Non-Negative Graph Embedding

Jianchao Yang[1], Shuicheng Yang[2], Yun Fu[1], Xuelong Li[3], Thomas Huang[1]
[1]ECE Department, University of Illinois at Urbana-Champaign, USA
[2]ECE Department, National University of Singapore, Singapore
[3]School of CSIS, University of London, UK

Abstract

We introduce a general formulation, called non-negative graph embedding, for non-negative data decomposition by integrating the characteristics of both intrinsic and penalty graphs [17]. In the past, such a decomposition was obtained mostly in an unsupervised manner, such as Non-negative Matrix Factorization (NMF) and its variants, and hence unnecessary to be powerful at classification. In this work, the non-negative data decomposition is studied in a unified way applicable for both unsupervised and supervised/semi-supervised configurations. The ultimate data decomposition is separated into two parts, which separatively preserve the similarities measured by the intrinsic and penalty graphs, and together minimize the data reconstruction error. An iterative procedure is derived for such a purpose, and the algorithmic non-negativity is guaranteed by the non-negative property of the inverse of any M*-matrix. Extensive experiments compared with NMF and conventional solutions for graph embedding demonstrate the algorithmic properties in sparsity, classification power, and robustness to image occlusions.*

1. Introduction

Techniques for non-negative and sparse representation have been well studied in recent years to find non-negative basis with few nonzero elements [5]. Non-negative Matrix Factorization (NMF) [9] is the pioneering work for such a purpose. It imposes non-negativity constraints in data reconstruction, and requires that the elements of the projection vectors, *i.e.*, bases, together with the low-dimensional representations, are all non-negative. This ensures that the basis vectors shall be combined to form an image in a non-subtractive way.

Li *et al.* [11] imposed extra constraints to reinforce the basis sparsity of NMF; also matrix-based NMF has been extended to Non-negative Tensor Factorization (NTF) [4] for handling the data encoded as high-order tensors. Wang *et al.* proposed the Fisher-NMF [16], which was further studied by Kotsia *et al.* [8], by adding an extra term of scatter difference to the objective function of NMF. Tao *et al.* [13] proposed to employ local rectangle binary features for image reconstruction. Recently, Yan *et al.* [18] proposed a supervised algorithm for learning semantic localized patterns with binary projections, and each non-overlapping subset of features constitute a binary projection vector for feature extraction.

There is psychological and physiological evidence for parts-based representations in the brain [9]. But most previous algorithms for non-negative data decomposition are unsupervised and motivated for data reconstruction. Hence, they are still far from explaining the functions of the brain, which shows great capability in learning and classification. There were some attempts [16][8] to search for non-negative representation in a supervised manner, but their solution was derived by directly combining the objective functions for NMF and the specific Maximum Margin Criterion [10], which are essentially contrary to each other, hence the solution is not the best for either data reconstruction or classification. Although varieties of dimensionality reduction algorithms [2][3][6] have been proposed in pattern recognition, Yan *et al.* claimed that most of them, such as Principal Component Analysis (PCA) [7][14], LDA [1][12] and the recently proposed Marginal Fisher Analysis (MFA) [17], can be unified with a general formulation, called graph embedding [17]. A natural question is whether there exists a unified formulation to search for non-negative data decomposition by following the criteria of different dimensionality reduction algorithms. We give a positive answer to this question in this work.

We present a general formulation, called non-negative graph embedding, such that all the algorithms unified within the graph embedding framework can be easily extended to obtain their non-negative solutions. First, the ultimate data decomposition is divided into two parts. One part preserves the characteristics of the intrinsic graph, which describes the favorite similarities of the data pairs, and the other retains the characteristics of the penalty graph, which describes the unfavored similarities for the specific targets of a certain algorithm. On the other hand, these two parts jointly target the best reconstruction of the original data. Then, an iterative procedure is presented to search for such two parts, and the non-negativity of the solution is guaranteed by the theoretical fact that the inverse of any M-matrix is non-negative. Meanwhile, the auxiliary functions are used to prove the algorithmic convergence.

The remainder of this paper is organized as follows: In Section 2, we introduce the details of non-negative graph embedding, followed by algorithmic analysis in Section 3. Section 4 demonstrates the detailed experimental results, and we conclude this paper in Section 5.

