtain a local solution of MD-NMF. FGD contains a Newton method to search the optimal step length. In addition, FGD includes MUR as a special case and can be applied to optimizing NMF and its variants.
A. Objective Function

From the perspective of dimension reduction, NMF learns to represent original samples \( V = \{v_1, \ldots, v_m\} \) denoted by \( m \)-dimensional nonnegative vectors \( v_i \in \mathbb{R}_+^m \), as a linear combination of low-dimensional bases represented by a nonnegative matrix \( W \in \mathbb{R}_+^{m \times r} \) in a manner of addition without subtraction, i.e., \( V = WH \), wherein \( H \in \mathbb{R}_+^{r \times m} \) is the coefficient matrix. Thus, it can be deemed as a function \( f(v) = h \), subject to \( v = WH \), to preserve the local geometry of the distribution of samples \( V \). Suppose \( V \) is sampled from a probability distribution \( P_V \) on a manifold \( \mathcal{M} \) embedded in a high-dimensional ambient space, and this can be achieved by penalizing the gradient \( \nabla_M f \) along the manifold \( \mathcal{M} \)

\[
\int_{V \in \mathcal{M}} \left\| \nabla_M f \right\|^2 dP_V
\]  

(4)

where the integral is taken over the probability distribution \( P_V \). However, both the manifold \( \mathcal{M} \) and marginal distribution \( P_V \) are unknown in practice, so we use an empirical estimate of the penalty (4).

Fortunately, it has been well studied in manifold learning that the regularization can be approximated by using the graph Laplacian of samples \( V \). The main idea is to construct an adjacent graph whose vertexes correspond to samples, and the pairwise edge weights \( S_{ij} \) reflect the extent to which two samples are close. Traditionally, the edge weight is defined by the heat kernel \( S_{ij} = e^{-(\|v_i - v_j\|^2 / \delta^2)} \), with a predefined \( \delta \). By ignoring \( \delta \), the edge weight matrix reduces to a 0–1 matrix with entries defined by

\[
S_{ij} = \begin{cases} 1, & \text{if } v_i \in N_k(v_j) \text{ or } v_j \in N_k(v_i) \\ 0, & \text{otherwise} \end{cases}
\]

(5)

where \( N_k(v_i) \) consists of \( k \) NNs of \( v_i \). In the rest of the paper, we use this strategy to construct the edge weight matrices. By using (5), the regularization (4) can be approximated by \( f^T L f \), wherein \( L = D - S \) is graph Laplacian matrix [13], and the \( i \)-th element of the diagonal matrix \( D = \sum_j S_{ij} \).

Similar to MFA [45], [43] and DLA [49], [50], MD-NMF encodes both the data geometric structure and the discriminative information. For each sample \( v_i \), according to (5), the first graph is constructed by selecting \( k_1 \) NNs from samples that have the same label as \( v_i \). Since there are \( r \) independent components in the bases \( W = \{w_1, \ldots, w_r\} \), we take the summation of \( r \) regularizations

\[
\sum_{i=1}^{r} f_i^T L_1 f_i = \text{tr}(F_1 L_1 F_1^T)
\]

(6)

where \( L_1 \) is the Laplacian matrix of the first adjacent graph and \( F = \{f_1, \ldots, f_r\}^T \).

To encode the discriminative information, we maximize margins between different classes. According to (5), the second adjacent graph for \( v_i \) is constructed by selecting \( k_2 \) NNs from samples that have labels different from that of \( v_i \). Another penalty is built by summing over the \( r \) independent regularizations

\[
\sum_{i=1}^{r} f_i^T L_2 f_i = \text{tr}(F_2 L_2 F_2^T)
\]

(7)

where \( L_2 \) is the Laplacian matrix of the second adjacent graph.

We minimize (6) to retain the data geometric structure, while we maximize (7) to make the samples in different classes separable. Thus, by combining (6) and (7), we have

\[
\text{tr} \left( F \left( \frac{1}{2} L_1^{-1/2} L_2 L_1^{-1/2} F^T \right)^T \right).
\]

(8)

Since \( L_2 \) is not necessarily positive definite, it could be non-invertible. From [4], we add a tiny perturbation to the diagonal of the graph Laplacian matrix, i.e., \( \tilde{L}_2 = L_2 + \zeta I \), to ensure that it is always invertible. It has been shown that the solution obtained by the regularized graph Laplacian matrix is consistent with the original one as long as \( \zeta \) is fixed to a small values. In all experiments, we empirically set \( \zeta \) as \( 10^{-4} \text{tr}(L_2) \). In the rest of the paper, \( \tilde{L}_2 \) implies the perturbed matrix \( L_2 \).

By incorporating (8) with NMF, and letting \( F = H \), we obtain

\[
\min_{W,H} KL(V,WH) + \frac{\gamma}{2} \text{tr} \left( H \left( \frac{1}{2} L_1^{-1/2} L_2 L_1^{-1/2} F^T \right)^T \right)
\]

\[
\text{s.t.} \quad W \geq 0, H \geq 0
\]

(9)

where \( \gamma \) is a tradeoff parameter and \( KL(\cdot,\cdot) \) is the KL divergence. Although other metrics, e.g., Frobenius norm (see Section IV), can be used to measure the distance between \( V \) and \( WH \), the experimental results show that the KL divergence-based algorithm performs pretty well.

In order to ensure the learned bases to be parts-based, we penalize (9) by making bases orthogonal [15]

\[
\sum_{i \neq j} w_i^T w_j = \text{tr}(W^T W), \quad e = I - I
\]

(10)

where \( I \) signifies the matrix whose elements are all one. The intuition behind is that different parts of a face, e.g., nose and eyes, appear at different positions on the face.

In order to penalize nonsmoothness of \( H \), according to [35], the Tikhonov regularization is imposed over the coefficient matrix \( H \)

\[
\sum_{i,j} h_{ij}^2 = \text{tr}(HH^T)
\]

(11)

By combining (9)–(11), we arrive at the objective function of the proposed MD-NMF

\[
F(W, H) = KL(V,WH) + \frac{\alpha}{2} \text{tr}(W^T W) + \frac{\beta}{2} \text{tr}(HH^T)
\]

\[
+ \frac{\gamma}{2} \text{tr} \left( H \left( \frac{1}{2} L_1^{-1/2} L_2 L_1^{-1/2} F^T \right)^T \right)
\]

(12)

where \( \alpha \) and \( \beta \) are the trade-off parameters of the orthogonality and the Tikhonov regularizations. We use 1/2 over \( \alpha, \beta, \) and \( \gamma \) to simplify deductions. The model is stable over wide ranges of \( \alpha \) and \( \beta \) with small values. Other regularizations, e.g., the transfer subspace learning regularization [37], can be incorporated to (12), which we will discuss in future works.

B. Multiplicative Update Rule

Although (12) is nonconvex with respect to both \( W \) and \( H \), it is convex with respect to either \( W \) or \( H \). Therefore, we present
an iterative algorithm to update $W$ and $H$ alternately based on MUR

$$W \leftarrow W \odot \frac{V}{\text{HT}} \left( \frac{W}{EH} + \alpha We \right) \quad (13)$$

$$H \leftarrow H \odot \frac{W^T V}{\text{HT}} \left( \gamma HS \right) \quad (14)$$

where $E \in \mathbb{R}^{m \times n}$ is a rectangle matrix whose elements are all one, $e$ is defined as that in (10), $\odot$ denotes the element-wise multiplication, the division used in above two formulas are element-wise as well, $D = (L_c^{-1/2})^T D_g L_c^{-1/2}$, and $S = (L_c^{-1/2})^T S_g L_c^{-1/2}$. The detailed derivations are given in Appendix A. Theorem 1 shows that $D$ and $S$ are nonnegative matrices.

**Theorem 1:** Both $D$ and $S$ are nonnegative matrices.

We leave the detailed proof in the Appendix B.

We can iteratively update $W$ and $H$ until the objective value of (12) does not change. The procedure is summarized in Algorithm 1.

**Algorithm 1 MUR optimization for MD-NMF**

**Input:** $V \in \mathbb{R}^{m \times n}$, $D, S, 1 \leq r \leq \min\{m, n\}$

**Output:** $W \in \mathbb{R}^{m \times r}$, $H \in \mathbb{R}^{r \times n}$

1. Initialize $W_0$, $H_0$, $e$, $e_k$, $k = 0$.

**repeat**

2.1: Update $H_{k+1}$ as

$$H_{k+1} = H_k \odot \frac{W_k^T V}{W_k E + \beta H_k + \gamma H_k \bar{D}}.$$  

2.2: Update $W_{k+1}$ as

$$W_{k+1} = W_k \odot \frac{V}{W_k E_{k+1} \left( \frac{W_k}{E_{k+1}} \right) + \alpha W_k e}.$$  

2.3: $k = k + 1$.

**until** Stopping criteria is met.

Since $V$ can be approximated by $WH$ column-wise, i.e., $v_i \approx Wh_i$, we can naturally project a sample $x$ from the original high-dimensional space to the low-dimensional space, or equivalently, $y = W^Tx$, wherein the projection matrix $W^T = (W^T W)^{-1} W^T$ is the pseudoinverse of $W$.

Section III-C shows that MUR can decrease the objective function of MD-NMF (12). The time cost of one iteration round of MUR is $O(tmn)$. Since it converges slowly, the overall time cost is large. In Section III-D, we present FGD to optimize MD-NMF. FGD contains a Newton method to search the optimal step length, and thus, FGD converges much faster than MUR. In addition, FGD includes MUR as a special case and can be applied to optimizing NMF and its variants.

**C. Convergence Analysis**

By introducing an auxiliary function, we can prove the convergence of MUR for MD-NMF.

**Definition 1:** The function $G(x, x^t)$ is an auxiliary function for $F(x)$, if $G(x, x^t) \geq F(x)$ and $G(x, x) = F(x)$.

**Lemma 1:** If $G(x, x^t)$ is an auxiliary function of $F(x)$, then $F(x)$ is nonincreasing under the update $x^{t+1} = \arg \min_x G(x, x^t)$.

**Proof:** $F(x^{t+1}) \leq G(x^{t+1}, x^t) \leq G(x^t, x^t) = F(x^t)$.  

**Lemma 2:** The objective function (12) is nonincreasing by using the update (13).

According to Lemma 1, Lemma 2 shows that the objective function (12) does not increase by using the update (13) with $H$ fixed.

In order to prove that the objective function (12) does not increase by using the update (14) with $W$ fixed, we first introduce an inequality in Lemma 3.

**Lemma 3:** Given an arbitrary positive number $\lambda > 0$, for any two positive variables $a > 0$ and $b > 0$, the following inequality holds:

$$a - b + \frac{\lambda}{2}(a - b)^2 \geq b \log \frac{a}{b}.$$  

According to Lemma 3, we have the following Lemma 4.

**Lemma 4:** The objective function (12) is nonincreasing by using the update (14).

According to Lemmas 2 and 4, we have Theorem 2 that MUR for MD-NMF converges.

**Theorem 2:** The MUR-based optimization does not increase the objective function of MD-NMF (12).

**Proof:** According to Lemma 2, we have

$$F(W^{t+1}, H^t) \leq F(W^t, H^t).$$  

Similarly, according to Lemma 4, we have

$$F(W^{t+1}, H^{t+1}) \leq F(W^t, H^t).$$  

By combining (15) and (16), we have

$$F(W^{t+1}, H^{t+1}) \leq F(W^t, H^t).$$  

Thus, MUR reduces the objective function (12) in each iteration round.

The strict equality in (17) holds when MUR stops and the gradients of $F(W, H)$ with respect to $W$ and $H$ vanish. Thus, we can conclude that MUR-based optimization for MD-NMF converges to a limit point. In constrained optimization, any local minimum must be a stationary point. However, the MUR for NMF does not guarantee this stationarity of the obtained limit point, Lin [29] modified MUR and made it converge to a stationary point.

**D. Fast Gradient Descent**

Although MUR can optimize MD-NMF, it converges slowly. In this section, we present FGD that uses a Newton method to search the optimal step length to minimize the objective function.
(12) in the direction of scaled negative gradient, at each iteration round.

Without loss of generality, we take the procedure for optimizing $H$, for example, and $W$ can be updated similarly. The overall optimization is alternatively updating $H$ with $W$ fixed and updating $W$ with $H$ fixed. At the iteration round $k$, according to (14), the scaled negative gradient can be written in matrix form as

$$\nabla_H = H_k \odot \frac{W^T V}{W H_k} + \gamma H_k S \frac{W^T E + \beta H_k + \gamma H_k D}{W^T E + \beta H_k + \gamma H_k D} - H_k. \quad (18)$$

To make $H$ nonnegative, the step length should be selected in the range of

$$D_\theta = \{ \theta \mid H_k + \theta \nabla_H \succeq 0, \theta > 0 \}.$$ 

Therefore, MUR can be deemed as a special case of FGD by setting $\theta = 1$. MUR does not converge quickly, because the step length $\theta$ is not optimal. In FGD, we use Newton method to obtain the optimal step length $\theta$ to minimize $F(W, H_k + \theta \nabla H)$. This problem can be written as

$$\min_{\theta} \phi(\theta), \quad \text{s.t. } \theta > 0$$

where $\phi(\theta) = F(W, H_k + \theta \nabla_H)$. According to the objective function (12), we obtain the first-order and second-order derivatives of $\phi(\theta)$, which are

$$\phi'(\theta) = (\beta tr \left( \nabla_H \nabla_H^T \right) + \gamma tr \left( \nabla_H L \nabla_H^T \right)) \theta - \sum_{i,j} \left( V_{ij} / (W H_k)_{ij} + (W \nabla H)_{ij} \right) \theta + \sum_{i,j} \left( (W \nabla H)_{ij} \right)$$

$$\phi''(\theta) = \beta tr \left( \nabla_H \nabla_H^T \right) + \gamma tr \left( \nabla_H L \nabla_H^T \right) + 2 \sum_{i,j} \left( \left( (W H_k)_{ij} + (W \nabla H)_{ij} \theta \right) \theta \right)$$

respectively, where $L = D - S$. Note that the second-order derivative is positive, i.e., $\phi''(\theta) > 0$, which implies that $\phi(\theta)$ is convex, and thus, there exists the global minimum for $\phi(\theta)$. The Newton method for searching the optimal step length $\theta$ in the iteration round $i(i \geq 0)$ is thus given by

$$\theta_{k+1} = \theta_i - \frac{\phi'(\theta_i)}{\phi''(\theta_i)}. \quad (21)$$

The algorithm stops if $\theta_{k+1} - \theta_i = 0$. The initial step length $\theta_0$ of (21) is set as $\theta_0 = 0$, wherein $\theta^{k-1}$ is the optimal step length of the previous iteration round. This setting makes the Newton method (21) converges quickly, and empirically only a few iteration rounds are required. The final step length is chosen to ensure that the new factor value, i.e., $H^{k+1}$, is as close as possible to the minimum along the $\nabla H$ direction without exceeding the positive quadrant. Formally, we have

$$\theta^{k+1} = \min \{ \theta^*, \theta' \} \quad (22)$$

where $\theta' = \tau \sup(D_\theta)$, and $\tau(0 < \tau < 1)$ is used to ensure that $H^{k+1}$ is not too close to the boundary. Note that $\sup(D_\theta) = \max \{ (H_{k+1}) / (\nabla H_{k+1}) \mid \nabla H_{k+1} < 0 \}$, and $\sup(D_\theta) > 1$, because $1 \in D_\theta$ and $H_k + \nabla H > 0$. The inequality $\theta' > 1$ holds, if $\tau$ is selected from $((1) / (\sup(D_\theta))) \cdot 1$. In our experiments, we set $\tau = (1) / (\sup(D_\theta)) \times 0.01 + 0.99$ to make $\tau$ sufficiently close to 1. With the new step length obtained by (22), $H$ is updated by

$$H_{k+1} = H_k + \theta^{k+1} \nabla H \quad (23)$$

and the step length $\theta^{k+1}$ will be retained as the initial step length for the Newton method (21) in next iteration round. The procedure is summarized in Algorithm 2.

**Algorithm 2 FGD procedure for updating $H$**

**Input:** $H_k \in \mathbb{R}^{r \times n}$, $\theta_k$

**Output:** $H_{k+1} \in \mathbb{R}^{r \times n}$, $\theta^{k+1}$

1: Initialize $\theta_0 = \theta^k, i = 0$.
2: Calculate $\nabla H$ as (18).
3: Calculate $\lambda = \max \{ (H_{k+1}) / (\nabla H_{k+1}) \mid \nabla H_{k+1} < 0 \}$.
4: Set $\theta' = 0.01 + 0.99 \times \lambda$.

**repeat**

5.1: Update $\theta_{k+1}$ as,

$$\theta_{k+1} = \theta_i - \frac{\phi'(\theta_i)}{\phi''(\theta_i)}.$$  

5.2: $i = i + 1$.

**until** Stopping criteria is met

6: Set $\theta^{k+1} = \min \{ \theta_i, \theta' \}$.  

7: Update $H_{k+1}$ as $H_{k+1} = H_k + \theta^{k+1} \nabla H$.

Based on a similar procedure, we can get the procedure to update $W$. By alternatively updating $W$ and $H$, we can obtain a local solution for MD-NMF. The convergence of FGD is guaranteed by the following theorem.