2. Non-Negative Graph Embedding

For a classification problem, we assume that the training sample data are given as $X = [x_1, x_2, \dots, x_N]$, where $x_i \in \mathbb{R}^m$, N is the total number of training samples, and the corresponding class labels are denoted as $\{c_i | c_i \in \{1, ..., N_c\}\}_{i=1}^N$ with N_c is the class number. Denote the sample number of the cth class as n_c. Since in practice the feature dimension m is often very large, it is usually necessary to transform the original high-dimensional data into a low-dimensional feature space for facilitating and enhancing subsequent process. In this work, we address the problem of general non-negative data decomposition, applicable for both unsupervised and supervised/semi-supervised configurations. Note that we utilize in this work the following rule to facilitate presentation: for any matrix A, A_i means the ith row vector of A, its corresponding lowercase version a_i means the ith column vector of A, and A_{ij} denotes the element of A at the ith row and jth column.

2.1. Motivations

Non-Negative Matrix Factorization (NMF) factorizes the data matrix into one lower-rank *non-negative* basis matrix and one *non-negative* coefficient matrix. Its objective function is,

$$\min_{W,H} \|X - WH\|, \;\; s.t. \;\; W, H \geq 0, \tag{1}$$

where $W \in \mathbb{R}^{m \times r}$ is the basis matrix and $H \in \mathbb{R}^{r \times N}$ is the coefficient matrix. Usually, $r¡min(m, N)$ for dimensionality reduction. NMF is unsupervised and has been proposed for data reconstruction, and hence the derived coefficient matrix is unnecessary to be great at classification capability. Classification is generally the ultimate target of feature extraction, and hence it is desirable to derive non-negative solution for data decomposition in supervised and semi-supervised configurations.

Despite the different motivations of different algorithms for dimensionality reduction, Yan *et al.* [17] claimed that most of them can be explained within a unified framework, called graph embedding. Let $G = \{X, S\}$ be an undirected weighted graph with vertex set X and similarity matrix $S \in \mathbb{R}^{N \times N}$. Each element of the real symmetric matrix S measures for a pair of vertices the similarity, which is assumed to be non-negative in this work. The diagonal matrix D and the Laplacian matrix L of a graph G are defined as,

$$L = D - S, \;\; D_{ii} = \sum_{j \neq i} S_{ij}, \;\;\; \forall\, i. \tag{2}$$

Graph embedding generally involves an intrinsic graph G, which characterizes the favorite relationship among the training data, and a penalty graph $G^p = \{X, S^p\}$, which characterizes the unfavorable relationship among the training data, with $L^p = D^p - S^p$, where D^p is the diagonal matrix as defined in (2), and then two targets of graph-preserving are given as follows,

$$\begin{cases} \max_{H^1} \sum_{i \neq j} \|h_i^1 - h_j^1\|^2 S_{ij}^p, \\ \min_{H^1} \sum_{i \neq j} \|h_i^1 - h_j^1\|^2 S_{ij}, \end{cases} \tag{3}$$

where $H^1 = [h_1^1, h_2^1, \cdots, h_N^1] \in \mathbb{R}^{d \times N}$ are the desired low-dimensional representations for the training data.

Although several procedures [15] have been proposed to tackle this problem, these solutions are not guaranteed to be non-negative. Motivated by the non-negativity of the solution and the algorithmic commonness unified as graph embedding, we study in this work a general framework, called non-negative graph embedding, such that all the dimensionality reduction algorithms unified by graph embedding can easily yield their non-negative solutions by following this new formulation, and consequently the non-negative data decomposition can be conducted in a supervised or semi-supervised configuration.

2.2. Problem Formulation

To formulate the non-negative graph embedding, we follow hereafter all the notations in Section 2.1. To make the notations of NMF and graph embedding consistent, we divide the coefficient matrix H into two parts, namely,

$$H = \begin{bmatrix} H^1 \\ H^2 \end{bmatrix}, \tag{4}$$

where H^1 is defined the same as in Eqn. (3), and $H^2 = [h_1^2, h_2^2, \cdots, h_N^2] \in \mathbb{R}^{(r-d) \times N}$, where $d < r$. Correspondingly the basis matrix W is also divided into two parts,

$$W = [W^1 \; W^2], \tag{5}$$

where $W^1 \in \mathbb{R}^{m \times d}$ and $W^2 \in \mathbb{R}^{m \times (r-d)}$.