**Theorem 3:** The FGD-based optimization does not increase the objective function of MD-NMF (12).

**Proof:** Since MUR converges to a limit point, it is sufficient to show that FGD further reduces the objective function (12), i.e., $\phi(\theta^{k+1}) \leq \phi(\theta)$. We prove this inequality in the following two scenarios.

1) If $\theta^{k+1} = \theta^*$, it is direct to have $\phi(\theta^{k+1}) \leq \phi(\theta)$, because $\theta^* \geq \theta$ is the optimal step length.

2) If $\theta^{k+1} = \theta'$, it implies that $\theta' < \theta^*$. In addition, $\theta' > 1$. Thus, according to above analysis, we have $1 < \theta' < \theta^*$, and there exists $\mu(0 < \mu < 1)$, subject to $\theta' = \mu \theta^* + 1 - \mu$. According to Jensen’s inequality, we have $\phi(\theta') \leq \mu \phi(\theta^*) + (1 - \mu) \phi(1)$ because $\phi(\theta)$ is convex. Since $\phi(\theta^*) \leq \phi(1)$, we have $\phi(\theta^{k+1}) \leq \phi(\theta') < \phi(1)$. By combining scenarios (1) and (2), we have $\phi(\theta^{k+1}) \leq \phi(1)$. This completes the proof.

The time cost of the Newton method (21) in FGD is $\#\text{iteration} \times O(mn)$, wherein $\#\text{iteration}$ is the number of iteration rounds to obtain the optimal step.
length. Thus, the time cost of one iteration round of FGD #iteration \times O(mn) + O(mnr) is comparable to that of one iteration round of MUR O(mnr), if #iteration is small. Our experiments show that #iteration is usually 3–5. However, FGD converges much faster than MUR, so the overall time cost of FGD is much smaller than that of MUR.

Table I compares the objective values, CPU seconds and the corresponding iteration numbers of FGD and MUR for optimizing MD-NMF on YALE [6], ORL [36], and UMIST [19] face datasets. For each test, the initialization of FGD and that of MUR are identical.

For the test on the YALE dataset, wherein the sample matrix is of 1600 \times 165 and the subspace dimension is 50, FGD converges after 180 iteration rounds, while MUR uses 4450 iteration rounds, as shown in Fig. 3(a). Fig. 3(b) shows that FGD converges within 28 s, while MUR costs 282 s for the same problem. Other experimental results on YALE [6], ORL [36], and UMIST [19] reported in Table I also suggest the efficiency of FGD for optimizing MD-NMF. Although it is difficult to theoretically guarantee that FGD converges to a local minimum, the experimental results show that it usually converges to a local minimum.

**IV. VARIANT OF MD-NMF**

In the rest of the paper, we briefly denote this variant as FMD-NMF. To optimize FMD-NMF, we apply MUR, FGD, and PGD.

**A. Algorithms for FMD-NMF**

It has been shown that the Frobenius norm can measure the distance between the original matrix \( V \) and its approximation \( WH \). Therefore, it is reasonable to replace the KL divergence in MD-NMF (12) with the Frobenius norm, and then, we arrive at the definition of Frobenius norm MD-NMF (FMD-NMF)

\[
\tau_{\min_{W \geq 0, H \geq 0}} f(W, H) = \frac{1}{2} \| V - WH \|_F^2 + \frac{\alpha}{2} \text{tr}(W^T W) + \frac{\gamma}{2} \text{tr}(H \hat{L} \hat{H}^T) 
\]

where \( \| \cdot \|_F \) refers to the matrix Frobenius norm, and \( \hat{L} = L + \beta / \gamma I \), the constant 1/2 over each item is used for simplifying related mathematical deductions.

The gradients of \( f(W, H) \) with respect to \( H \) and \( W \) are

\[
\nabla_H f(W, H) = W^T WH - W^T V + \gamma \hat{L} \\
\nabla_W f(W, H) = WHH^T - VH^T + \alpha W \varepsilon.
\]
By using the gradient descent method, and fixing the step length to guarantee the nonnegativity of $W$ and $H$, we can obtain MUR for FMD-NMF as
\[
H \leftarrow H - \frac{W^T V + \gamma HS}{W^T W H + \gamma D} \quad \text{(27)}
\]
\[
W \leftarrow W - \frac{V H^T}{W H H^T + \alpha W} \quad \text{(28)}
\]
where $D = D + (\beta / (\gamma))$. Theorem 4 shows that the objective function $f(W, H)$ is nonincreasing under both update (27) and (28). By alternatively using (27) with $W$ fixed and using (28) with $H$ fixed, we obtain a local solution of (24).

**Theorem 4:** The objective function $f(W, H)$ is nonincreasing under update (27) with $W$ fixed and under (28) with $H$ fixed.

**Proof:** In the iteration round $k$, given $W_k$, we construct an auxiliary function for $f(W_k, H)$ as
\[
G(H, H_k) = f(W_k, H_k) + \langle \nabla_H f(W_k, H_k), H - H_k \rangle + \frac{W_k^T W_k H_k + \gamma \tilde{D}}{H_k} (H - H_k)^2
\]
where $(X)^2$ is the elementwise square of the matrix $X$. Since $G(H, H_k) = f(W_k, H_k)$, we only need to show $f(W_k, H) \leq G(H, H_k)$. To have this, we obtain the Taylor series expansion of $f(W_k, H)$ at $H_k$, and then the $(i, j)$th element of $H$ is
\[
f(W_k, H_{ij}) = f(W_k, H_k)_{ij} + \langle \nabla_H f(W_k, H_k), H_{ij} - H_k_{ij} \rangle + \left( W_k^T W_k \right)_{ii} (H_{ij} - H_k_{ij})^2.
\]
Since $W_k \geq 0$ and $H_k \geq 0$, we have
\[
(W_k^T W_k)_{ii} \leq \sum_l (W_k^T W_k)_{il} H_{klij} = \frac{(W_k^T W_k)_{ij}}{H_{kij}}.
\]
By Theorem 1, we have
\[
\tilde{L}_{jj} = \tilde{D}_{jj} - S_{jj} \leq \tilde{D}_{jj} \leq \sum_l H_{klij} \tilde{D}_{ljj}. \quad \text{(31)}
\]
By substituting (30) and (31) into (29), we prove that $f(W_k, H_{ij}) \leq G(H_{ij}, H_k)$, and thus, $G(H, H_k)$ is an auxiliary function of $f(W_k, H)$. According to Lemma 1, $f(W_k, H)$ is nonincreasing at the minimum of $G(H, H_k)$. By setting $\partial_H G(H, H_k) = 0$, we arrive at $H_{k+1} = H_k \cdot (W_k^T V + \gamma HS)/(W_k^T W_k H_k + \gamma H_k \tilde{D})$, and thus, (27) does not increase the objective function $f(W, H)$. Based on the same procedure, we can show that (28) does not increase the objective function $f(W, H)$ with $H$ fixed.

Although MUR can obtain a local solution, it converges slowly. Thus, we apply the proposed FGD to optimizing FMD-NMF. From Section III-D, we take the procedure for optimizing $H$, for example, and $W$ can be optimized by using the same procedure. At the iteration round $k$, $H_{k+1}$ is updated by
\[
H_{k+1} = H_k + \theta^{k+1} \nabla H_k \quad \text{(32)}
\]
where $\nabla H_k = H_k \cdot (V^T V + \gamma H_k S)/(V^T W H_k + \gamma H_k \tilde{D}) - H_k$, and $\theta^{k+1}$ is the step length that is used to sufficiently decrease the objective function $f(W, H)$ along $\nabla H_k$ and to guarantee the nonnegativity of $H_{k+1}$. Note that MUR in (27) can be deemed as a special case of FGD by setting $\theta^{k+1} = 1$, but MUR converges slowly because the step length, i.e., 1, is not optimal. Thus, similar to (22), $\theta^{k+1}$ can be calculated as
\[
\theta^{k+1} = \min \{ \theta^*, \theta' \}
\]
where $\theta'$ is used to ensure that $H_{k+1}$ does not exceed the nonnegative orthant and does not stay too close to the boundary, and $\theta^*$ is the minimum of
\[
\theta^* = \arg \min_{\theta} f(W, H_k + \theta \nabla H_k).
\]
Different from FGD for MD-NMF, $f(W, H_k + \theta \nabla H_k)$ is of quadratic form with respect to $\theta$, so the optimal solution can be obtained as
\[
\theta^* = \frac{\langle W, V \nabla H_k \rangle - \langle W^T W H_k \nabla H_k \rangle - \gamma \langle L, \nabla H_k \rangle}{\langle \nabla H_k \nabla H_k, W^T W \rangle + \gamma \langle \nabla H_k \nabla H_k, L \rangle}.
\]
(34)