The desired low-dimensional representation H^1 of the training data aims to retain properties of the intrinsic graph and at the same time avoid the properties of the penalty graph, and hence is unnecessary to be sufficient at reconstructing the original data. Here, (W^2, H^2) are considered as the complementary space of (W^1, H^1), and they together reconstruct the original data in an additive manner. As stated in (3), there exist two objectives for graph embedding. From the complementary property between H^1 and H^2, the first objective in (3) is transformed into,

$$\min_{H^2} \sum_{i \neq j} \|h_i^2 - h_j^2\|^2 S_{ij}^p. \tag{6}$$

Then, we have the objective function for non-negative graph embedding as follows,

$$\min_{W,H} \sum_{i \neq j} \|h_i^1 - h_j^1\|^2 S_{ij} + \sum_{i \neq j} \|h_i^2 - h_j^2\|^2 S_{ij}^p \\ + \lambda \|X - WH\|^2, \;\; s.t. \;\; W, H \geq 0, \tag{7}$$

where λ is a positive parameter for balancing the two parts for graph embedding and data reconstruction. In this work, λ may be set as 1 because the two parts are at similar scale levels.

From the definitions in Eqn. (3), we have

$$\sum_{i \neq j} \|h_i^1 - h_j^1\|^2 S_{ij} = Tr(H^1 L {H^1}^T) \,, \tag{8}$$

$$\sum_{i \neq j} \|h_i^2 - h_j^2\|^2 S_{ij}^p = Tr(H^2 L^p {H^2}^T). \tag{9}$$

As stated in Section 2.1, W is the basis matrix and hence it is natural to require that the column vectors of W are normalized. But this constraint makes the optimization problem much more complex, and in this work, we compensate the norms of the bases into the coefficient matrix and rewrite the first two terms of (7) as

$$Tr(H^1 L {H^1}^T) \Rightarrow Tr(Q_1 H^1 L {H^1}^T Q_1^T), \tag{10}$$

$$Tr(H^2 L^p {H^2}^T) \Rightarrow Tr(Q_2 H^2 L^p {H^2}^T Q_2^T), \tag{11}$$

where the matrix $Q_1 = diag\{\|w_1^1\|, \|w_2^1\|, \cdots, \|w_d^1\|\}$ and $Q_2 = diag\{\|w_1^2\|, \|w_2^2\|, \cdots, \|w_{r-d}^2\|\}$, where w_i^k denotes the ith column vector of matrix W^k, $k = 1, 2$.

Then, finally we have the objective function,

$$\min_{W,H} Tr(Q_1 H^1 L {H^1}^T Q_1^T) + Tr(Q_2 H^2 L^p {H^2}^T Q_2^T) \\ + \lambda \|X - WH\|^2, \;\; s.t. \;\; W, H \geq 0. \tag{12}$$

This objective function is biquadratic, and generally there is no closed-form solution. We present in the next subsection an iterative procedure for computing the non-negative solution.

2.3. Convergent Iterative Procedure

Most iterative procedures for solving high-order optimization problems transform the original intractable problem into a set of tractable sub-problems, and finally obtain the convergence to a local optimum. Our proposed iterative procedure also follows this philosophy and optimizes H and W alternately.

2.3.1 Preliminaries

Before formally describing the iterative procedure for non-negative graph embedding, we first introduce two concepts: auxiliary function and M-matrix, and some lemmas which shall be used for the algorithmic deduction and convergence proof.

Definition-1 Function $G(A, A')$ is an auxiliary function for function $F(A)$ if the conditions

$$G(A, A') \geq F(A), \quad G(A, A) = F(A), \tag{13}$$

are satisfied.

Definition-2 A matrix B is called M-matrix if the conditions, 1) the off-diagonal entries are less than or equal to zeros, namely $B_{ij} \leq 0, i \neq j$; and 2) the real parts of all eigenvalues are positive.

From these two definitions, we have two lemmas with proofs omitted.

Lemma-1 If G is an auxiliary function, then F is non-increasing under the update

$$A^{t+1} = \arg\min_A G(A, A^t), \tag{14}$$

where t means the tth iteration.

Lemma-2 If B is an M-matrix, the inverse of the matrix B is non-negative, namely $B_{ij}^{-1} \geq 0, \forall\, i, j$.

2.3.2 Optimize W for given H

For a fixed H, the objective function in (12) with respect to the basis matrix W can be rewritten as

$$F(W) = Tr(W D^h W^T) + \lambda \|X - WH\|^2, \tag{15}$$

where the matrix $D^h = diag\{c_1^1, \cdots, c_d^1, c_1^2, \cdots, c_{r-d}^2\}$, the element $c_k^1 = \sum_{i \neq j} \|H_{ik}^1 - H_{jk}^1\|^2 S_{ij}$ and $c_k^2 = \sum_{i \neq j} \|H_{ik}^2 - H_{jk}^2\|^2 S_{ij}^p$.