*Proof:*

1. Initialize $H_1 \geq 0, W_1 \geq 0$.
2. For $k = 1, 2, \ldots$,

\[
H_{k+1} = \arg \min_{H \geq 0} f(W_k, P[H - \lambda_H \nabla H f(W_k, H)]) \quad \text{(35)}
\]
\[
W_{k+1} = \arg \min_{W \geq 0} f(P[W - \lambda_W \nabla W f(W, H_{k+1})], H_{k+1}) \quad \text{(36)}
\]

where the operator $P[\cdot]$ projects the negative entries to zero, and $\lambda_H$ and $\lambda_W$ are step lengths for optimizing $H$ and $W$, respectively. Similar to [28], we introduce Armijo’s rule to determine both step lengths. Take the subprocedure for optimizing (35), for example, in the iteration round $k$, the Armijo’s rule is given as follows.
1. Initialize $H^1 = H^*_k$. Given $0 < \rho < 1, 0 < \sigma < 1$.
2. For $t = 1, 2, \ldots$,
   \[
   H^{t+1} = H^t - \rho z_t \nabla_H f(W_k, H^t) \tag{37}
   \]
where the step length of the $t$th iteration $\lambda_{H}^{t} = \rho z_t$, and $z_t$ is the first nonnegative integer $z$, for which following condition is satisfied
\[
 f(H^{t+1}) - f(H^t) \leq \nabla_H f(W_k, H^t), H^{t+1} - H^t, \tag{38}
\]
By trying the step lengths $1, \rho, \rho^2, \ldots$, Bertsekas [8] has proved that the step length $\lambda_{H}^{t}$ always exists and there exists a stationary point of (35) among the limit points of \{ $H^{t}$ \}. Empirically, we set $\sigma = 0.01$ and $\rho = 0.1$ in our experiments. Since computing the objective values in (38) is time-consuming, Lin [28] presented a strategy to reduce the time complexity. The strategy in [28] is that, for any two consecutive iterations, (38) can be rewritten as
\[
 (1 - \sigma)\nabla_H f(W_k, H^t), H^{t+1} - H^t \nonumber \\
+ \frac{1}{2} (H^{t+1} - H^t, (W_k^T W_k) (H^{t+1} - H^t)) \nonumber \\
+ \frac{\gamma}{2} (H^{t+1} - H^t, (H^{t+1} - H^t) \tilde{L}) \leq 0, \tag{39}
\]
The time cost of (39) is small.

According to Lin [28], by adding a large upper bound to all the entries of $H$ and $H$, the problem (24) can be modified as a bound-constrained problem. This means there exists at least one limit point in the sequence \{ $W_k, H_k$ \} generated by PGD for such bound-constrained problem. Therefore, PGD for FMD-NMF converges to a stationary point.

### Table II

<table>
<thead>
<tr>
<th>Objective Value</th>
<th>MUR</th>
<th>FGD</th>
<th>PGD</th>
</tr>
</thead>
<tbody>
<tr>
<td>YALE(1600,105)</td>
<td>798.520</td>
<td>720.227</td>
<td>712.221</td>
</tr>
<tr>
<td>ORL(1024,200)</td>
<td>625.230</td>
<td>566.647</td>
<td>482.291</td>
</tr>
<tr>
<td>UMIST(1600,60)</td>
<td>390.471</td>
<td>349.798</td>
<td>345.766</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time (second)</th>
<th>MUR</th>
<th>FGD</th>
<th>PGD</th>
</tr>
</thead>
<tbody>
<tr>
<td>YALE(1600,105)</td>
<td>25</td>
<td>0.86</td>
<td>0.88</td>
</tr>
<tr>
<td>ORL(1024,200)</td>
<td>12</td>
<td>0.87</td>
<td>0.71</td>
</tr>
<tr>
<td>UMIST(1600,60)</td>
<td>8</td>
<td>0.87</td>
<td>0.63</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Iteration Number</th>
<th>MUR</th>
<th>FGD</th>
<th>PGD</th>
</tr>
</thead>
<tbody>
<tr>
<td>YALE(1600,105)</td>
<td>25</td>
<td>0.86</td>
<td>0.88</td>
</tr>
<tr>
<td>ORL(1024,200)</td>
<td>12</td>
<td>0.87</td>
<td>0.71</td>
</tr>
<tr>
<td>UMIST(1600,60)</td>
<td>8</td>
<td>0.87</td>
<td>0.63</td>
</tr>
</tbody>
</table>

**B. Computational Complexity**

We first analyze the computational complexities of MUR, FGD, and PGD for FMD-NMF, and then compare them on three real-world datasets. Since MUR, FGD, and PGD are all alternating optimization algorithms, it is reasonable to evaluate their time costs by comparing their complexities of one iteration round.

1. For MUR, by using the trick in [28], we compute $WHH^T$ as $W(HH^T)$, and thus, reduce the computational complexity of (27) and (28) from $O(mnr)$ to $O(\max\{m,n\}r^2)$ for $r \ll \min\{m,n\}$. Therefore, the complexity of MUR is $O(\max\{m,n\}r^2 + r\gamma^2)$.  
2. For FGD, the main time cost is spent on calculating the step lengths in (35) and (36). Since the step lengths are obtained by Armijo’s rule, from [28], the complexity of PGD is $O(mnr) + \#iterations \times O(mnr^2)$, wherein $\gamma$ is the trial number of checking the (38) and $\#iterations$ is the iteration number of the subprocedure. Though PGD can obtain the stationary point (see Theorem 5), the time cost is prohibitive, especially when both $\#iterations$ and $\gamma$ is relatively large.

Table II compares the objective values, CPU seconds and the corresponding iteration numbers of MUR, FGD, and PGD for optimizing FMD-NMF on YALE [6], ORL [36], and UMIST [19] dataset. The experimental setting is the same as that used in Section III-D. For each test, initializations and parameter settings of MUR, FGD, and PGD are identical. Table II shows that FGD converges in fewer CPU seconds and obtains lower objective values than MUR. Although PGD could obtain slightly lower objective values than FGD, it spends more CPU seconds in all problems on all tests. Therefore, FGD is an efficient method for optimizing FMD-NMF. In Section V-G, we further show that
FGD performs even better than PGD for optimizing FMD-NMF in terms of face-recognition accuracy.

V. EXPERIMENTS

Here, we evaluate the performance of the proposed MD-NMF comparing with six representative algorithms, which are NMF [24], PCA [18], LNMF [26], GNMF [12], DNMF [47], and NGE [44], on four popular face image datasets including YALE [6], ORL [36], UMIST [19], and CMU PIE [19]. Fig. 5 shows example images of YALE, ORL, UMIST, and PIE datasets. All face images of four datasets were aligned according to the eye position. Each pixel of images was linearly rescaled to the gray level of 256, and each image was rearranged to a vector. Different numbers (3, 5, 7) of images were randomly selected from each individual to constitute the training set, and the rest images make up the test set. The training set was used to learn basis for the low-dimensional space. The test set was used to report the accuracy of face recognition in the learned low-dimensional space. The accuracy was calculated as the percentage of samples in the test set that were correctly classified using the NN rule. These trials were independently conducted five times and the averaged accuracy was reported. In our experiment, we also evaluate the performance of the variant FMD-NMF and set all the parameters in (24) same as that of MD-NMF, respectively.

A. YALE Dataset

In the YALE [6] dataset, there are 165 frontal view face images of 15 individuals. Eleven images were collected from each individual with varying facial expressions (smile or sad) and configurations and each image was normalized to $40 \times 40$ pixel array and reshaped to a vector. Fig. 4 shows the average face-recognition accuracy versus the dimension of the subspace. Table III gives the best accuracy and corresponding dimension of all the algorithms. According to Fig. 4(a) and (b), it can be seen that MD-NMF outperforms most algorithms and is comparable with LNMF, as shown in Fig. 4(c). Considering the best accuracy, Table III shows that: 1) MD-NMF is superior to NMF and its representative variants, and 2) FMD-NMF is comparable to NGE.

B. ORL Dataset

The Cambridge ORL [36] dataset consists of 400 images collected from 40 individuals. There are ten images for each individual with varying lighting, facial expressions and facial details (with-glasses or no-glasses). All images were taken in the same dark background, and each image was normalized to $32 \times 32$ pixel array and reshaped to a long vector. Fig. 6 shows the average face-recognition accuracy versus the dimension of the subspace. Table IV gives the best recognition accuracy and corresponding dimension for all the algorithms. Fig. 6 and Table IV
Fig. 6. Face-recognition accuracy on the ORL dataset. We randomly selected three (a), five (b), and seven images from each person to train a model and used the rest images in the dataset for test, so we had three types of partitions on this dataset. We conducted five trials for each partition and compared the performance of different algorithms based on the averaged accuracy of the five trials on each dimension for each type of the partition.

Fig. 7. Face-recognition accuracy on the UMIST dataset. We randomly selected (a) three, (b) five, (c) and seven images from each person to train a model and used the rest images in the dataset for test, so we had three types of partitions on this dataset. We conducted five trials for each partition and compared the performance of different algorithms based on the averaged accuracy of the five trials on each dimension for each type of the partition.

<table>
<thead>
<tr>
<th>TABLE IV</th>
<th>BEST ACCURACY ON THE ORL DATASET</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm</td>
<td>3</td>
</tr>
<tr>
<td>PCA [18]</td>
<td>0.800(115)</td>
</tr>
<tr>
<td>NMF [24]</td>
<td>0.775(25)</td>
</tr>
<tr>
<td>LNMF [26]</td>
<td>0.804(95)</td>
</tr>
<tr>
<td>GNMF [12]</td>
<td>0.776(25)</td>
</tr>
<tr>
<td>DNMF [47]</td>
<td>0.806(50)</td>
</tr>
<tr>
<td>NGE [44]</td>
<td>0.804(75)</td>
</tr>
<tr>
<td>MD-NMF</td>
<td><strong>0.854</strong>*(115)</td>
</tr>
<tr>
<td>FMD-NMF</td>
<td>0.838(90)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE V</th>
<th>BEST ACCURACY ON THE UMIST DATASET</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm</td>
<td>3</td>
</tr>
<tr>
<td>PCA [18]</td>
<td>0.713(60)</td>
</tr>
<tr>
<td>NMF [24]</td>
<td>0.640(20)</td>
</tr>
<tr>
<td>LNMF [26]</td>
<td>0.713(55)</td>
</tr>
<tr>
<td>GNMF [12]</td>
<td>0.696(80)</td>
</tr>
<tr>
<td>DNMF [47]</td>
<td>0.696(15)</td>
</tr>
<tr>
<td>NGE [44]</td>
<td>0.698(20)</td>
</tr>
<tr>
<td>MD-NMF</td>
<td><strong>0.802</strong>*(105)</td>
</tr>
<tr>
<td>FMD-NMF</td>
<td>0.736(20)</td>
</tr>
</tbody>
</table>

show that MD-NMF outperforms NMF and its variants, and FMD-NMF performs comparable with NGE.

C. UMIST Dataset

The UMIST [19] dataset contains 575 face photos collected from 20 people. At least 41 and at most 82 photos were taken from each person varying in poses from profile to frontal views. Each photo was normalized to 40 x 40 pixel array and reshaped to a vector. Fig. 7 shows the average accuracies versus the subspace dimension. The best accuracy and corresponding dimension for each algorithm are given in Table V. Table V shows that MD-NMF outperforms other algorithms. Considering the best accuracy, Table V also shows that FMD-NMF performs comparably with NGE.

D. PIE Dataset

The CMU PIE [38] dataset contains 41 368 face photos collected from 68 subjects. For each subject, images are of 13 poses, 43 illumination conditions, and four expressions. In our
experiment, a subset of images with pose ID C27 and different illumination conditions is used, and thus, there are 21 images for each subject. Each face image was normalized to 32×32 pixel array and reshaped to a vector. Fig. 8 shows the average accuracies versus the subspace dimensionality. The best accuracy and corresponding dimensionality for each algorithm are given in Table VI. Considering the best accuracy, Table VI shows that MD-NMF and FMD-NMF performed almost perfectly and all NMF related algorithms perform well on this dataset.

We experimentally selected these best subspace dimensions for respective NMF-related algorithms. From Tables III to VI, each number in parentheses shows the best subspace dimension for the corresponding algorithm on each dataset. This strategy is adopted in recent NMF related publications, such as [27] and [48], for performance evaluation. Curves in Figs. 4, 6, and 7 show that the accuracy curve of MD-NMF is usually above the accuracy curves of other algorithms, and thus, MD-NMF usually performs better than the compared baseline algorithms and MD-NMF generalizes better than the compared baseline algorithms in terms of subspace dimension.

The best subspace dimensions for MD-NMF are higher than those for NMF, DNMF, and GNMF, because the bases learned by MD-NMF are much sparser than those learned by other NMF variants (see Fig. 10 and Table VII). Since LNMF learns sparsest bases (see Table VII), its best subspace dimensions are relatively higher, as shown in Tables IV–VI.

### E. Parts-Based Learning

This section studies the ability of parts-based representation of the proposed MD-NMF algorithm. From [24], PCA learns a distributed representation, in which each face is represented by all the basis images or Eigenfaces [6]. By using the non-negativity constraint, NMF learns basis images considered to be parts-based, which contain several versions of mouths, noses and other facial parts. However, it fails to obtain the parts-based representation on some database (see Fig. 2).

We compare the bases learned by the proposed MD-NMF and FMD-NMF with those learned by NMF, LNMF, and NGE, on different datasets, i.e., YALE [6], ORL [36], UMIST [19], and PIE [38]. Fig. 10 presents these bases for subspace of dimensionality 25, sorted in descending order of the diagonal elements of $HH^T$, according to Li et al. [26]. The higher pixel is shown in darker color. These figures indicate that the proposed MD-NMF

---

**TABLE VI**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>3</th>
<th>5</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA [18]</td>
<td>0.727(120)</td>
<td>0.874(120)</td>
<td>0.944(120)</td>
</tr>
<tr>
<td>NMF [24]</td>
<td>0.999(100)</td>
<td>1.000(80)</td>
<td>1.000(80)</td>
</tr>
<tr>
<td>LNMF [26]</td>
<td>0.713(120)</td>
<td>0.859(120)</td>
<td>0.928(120)</td>
</tr>
<tr>
<td>GNMF [12]</td>
<td>0.795(115)</td>
<td>0.802(110)</td>
<td>0.803(110)</td>
</tr>
<tr>
<td>DNMF [47]</td>
<td>0.971(120)</td>
<td>0.992(120)</td>
<td>0.998(120)</td>
</tr>
<tr>
<td>NGE [44]</td>
<td>0.986(120)</td>
<td>0.997(120)</td>
<td>1.000(120)</td>
</tr>
<tr>
<td>MD-NMF</td>
<td><strong>1.000(90)</strong></td>
<td><strong>1.000(60)</strong></td>
<td><strong>1.000(35)</strong></td>
</tr>
<tr>
<td>FMD-NMF</td>
<td>0.999(70)</td>
<td>1.000(55)</td>
<td>1.000(25)</td>
</tr>
</tbody>
</table>

---

Fig. 8. Face-recognition accuracy on the PIE dataset. We randomly selected (a) three, (b) five, and (c) seven images from each person to train a model and used the rest images in the dataset for test, so we had three types of partitions on this dataset. We conducted five trials for each partition and compared the performance of different algorithms based on the averaged accuracy of the five trials on each dimension for each type of the partition.

Fig. 9. Face-recognition accuracy versus (a) $\alpha$ with $\beta$, $\gamma$ fixed, (b) $\beta$ with $\alpha$, $\gamma$ fixed, and (c) $\gamma$ with $\alpha$, $\beta$ fixed on YALE, ORL, and UMIST dataset. The proposed MD-NMF model (12) is stable when varying $\alpha$, $\beta$, and $\gamma$ within $[10^{-4}, 10^{-2})$, $(10^{-2}, 1)$, and $(10, 100)$, respectively.
and its variant FMD-NMF could learn parts-based representation. According to Hoyer [23], the sparseness can be used to quantify the basis images, and the sparseness was defined as

$$\text{sparseness}(w) = \frac{\sqrt{n} - \sum |w_k|}{\sqrt{n} - 1}$$

where $n$ is the dimensionality of vector $w$. Table VII shows the average sparseness of the columns in the learned basis by NMF, LNM, NGE, MD-NNF, and FMD-NMF. Both MD-NMF and FMD-NMF bases are sparser than those of NMF and NGE. This is consistent with bases shown in Fig. 10. LNM bases are sparser than our methods, but it cannot perform as well as MD-NMF. This is because it ignores both the discriminative information and the local geometry. It is interesting that the FMD-NMF bases are sparser than those of MD-NMF. This is because the PGD can obtain the stationary point for FMD-NMF, while the FGD for MD-NMF cannot; however, MD-NMF performs well as shown in our experiment.