From the objective function, we notice that different row vectors of W are independent to each other for optimization, and hence the objective function can be further simplified into row-wise form as

$$F(W_i) = W_i D^h W_i^T + \lambda \|X_i - W_i H\|^2, \tag{16}$$

where W_i is the ith row vector of W and X_i is the ith row vector of the data matrix X. Here, we denote the function $f(W_i) = \lambda\|X_i - W_i H\|^2$.

The auxiliary function of $F(W_i)$ is defined as

$$G(W_i, W_i^t) = W_i D^h W_i^T + f(W_i^t) + \triangledown f(W_i^t)(W_i - W_i^t)^T + \frac{1}{2}(W_i - W_i^t)K(W_i^t)(W_i - W_i^t)^T,$$

where $\triangledown f(W_i^t)$ is the gradient vector of $f(W_i)$ with respect to W_i at the point W_i^t. $K_{ij}(W_i^t) = \delta_{ij}(\lambda W_i^t H H^T)_j / W_{i,j}^t$ where δ_{ij} is an indicator function.

From the proof in [9], it is easy to prove that $G(W_i, W_i^t)$ is the auxiliary function of $F(W_i)$. Then, W_i^{t+1} can be computed by minimizing $G(W_i, W_i^t)$.

By setting $\frac{\partial G(W_i, W_i^t)}{\partial W_i} = 0$, we have

$$W_i^{t+1} = \lambda\, X_i\, H^T (K(W_i^t) + 2D^h)^{-1}. \quad (17)$$

It is obvious that the updated basis vector W_i^{t+1} is still non-negative if the matrices H and W_i^t are non-negative. After the updating of W^{t+1}, we normalize the column vectors of W^{t+1} and consequently convey the norm to the coefficient matrix, namely,

$$H_{ij} \Leftarrow H_{ij} \times \|w_i^{t+1}\|, \forall\, i, j, \quad (18)$$
$$w_j^{t+1} \Leftarrow w_j^{t+1} / \|w_j^{t+1}\|, \forall\, j, \quad (19)$$

where w_j^{t+1} is the jth column vector of the basis matrix W^{t+1}. Note that the updating of W and H in (18-19) will not change the value of the objective function in Eqn. (12).

2.3.3 Optimize H for given normalized W

Then based on the normalized W in Eqn. (19), the objective function in (12) with respect to H for given W can be written as

$$F(H) = Tr(H^1 L H^{1^T}) + Tr(H^2 L^p H^{2^T}) + \lambda\|X - WH\|^2,$$

and here we denote $f(H) = \lambda\|X - WH\|^2$.

The auxiliary function of $F(H)$ is designed as

$$G(H, H^t) = Tr(H^1 L H^{1^T}) + Tr(H^2 L^p H^{2^T}) + g(H, H^t),$$

where we have

$$g(H, H^t) = f(H^t) + \sum_{j=1}^{N} Tr(\triangledown f(h_j^t)^T (h_j - h_j^t)) + \sum_{j=1}^{N} \frac{1}{2}(h_j - h_j^t)^T K^j(h_j^t)(h_j - h_j^t), \quad (20)$$

where $K^j(h_j^t)_{ij} = \delta_{ij}(\lambda W^T W h_j^t)_i / H_{ij}^t$.

From the proof in [9], it is obvious that $g(H, H^t)$ is the auxiliary function of $f(H)$, and it is direct to conclude that $G(H, H^t)$ is the auxiliary function of $F(H)$. Then a refined H^{t+1} can be obtained by minimizing the objective function $G(H, H^t)$ with respect to H.

The gradient matrix of $G(H, H^t)$ with respect to H is

$$\frac{\partial G(H, H^t)}{\partial H} = \begin{bmatrix} 2H^1 L \\ 2H^2 L^p \end{bmatrix} + \triangledown f(H^t) + [K^1(h_1^t)(h_1 - h_1^t), \cdots, K^N(h_N^t)(h_N - h_N^t)]. \quad (21)$$

By setting $\frac{\partial G(H, H^t)}{\partial H} = 0$, we can derive H^{t+1} as follows. If $i \leq d$, then the ith row vector, denoted as H_i^{t+1}, of the coefficient matrix H^{t+1} can be updated by setting

$$\frac{\partial G(H, H^t)}{\partial H_i} = 2H_i L + \triangledown f(H_i^t) + (H_i - H_i^t)K^i = 0, \quad (22)$$

where the matrix $K^i = diag\{K^1(h_1^t)_{ii}, \cdots, K^N(h_N^t)_{ii}\}$. Then, we have

$$H_i = \lambda\, w_i^T\, X (K^i + 2L)^{-1}. \quad (23)$$

Similarly, if $d < i \leq r$, then H_i^{t+1} can be updated by setting

$$\frac{\partial G(H, H^t)}{\partial H_i} = 2H_i L^p + \triangledown f(H_i^t) + (H_i - H_i^t)K^i = 0, \quad (24)$$

and then we have

$$H_i = \lambda\, w_i^T\, X (K^i + 2L^p)^{-1}. \quad (25)$$

To prove the non-negativity of new H_i, we have the following theorems.