**F. Parameter Selection**

In the proposed MD-NMF (12), there are three tradeoff parameters, i.e., $\alpha$, $\beta$, and $\gamma$. It is time-consuming to select all these parameters based on the grid search. Fortunately, $\alpha$, $\beta$, and $\gamma$ affect the performance slightly if they are set in feasible ranges that can be obtained by using the following procedure: 1) for a given dataset $V$, optimize $W$ and $H$ by setting all parameters to zero; 2) calculate $S_\alpha = KL(V,WH)/\text{tr}(WHW^T)$, $S_\beta = KL(V,WH)/\text{tr}(HH^T)$, and $S_\gamma = KL(V,WH)/\text{tr}(HLH^T)$; and 3) set the range of $\alpha$ as $[S_\alpha \times 10^{-2}, S_\alpha \times 10^2]$, set the range of $\beta$ as $[S_\beta \times 10^{-3}, S_\beta \times 10^2]$ and the range of $\gamma$ as $[S_\gamma \times 10^{-3}, S_\gamma \times 10^2]$. Fig. 9 shows this strategy works well on YALE [6], ORL [36], and UMIST [19] datasets. In this experiment, we randomly selected three images from each identity to form the training set and the remainder images for test. These trails were independently performed five times, and the average classification accuracy was reported. The recognition accuracies are consistent when each of $\alpha$, $\beta$, and $\gamma$ is selected from a wide range. In all the above experiments, we set $\alpha = 0.01$, $\beta = 0.1$, and $\gamma = 100$.  

![Fig. 10. Learned bases of the (first row) YALE, (second row) ORL, (third row) UMIST, and (fourth row) PIE datasets. Each subfigure represents a basis vector, and the higher value pixel is shown in darker color. The LNM (column (b)) obtains the sparsest bases. NGE bases (column (c)) are parts-based, and the MD-NMF bases (column (d)) as well as the FMD-NMF bases (column (e)) are sparser than those of NGE and NMF (see Table VII). This is because the proposed MD-NMF explicitly guarantees this parts-based representation in (12).](image-url)
Another two critical parameters are the number of NNs ($k_1, k_2$) used to construct the two adjacent graphs. We studied their effects on the face-recognition accuracy on the YALE [6] dataset, where the training set is comprised of seven images randomly chosen from each individual. By definition of these adjacent graphs, $k_1$ varies from 1 to $\frac{N}{(C)} - 1$ wherein $N$ is the number of samples in the training set and $C$ is the class number, and $k_2$ varies from 1 to $\frac{N}{(C)} \times (C - 1)$. In this experiment, $1 \leq k_1 \leq 6$ and $1 \leq k_2 \leq 98$, $C = 15$ and $N = 105$. Fig. 11(a) presents the face-recognition accuracy versus $k_2$ with $k_1$ fixed to 4. A peak arises when $k_2 = 37$. Fig. 11(b) shows the face-recognition accuracy versus $k_1$ with $k_2$ fixed to 37. There appears a peak when $k_1 = 4$. It can be seen that the performance varies with different $k_1$ and $k_2$. Table VIII presents the structures of adjacent graphs ($k_1, k_2$) with which the best performance was achieved on the YALE [6], ORL [36], and UMIST [19] datasets, respectively. Table VIII shows that the algorithm performs well when $k_3$ is selected around $N_c - 1$, wherein $N_c$ is the sample number in each class of the training set. Thus, we fix $k_1 = \min\{N_c, 3\}$, and set $k_2 = 20$ for experiments on the PIE [38] dataset. Fig. 8 and Table VI show that the proposed MD-NMF performs well under this setting.

### G. FGD Versus PGD for FMD-NMF

To evaluate the effectiveness of stationary point for FMD-NMF, we solved FMD-NMF with PGD as well as FGD and compared their face-recognition performance on the ORL [36] dataset. The experimental setting is the same as Section V-B. Fig. 12 shows the average face-recognition accuracy versus the dimension of the subspace. Although FGD does not guarantee the convergence to stationary point like PGD (see Section IV-A), the FMD-NMF solved by FGD outperforms the FMD-NMF solved by PGD, as shown in Fig. 12. Thus, FGD works well for optimizing both MD-NMF and FMD-NMF.

### VI. CONCLUSION

In this paper, we proposed the MD-NMF that incorporates both local geometry and discriminative information in the learned subspace. We built two types of adjacent graphs on each sample, according to the sample label information. One
was used to build up the regularization that perversely the data geometric structure, and the other was used to maximize the margins of different classes.

We introduced an MUR to optimize MD-NMF and theoretically showed its convergence. Since MUR converges slowly, we further proposed an FGD to optimize MD-NMF. FGD uses a Newton method to search the optimal step length, and thus, it converges quickly. We also proved the convergence of FGD. In addition, FGD includes MUR as a special case and can be applied to optimizing NMF and its variants.

Empirical studies show that FGD converges much faster than MUR for optimizing MD-NMF. The application in the variant of MD-NMF also shows the efficiency of FGD. Experimental results on popular face image datasets suggest the effectiveness of MD-NMF compared with NMF and its representative variants.

**APPENDIX A**

**MULTIPLICATIVE UPDATE RULE**

In order to simplify derivations of the MUR for MD-NMF, we rewrite objective function (12) in two subfunctions

\[
\begin{align*}
F_1(W, H) &= \sum_{i,j} \left( V_{ij} \log \frac{V_{ij}}{\sum_k W_{ik}H_{kj}} - V_{ij} + \sum_k W_{ik}H_{kj} \right) \\
F_2(W, H) &= \alpha \text{tr}(W eW^T) + \frac{\beta}{2} \text{tr}(H H^T) + \gamma \left( H \left( I_c^{-1/2} L_g L_c^{-1/2} H^T \right) \right)
\end{align*}
\]

(40)

where \( F_1 \) is the KL divergence between \( V \) and \( WH \). The first-order partial derivatives of \( F_2 \) with respect to \( W_{ij} \) and \( H_{ij} \) are

\[
\begin{align*}
\frac{\partial F_2}{\partial W_{ij}} &= \alpha (We)_{ij} \\
\frac{\partial F_2}{\partial H_{ij}} &= \beta H_{ij} + \gamma \left( H \left( I_c^{-1/2} L_g L_c^{-1/2} H^T \right) \right)_{ij}
\end{align*}
\]

(41)

The first-order partial derivatives of \( F_1 \) with respect to \( W_{ij} \) and \( H_{ij} \) are

\[
\begin{align*}
\frac{\partial F_1}{\partial W_{ij}} &= -\sum_k \frac{V_{ij} H_{kl}}{\sum_k W_{ik}H_{kl}} + \sum_t H_{jt} \\
\frac{\partial F_1}{\partial H_{ij}} &= -\sum_k \frac{V_{ij} H_{kl}}{\sum_k W_{ik}H_{kl}} + \sum_t W_{ti}
\end{align*}
\]

(42)

According to (41) and (42), we have the gradients of \( F \) with respect to \( W_{ij} \) and \( H_{ij} \)

\[
\begin{align*}
\nabla W_{ij} &= -\sum_k \frac{V_{ij} H_{jkl}}{\sum_k W_{ik}H_{kl}} + \sum_t H_{jt} + \alpha (W^t e)_{ij} \\
\nabla H_{ij} &= -\sum_k \frac{V_{ij} H_{jkl}}{\sum_k W_{ik}H_{kl}} + \sum_t W_{ti} + \beta H_{ij} + \gamma \left( H \left( I_c^{-1/2} L_g L_c^{-1/2} H^T \right) \right)_{ij}
\end{align*}
\]

(44)

By some algebra, we can rewrite the gradients (43) and (44) in positive and negative parts as follows:

\[
\begin{align*}
\nabla W_{ij}^+ &= \nabla W_{ij} - \nabla W_{ij}^- \\
\nabla H_{ij}^+ &= \nabla H_{ij} - \nabla H_{ij}^-
\end{align*}
\]

(45)

where

\[
\begin{align*}
\nabla W_{ij}^+ &= \sum_k H_{jkl} + \alpha (W^t e)_{ij} \\
\nabla W_{ij}^- &= \sum_k \frac{V_{ij} H_{jkl}}{\sum_k W_{ik}H_{kl}} \\
\nabla H_{ij}^+ &= \sum_t W_{ti} + \beta H_{ij} + \gamma (H^t D)_{ij} \\
\nabla H_{ij}^- &= \sum_k \frac{V_{ij} H_{jkl}}{\sum_k W_{ik}H_{kl}} + \gamma (H^t S)_{ij}
\end{align*}
\]

and \( D = L_c^{-1/2} D_g (L_c^{-1/2})^T, S = L_c^{-1/2} S_g (L_c^{-1/2})^T, D_g \) is the diagonal of the graph Laplacian matrix \( L_g \), and \( S_g = D_g - L_g \). Both the positive gradients \( \nabla W_{ij}^+, \nabla H_{ij}^+ \) and the negative gradients \( \nabla W_{ij}^-, \nabla H_{ij}^- \) are nonnegative because the sample matrix \( V \) is nonnegative, \( W \) and \( H \) are initialized nonnegative, \( D \) and \( S \) are also nonnegative guaranteed by Theorem 1, which will be proved in Appendix B. According to (45), at the iteration round \( t \), we update both \( W \) and \( H \) in their negative gradient directions

\[
\begin{align*}
W_{ij}^{t+1} &= W_{ij}^t - \eta_W^t \nabla W_{ij}^- \\
H_{ij}^{t+1} &= H_{ij}^t - \eta_H^t \nabla H_{ij}^-
\end{align*}
\]

To guarantee the nonnegativity of \( W \) and \( H \) during the iterative procedure, we select

\[
\begin{align*}
\eta_W^t &= \theta_W \frac{W_{ij}^t}{\nabla W_{ij}^+} (\theta_W > 0) \\
\eta_H^t &= \theta_H \frac{H_{ij}^t}{\nabla H_{ij}^+} (\theta_H > 0)
\end{align*}
\]

where \( \theta_W \) and \( \theta_H \) are the step lengths of factor \( W \) and \( H \) respectively. Accordingly to Lee and Seung [25], at each iteration round, we set step lengths \( \theta_W = \theta_H = 1 \), i.e.,

\[
\begin{align*}
W_{ij}^{t+1} &= W_{ij}^t \nabla W_{ij}^- \nabla W_{ij}^+ \\
H_{ij}^{t+1} &= H_{ij}^t \nabla H_{ij}^- \nabla H_{ij}^+
\end{align*}
\]

(46)

It is direct that the optimization procedure converges when the gradient vanishes, i.e., \( \nabla W_{ij}^- = \nabla W_{ij}^+ = \nabla W_{ij} = 0 \). It is intrinsically a scaled gradient descent method, where the descent direction \( P \) is the negative gradient direction \( -\nabla D \) scaled by a
positive matrix \( d > 0 \), i.e., \( P = -d \odot \nabla D \). According to (46), we obtain the MUR

\[
W_{ij}^{t+1} = W_{ij}^{t} + \frac{\sum_{l}(V_{ik}H_{jl}^t / \sum_k W_{ik}^t H_{kj}^t)}{\sum_i H_{jl}^t + \alpha(W^t)e_{ij}} \tag{47}
\]

\[
H_{ij}^{t+1} = H_{ij}^{t} + \frac{\sum_i(V_{ik}W_{ij}^t / \sum_k W_{ik}^t H_{kj}^t) + \gamma(H^tS)_{ij}}{\sum_i W_{hi}^t + \beta H_{ij}^t + \gamma(H^tD)_{ij}} \tag{48}
\]

**APPENDIX B**

**PROOF OF NONNEGATIVITY**

Before we prove Theorem 1, let us define \( M \)-matrix.

**Definition 2:** Matrix \( B \) is an \( M \)-matrix if the following two conditions hold:

1) the off-diagonal entries are nonpositive, i.e., \( B_{ij} \leq 0 \), \( i \neq j \);

2) the real parts of all eigenvalues are positive.

Based on **Definition 2**, we come up with the following two lemmas according to [7] and [1], respectively.

**Lemma 5:** If \( B \) is an \( M \)-matrix and invertible, then the inverse of the matrix \( B \) is nonnegative matrix, i.e., \( B^{-1} \succeq 0 \).

**Lemma 6:** If \( B \) is a symmetric positive-definite \( M \)-matrix, then the square root of \( B \), namely \( B^{1/2} \), exists and is also a symmetric positive-definite \( M \)-matrix.

**Theorem 1:** Both \( D \) and \( S \) are nonnegative matrices.

**Proof:** Recall that \( D = (L_c^{-1/2})^T D_g L_c^{-1/2} \) and \( S = (L_c^{-1/2})^T S_g L_c^{-1/2} \), wherein \( D_g \) and \( S_g \) are nonnegative matrices. According to (8), \( L_c \) is symmetric positive definite and also an \( M \)-matrix from definition of graph Laplacian matrix. According to Lemma 6, the square root of \( L_c \) is an \( M \)-matrix, i.e., \( L_c^{-1/2} \) is an \( M \)-matrix, so \( L_c^{-1/2} \) is a nonnegative matrix according to Lemma 5. Since \( D \) is a product of nonnegative matrices, it is nonnegative. We have the same result for \( S \). This completes the proof.

**APPENDIX C**

**PROOFS OF CONVERGENCE**

We prove the convergence by constructing two auxiliary functions for \( F(W, H) \) with fixed \( W \) and \( H \), respectively. Based on (40), we construct auxiliary functions for \( F_1 \) and \( F_2 \), and add them together.

**Lemma 2:** The objective function (12) is nonincreasing by using the update (13).

**Proof:** Given \( H^t \), according to (40), we rewrite \( F_1 \) as

\[
F_1(W, H^t) = \sum_{i,j} \left( V_{ij} \log V_{ij} - V_{ij} \log \sum_k W_{ik} H_{kj}^t - V_{ij} + \sum_k W_{ik} H_{kj}^t \right) \tag{49}
\]

where \( \sum_k \rho_k = 1 \) for any \( 0 < \rho_k < 1 \). The inequality holds because the negative logarithmic function is convex. By setting \( \rho_k = (V_{ik} H_{kj}^t) / (\sum_k V_{ik} H_{kj}^t) \), we have an auxiliary function for \( F_1(W, H^t) \),

\[
G_1(W, H^t) = \sum_{i,j} \left( V_{ij} \log V_{ij} - V_{ij} + \sum_k W_{ik} H_{kj}^t \right) \tag{50}
\]

It is direct to verify that \( G_1(W, W^t) = F(W, H^t) \). By adding \( F_2 \) and \( G_1 \) together, we have

\[
G(W, H^t) = G_1(W, W^t) + F_2(W, H^t)
\]

as an auxiliary function for \( F(W, H^t) \). The Taylor series expansion of \( F_2(W, H^t) \) at \( W_{ij} \) is

\[
F_2(W, H^t) = \alpha(W^t e_{ij}) (W_{ij} - W_{ij}^t) + C,
\]

where \( C \) is a constant, because its second-order derivative at \( W_{ij}^t \) is zero. According to (49) and (41), we then obtain the first-order derivative of \( G(W, W^t) \) with respect to \( W_{ij} \),

\[
\frac{\partial G(W, W^t)}{\partial W_{ij}} = - \sum_l \frac{V_{il} H_{jl}}{\sum_k W_{ik} H_{kj}} \frac{1}{W_{ij}} + \sum_l H_{jl} + \alpha(W^t e_{ij}) \tag{51}
\]

By setting \((\partial G(W, W^t))/\partial W_{ij} = 0\), we have

\[
- \sum_l \frac{V_{il} H_{jl}}{\sum_k W_{ik} H_{kj}} \frac{W_{lj}}{W_{ij}} + \sum_l H_{jl} + \alpha(W^t e_{ij}) = 0. \tag{52}
\]

By solving (50), we get the minimum of \( G(W, W^t) \).

\[
W_{ij} = W_{ij}^t \frac{\sum_l (V_{il} H_{jl}^t / \sum_k W_{ik} H_{kj}^t)}{\sum_l H_{jl}^t + \alpha(W^t e_{ij})}. \tag{53}
\]

Equation (51) is the update rule in (47). By rewriting (47) in matrix form and considering Lemma 1, we complete the proof.

**Lemma 3:** Given an arbitrary positive number \( \lambda > 0 \), for any two positive variables \( a > 0 \) and \( b > 0 \), the following inequality holds:

\[
a - b + \lambda \left( \frac{a - b}{2} \right)^2 \geq b \log \frac{a}{b}. \tag{54}
\]

**Proof:** We construct an auxiliary function

\[
\varphi(a, b) = a - b + \lambda \left( \frac{a - b}{2} \right)^2 - b \log \frac{a}{b}, \quad a, b > 0.
\]
Its first-order partial derivatives are
\[
\frac{\partial \varphi}{\partial a} = 1 + \lambda (a - b) - b \quad \text{and} \quad \frac{\partial \varphi}{\partial b} = \lambda (b - a) - \log \frac{a}{b} \bigg|_{a=b} = 0.
\]
Thus, \((a,a)\) is a stationary point of \(\varphi(a,b)\) for any \(a > 0\). According to (52), we obtain the second-order derivatives of \(\varphi(a,b)\) as follows:
\[
\varphi_{aa} = \frac{\partial^2 \varphi}{\partial a^2} = \lambda + \frac{b}{a^2},
\]
\[
\varphi_{ab} = \frac{\partial^2 \varphi}{\partial a \partial b} = -\lambda - \frac{1}{a},
\]
\[
\varphi_{bb} = \frac{\partial^2 \varphi}{\partial b^2} = \lambda + \frac{1}{b}.
\]
Since the second-order derivatives (53) satisfy
\[
\Delta = \varphi_{aa} \varphi_{bb} - \varphi_{ab}^2 = \frac{(a-b)^2}{a^2 b} \lambda \geq 0
\]
where \(\Delta = 0\) only if \(a = b\) because \(\lambda > 0\). In addition, \(\varphi(a,a) > 0\). Thus, the stationary point \((a,a)\) is the minimum of \(\varphi(a,b)\). Then, we have
\[
\varphi(a,b) \geq \varphi(a,a) = 0.
\]
It completes the proof.