Theorem-1 The matrices $K^i + 2L$ and $K^i + 2L^p$ are both M-matrices.

Proof: From the definition of K^i, we have

$$K_{jj}^i = K^j(h_j^t)_{ii} = (\lambda W^T W h_j^t)_i / H_{ij}^t \geq \lambda w_i^T w_i H_{ij}^t / H_{ij}^t = \lambda\|w_i\|^2 > 0. \quad (26)$$

It means that all diagonal elements of the diagonal matrix K^i are positive. On the other hand, the matrix L and L^p are positive semidefinite, and the off-diagonal elements are not larger than zero according the definition in Eqn. (3). Then we can conclude that 1) the eigenvalues of the matrices $K^i + 2L$ and $K^i + 2L^p$ are all positive, and 2) the off-diagonal elements of the matrices $K^i + 2L$ and $K^i + 2L^p$ are not larger than zero, so they are both M-matrices according to the Definition-2. □

Since $K^i + 2L$ and $K^i + 2L^p$ are both M-matrices, and hence their inverses are both non-negative according to the

Lemma-2. Consequently we have the Theorem-2 as follows.

Theorem-2 The updated $H_i = \lambda w_i^T X(K^i + 2L)^{-1}$ and $H_i = \lambda w_i^T X(K^i + 2L^p)^{-1}$ are non-negative.

After we obtain the updated H^{t+1}, the objective function value can be further reduced by rearranging the row order of the matrix H^{t+1} along with the corresponding column rearrangement of the matrix W^{t+1}. Here, we rewrite the objective function (12) as

$$Tr(H^1(L - L^p)H^{1^T}) + Tr(HL^pH) + \lambda\|X - WH\|^2.$$

It shows that the rearrangement of H will not affect the second term and $\lambda\|X - WH\|^2$[1], and hence we can rearrange H such that the fist term is minimized. This purpose can be easily achieved by selecting the d rows of H with minimal values of $H_i^{t+1}(L - L^p)H_i^{t+1^T}$.

The two-step procedure can be summarized as in Algorithm 1. The matrices W and H are iteratively optimized until the the convergence criterion is attained (see step 2.5 in Algorithm-1).

3. Algorithmic Analysis

In the section, we prove the convergence of the iterative procedure.

Theorem-3 The iterative procedure listed in Algorithm 1 converges to a local optimum.

Proof: Here we define

$$F(W, H) = Tr(H^1 L H^{1^T}) + Tr(H^2 L^p H^{2^T}) + \lambda\|X - WH\|^2.$$

According to the updating rule for W, we have

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

On the other hand, according to the updating rule for H, we have

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

Thus we can conclude that

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

We also have $F(W^t, H^t) \geq 0$, then $F(W^t, H^t)$ decreases monotonically and has lower bound; hence $F(W^t, H^t)$ will converge to a local optimum. □

Algorithm 1 . Non-negative Graph Embedding

```
1: Initialize $W^0, H^0$ as arbitrary non-negative matrices.
2: For $t = 0, 1, 2, \ldots, T_{max}$, Do
     1. For given $H = H^t$, update the basis matrix $W$ as:
            $W_i^{t+1} = \lambda\, X_i\, H^T(K(W_i^t) + 2D^h)^{-1},$
        where $W_i^{t+1}$ is the $i$th row vector of $W^{t+1}$.
     2. Normalize the column vectors of $W^{t+1}$,
            $H_{ij} \Leftarrow H_{ij} \times \|w_i^{t+1}\|, \forall\, i, j,$   (27)
            $w_j^{t+1} \Leftarrow w_j^{t+1}/\|w_j^{t+1}\|, \forall\, j.$   (28)
     3. For given $W = W^{t+1}$, update the matrix $H$ as:
        If $i \leq d$,
            $H_i^{t+1} = \lambda w_i^T X(K^i + 2L)^{-1}.$
        Else
            $H_i^{t+1} = \lambda w_i^T X(K^i + 2L^p)^{-1},$
        where $H_i^{t+1}$ is the $i$th row vector of $H^{t+1}$.
     4. Rearrange the row order of $H^{t+1}$ according to the
        value of $H_i^{t+1}(L - L^p)H_i^{t+1^T}$.
     5. If $\|W^{t+1} - W^t\| < \sqrt{mr}\,\varepsilon$ and $\|H^{t+1} - H^t\| <$
        $\sqrt{Nr}\,\varepsilon$ ($\varepsilon$ is set to $10^{-4}$ in this work), then break.
3: Output $W = W^t$ and $H = H^t$.
```

4. Experiments

In this section, we evaluate the effectiveness of our proposed non-negative graph embedding framework compared with the popular subspace learning algorithms including Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Marginal Fisher Analysis (MFA), and other non-negative basis pursuit algrithms Non-negative Matrix Factorization (NMF) and Localized NMF (LNMF). For the NGE algorithm, the intrinsic graph and penalty graph are set the same as those for MFA, where the number of nearest neighbors of each sample is fixed as 3 (2 for FERET database) and the number of shortest pairs from different classes is set as 20 in this work.

4.1. Data Sets

In our experiments, we use three benchmark face databases CMU PIE, ORL, and FERET [2]. All images are

[1] The corresponding column order of W need also be rearranged.

[2] Available at http://www.face-rec.org/databases/.

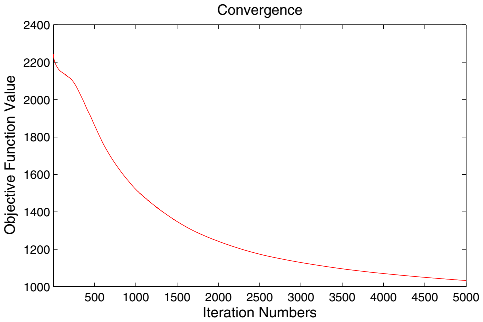

Figure 1. Objective function value decreases along with the increase of the iteration number.

aligned by fixing the locations of the two eyes. The CMU PIE (Pose, Illumination, and Expression) database contains more than 40,000 facial images of 68 people. In our experiment, a subset of five near frontal poses (C27, C05, C29, C09 and C07) and illuminations indexed as 08 and 11 is used, and therefore each person has ten images. For the FERET database, we use seventy people with six images for each person. The ORL database contains 40 persons, each with 10 images. For all the three databases, the images are normalized to 32-by-32 pixels, and half of the images from each person are randomly selected for modeling training and the other half for testing.

4.2. Algorithmic Convergence and Sparsity

In this subsection, we examine the convergence and sparsity properties of the NGE framework. As proved in the previous section, the update rules given in Algorithm 1 guarantees a local optimum solution for our objective function in Eqn. (7). In Fig. 1 we show how the objective function value decreases with increasing iterations on the FERET database. Our offline experiments show that generally NGE converges after about 5000 iterations.

For the aforementioned three databases, in which the faces are not strictly aligned, the original NMF algorithm has difficulty in finding sparse and localized bases. Li *et al.* [11] proposed the LNMF algorithm by adding an extra constraint term for reinforcing the sparsity. The basis matrices of NGE compared with those from NMF and LNMF on CMU PIE, ORL and FERET databases are depicted in Fig. 2-4, from which we can observe that: 1) the basis of LNMF are much more sparse than those of NMF; and 2) the basis of NGE are also very sparse, but its basis shows to be the combination of small regions instead of a small single region as in LNMF. The classification superiority of NGE over LNMF as introduced in the next subsection indicates that the discriminating information may be often characterized in a spatially contextual way.

4.3. Classification Capability

In this subsection, we evaluate the discriminating power of the non-negative algorithm NGE compared with the non-negative algorithms, NMF and LNMF, as well as the popular subspace learning algorithm PCA, LDA and MFA. The results from the original raw pixels without dimensionality reduction are taken as baselines. For LDA and MFA, we first conduct PCA to reduce the data to the dimension of $N - N_c$, where N is the number of training data and N_c is the number of classes, beforehand for avoiding the singular value issue as conventionally [1]. For all the non-negative algorithms NGE, NMF and LNMF, the parameter r is set as $N \times m/(N + m)$ in all the experiment settings, and d is

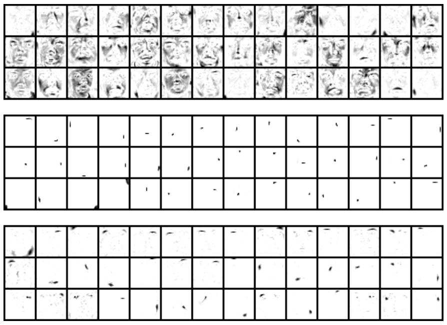

Figure 2. Basis matrices of the nonnegative algorithms NMF (1st row), LNMF (2nd row), and NGE (3rd row) based on the training data of the PIE database.