**Lemma 4:** The objective function (12) is nonincreasing by using the update (14).

**Proof:** Given \(W\), since the nonnegative logarithmic function is convex, we can construct an auxiliary function for \(F_1(W^t, H)\), in the same way used to construct \(G_1(W, W^t)\), as
\[
G_1(H, H^t) = \sum_{i,j} \left( V_{ij} \log V_{ij} - V_{ij} + \sum_k W_{ik} H_{kj} \right)
- V_{ij} \sum_k \left( \frac{W_{ik} H_{kj}^t}{\sum_k W_{ik} H_{kj}^t} \log W_{ik} H_{kj} - \frac{W_{ik} H_{kj}^t}{\sum_k W_{ik} H_{kj}^t} \log \sum_k W_{ik} H_{kj}^t \right) \quad (54)
\]
We can verify that \(G_1(H, H) = F(W^t, H)\). According to (40), we rewrite \(F_2(W^t, H)\) as
\[
F_2(W^t, H) = \frac{\alpha}{2} \text{tr}(W^t e W^T e) + \frac{\beta}{2} \text{tr}(H H^T) + \gamma^2 \text{tr}(H D H^T) - \frac{\gamma^2}{2} \text{tr}(H S H^T).
\]
Its first-order and second-order derivatives are
\[
\frac{\partial F_2(W^t, H)}{\partial H_{ij}} = \beta H_{ij} + \gamma (HD)_{ij} - \gamma (HS)_{ij}
\]
\[
\frac{\partial^2 F_2(W^t, H)}{\partial H_{ij}^2} = \beta + \gamma (D_{jj} - S_{jj}).
\]
Thus, we have the Taylor-series expansion of \(F_2(W^t, H)\) at \(H_{ij}\)
\[
F_2(H_{ij}) = F_2(H_{ij}^t) + \beta H_{ij} (H_{ij} - H_{ij}^t) + \frac{1}{2} \beta (H_{ij} - H_{ij}^t)^2,
\]
\[
\gamma (H^T D)_{ij} (H_{ij} - H_{ij}^t) + \frac{1}{2} \gamma D_{jj} (H_{ij} - H_{ij}^t)^2,
\]
\[
- \gamma (H^T S)_{ij} \left( H_{ij} - H_{ij}^t + \frac{1}{2} \beta (H^T S)_{ij} (H_{ij} - H_{ij}^t)^2 \right)
\]
\[
\leq F_2(H_{ij}^t) + \beta H_{ij} (H_{ij} - H_{ij}^t) + \frac{1}{2} \beta (H_{ij} - H_{ij}^t)^2,
\]
\[
+ \gamma (H^T D)_{ij} (H_{ij} - H_{ij}^t) + \frac{1}{2} \gamma (H^T D)_{ij} (H_{ij} - H_{ij}^t)^2,
\]
\[
- \gamma (H^T S)_{ij} \left( H_{ij} - H_{ij}^t + \frac{1}{2} \beta (H^T S)_{ij} (H_{ij} - H_{ij}^t)^2 \right) \quad (56)
\]
where the inequality comes from \(D_{jj} \leq ((H^T D)_{ij})/(H_{ij}^t)\). Recall that \(S = (L^{-1/2})^T S_y L^{-1/2}\) and \(S_y\) is positive definite, so \(S_y > 0\). By using Lemma 3, we come up with the inequality
\[
H_{ij} - H_{ij}^t + \frac{1}{2} \beta (H^T S)_{ij} (H_{ij} - H_{ij}^t)^2 \geq H_{ij}^t \log \frac{H_{ij}}{H_{ij}^t} \quad (57)
\]
According to inequalities (56) and (57), we have
\[
F_2(H_{ij}) \leq F_2(H_{ij}^t) + \beta H_{ij} (H_{ij} - H_{ij}^t) + \frac{1}{2} \beta (H_{ij} - H_{ij}^t)^2,
\]
\[
+ \gamma (H^T D)_{ij} (H_{ij} - H_{ij}^t) + \frac{1}{2} \gamma (H^T D)_{ij} (H_{ij} - H_{ij}^t)^2,
\]
\[
- \gamma (H^T S)_{ij} H_{ij}^t \log \frac{H_{ij}}{H_{ij}^t} \quad (58)
\]
We construct an auxiliary function of \(F_2(W^t, H_{ij})\) by using the right-hand side item in above inequality
\[
G_2(H_{ij}, H_{ij}^t) = F_2(H_{ij}^t) + \beta H_{ij}^t (H_{ij} - H_{ij}^t) + \frac{1}{2} \beta (H_{ij} - H_{ij}^t)^2,
\]
\[
+ \gamma (H^T D)_{ij} (H_{ij} - H_{ij}^t) + \frac{1}{2} \gamma (H^T D)_{ij} (H_{ij} - H_{ij}^t)^2,
\]
\[
- \gamma (H^T S)_{ij} H_{ij}^t \log \frac{H_{ij}}{H_{ij}^t} \quad (59)
\]
We can verify that \( G_2'(H_{ij}, H_{ji}) = F_2'(H_{ij}) \). By combining (54) and (59), we construct an auxiliary function for \( F(W^t, H) \) as
\[
G(H, H^t) = G_1(H, H^t) + G_2(H, H^t),
\]
(60)
From (60), we obtain the first-order derivative of \( G(H, H^t) \)
\[
\frac{\partial G(H, H^t)}{\partial H_{ij}} = -\sum_k V_{kj} W_{i} W_{kj}^H H_{ij}^H
- \sum_l W_{i}^H + \beta H_{ij}^H
+ \gamma (H^t D)_{ij} - (H^t S)_{ij} H_{ij}^H H_{ij}^H,
\]
By setting \( (\partial G(H, H^t)) / (\partial H_{ij}) = 0 \), we obtain the minimum of \( G(H, H^t) \) as
\[
H_{ij} = H_{ij}^H - \frac{\sum_k V_{kj} W_{i} W_{kj}^H + \gamma (H^t S)_{ij}}{\sum_l W_{i}^H + \beta H_{ij}^H + \gamma (H^t D)_{ij}}.
\]
(61)
Equation (61) is the update rule in (48). By rewriting (48) in matrix form and considering Lemma 1, we complete the proof.

ACKNOWLEDGMENT

The authors would like to thank the handling associate editor Prof. B. Sankur and anonymous reviewers for their constructive comments to this paper.

REFERENCES


Naiyang Guan received the B.S. and M.S. degrees from the National University of Defense Technology, Changsha, China, where he is currently working toward the Ph.D. degree with the School of Computer Science.

From October 2009 to October 2010, he was a Visiting Student with the School of Computer Engineering, Nanyang Technological University, Singapore. His current research interests include computer vision, image processing, and convex optimization.

Dacheng Tao (M’07) is currently a Professor of computer science with the Centre for Quantum Computation and Information Systems and the Faculty of Engineering and Information Technology, University of Technology, Sydney, NSW, Australia. He mainly applies statistics and mathematics for data analysis problems in data mining, computer vision, machine learning, multimedia, and video surveillance. He has authored or coauthored more than 100 scientific articles.

Prof. Tao was the recipient of the runner-up award for Best Theory/Algorithm Award at IEEE ICDM-07.

Zhigang Luo received the B.S., M.S., and Ph.D. degrees from the National University of Defense Technology, Changsha, China, in 1981, 1993, and 2000, respectively.

He is currently a Professor with the School of Computer Science, National University of Defense Technology. His current research interests include parallel computing, computer simulation, and bioinformatics.

Bo Yuan received the Bachelor degree from Peking University Medical School, Peking, China, in 1983, and the M.S. degree in biochemistry and the Ph.D. in molecular genetics both from the University of Louisville, Kentucky, in 1990 and 1995, respectively.

He is currently a Professor with the Department of Computer Science and Engineering with Shanghai Jiao Tong University (SJTU), Shanghai, China. He was a Tenure-Track Assistant Professor with the Ohio State University (OSU), while he was a Codirector for the OSU Program in Pharmacogenomics. At the OSU, he was the Founding Director for the OSU’s Genome Initiative during the early 2000s, leading one of the only three independent efforts in the world (besides the Human Genome Project and the Celera company), having assembled and deciphered the entire human and mouse genomes. His current research interests include biological networks, network evolution, stochastic process, biologically inspired computing, and bioinformatics, particularly on how these frameworks might impact the development of intelligent algorithms and systems.