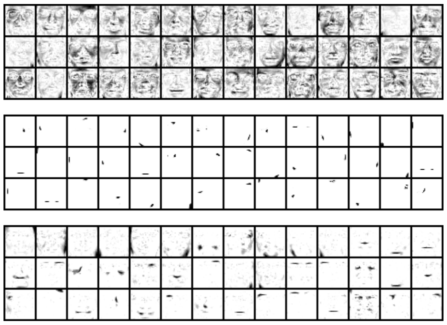

Figure 3. Basis matrices of the nonnegative algorithms NMF (1st row), LNMF (2nd row), and NGE (3rd row) based on the training data of the ORL database.

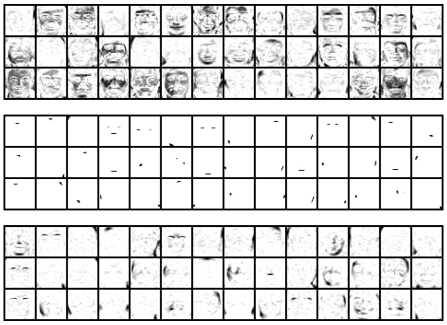

Figure 4. Basis matrices of the nonnegative algorithms NMF (1st row), LNMF (2nd row), and NGE (3rd row) based on the training data of the FERET database.

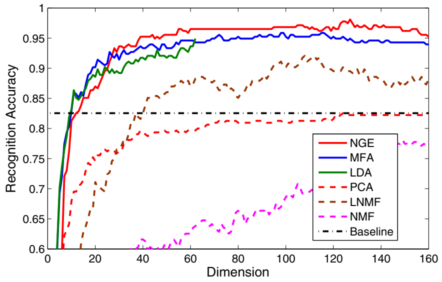

Figure 5. Face recognition accuracies over different feature dimensions for PCA, LDA, NMF, LNMF, MFA, and NGE algorithms on the PIE database. For better viewing, please see the color pdf file.

simply set to be N_c for NGE. For all these algorithms, we report the best results by exploring all possible feature dimensions as conventionally [17]. The comparison results on PIE, ORL and FERET databases are listed in Fig. 1. From these results, we can see that: 1) for the non-negative algorithms, LNMF performs on average much better than NMF, and NGE remarkably outperforms both NMF and LNMF on all the three databases; and 2) for the supervised subspace algorithms, NGE is comparable with or even better than MFA. On the PIE database, NGE significantly outperforms LDA and MFA, which is further validated by a more detailed comparison of face recognition accuracies over different feature dimensions as depicted in Fig. 5.

Table 1. Classification accuracies (%) of different algorithms on the three databases. Note that the number in parenthesis is the feature dimension with the best result.

Algorithm	PIE	ORL	FERET
Baseline	82.54	85.00	81.90
PCA	82.54 (124)	85.50 (105)	81.90 (141)
NMF	80.67 (208)	74.00 (158)	83.81 (174)
LNMF	92.06 (108)	87.50 (130)	81.90 (172)
NGE	**98.10 (127)**	**95.50 (121)**	**92.42 (152)**
LDA	94.92 (62)	94.50 (39)	91.60 (67)
MFA	95.87 (116)	95.50 (48)	91.43 (43)

4.4. Robustness to Image Occlusions

As showed in Fig. 2, 3 and 4, the discriminant bases from NGE are sparse and localized, which indicates that NGE is potentially more robust to image occlusions compared with algorithms with holistic bases such as PCA, LDA and MFA. To verify this point, we randomly add image occlusions of different sizes to the testing images. Several example faces from the PIE and ORL databases with occlusions are depicted in Fig. 6. The detailed recognition results of NGE compared with the above six algorithms are listed in Fig. 7. From all these results, the observation can be made that when the size of the occluded patch is less than 8-by-8 pixels, all these algorithms seem to be able to conquer the affect of image occlusions. However, when the occlusion size grows larger than 8-by-8 pixels, the performance of the algorithms with holistic basis *e.g.*, PCA, LDA, and MFA, will decrease quickly, while our algorithm NGE shows to be much more robust to the image occlusions, with improvements around 10 and 15 points separately over the second best algorithm for the occlusion sizes of 10-by-10 and 12-by-12 pixels. Another interesting observation is that NMF and LNMF are only comparable with PCA in all the settings [3]. One possible explanation for this may be that when the image size is small, the bases from PCA are much more robust to different affects, such as pose, illumination, and expression variations as well as image misalignments.

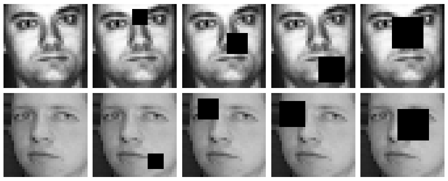

Figure 6. Sample images with occlusions from PIE (top row) and ORL (bottom row) databases. From left to right, the occlusion sizes are 0-by-0, 6-by-6, 8-by-8, 10-by-10, and 12-by-12 pixels respectively.

4.5. Discussions

In this subsection, we would like to discuss and highlight some aspects of our proposed NGE framework:

1. NGE is a general framework for non-negative data decomposition. It can be integrated with any subspace learning algorithm unified within the Graph Embedding framework as introduced in [17] to obtain (potentially sparse) non-negative discriminant basis. In this paper, we use MFA as a specific case, but the algorithm is ready to be extended to incorporate other graph embedding algorithms, unsupervised, supervised or semi-supervised, as long as **Theorem-1** is satisfied.

2. One similar work to NGE with LDA graph is the Fisher-NMF [16] algorithm, which is further refined in [8]. The experiments in [16] and [8] showed that Fisher-NMF indeed has improved classification power compared to the original unsupervised NMF and the later proposed LNMF [11]. But its performance is limited as reported in [8] that the face recognition performance of Fisher-NMF is inferior to LDA. Compared to Fisher-NMF, our NGE framework is superior in the following aspects: 1) in Fisher-NMF, the objective of NMF and that of Maximum Margin Criterion [10] are

[3] The results reported in [11] are based on faces of size 112-by-92 pixels, where NMF and LNMF can easily beat PCA.

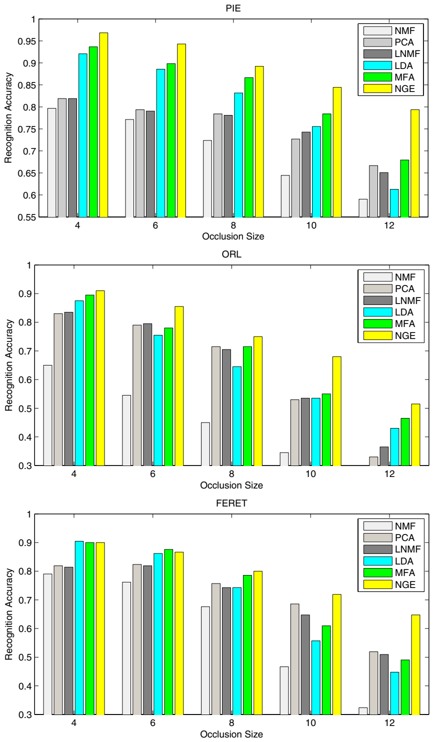

Figure 7. Recognition accuracy vs. Occlusion patch size. Top: results on the PIE face database. Middle: results on the ORL face database. Middle: results on the FERET face database. For better viewing, please see the color pdf file.

essentially contrary to each other, and hence the derived data decomposition is neither the best for data reconstruction nor the best for classification, while NGE provides a more reasonable mathematical formulation for these two purposes, and thus performs much better in classification; and 2) Fisher-NMF is designed by adopting the philosophy of LDA, while NGE is a general formulation.

3. All the experimental results validate the advantages of the marriage of basis non-negativity and supervised learning configuration, which inspires us to follow this integrated direction when develop new subspace learning techniques.

5. Conclusions and Future Works

In this paper, we proposed a general formulation for deriving non-negative solutions for all dimensionality reduction algorithms unified within the graph embedding framework. An iterative procedure was presented for such a purpose, and the inverse non-negativity of the *M*-matrix guarantees the non-negativity of the solution. The classification power and robustness to image occlusions are demonstrated by a specific case of our framework based on the MFA graphs. Further research on this topic includes: 1) currently the computational cost for the training of NGE is still relative high, more than two hours for a moderate size database. It will be useful to further improve algorithmic efficiency for larger size databases; 2) how to explicitly control the sparsity of the basis matrix while retaining the non-negativity of the updating rule; and 3) how to extend the current framework for tensor-based non-negative data decomposition.

Acknowledgement

This work is supported in part by U.S. government VACE Program, and in part by AcRF Tier-1 Grant of R-263-000-464-112/133, Singapore.

References

[1] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 711–720, 2002. 1, 6

[2] F. Chung. Spectral graph theory. *Regional Conferences Series in Mathematics*, 92, 1997. 1

[3] K. Fukunaga. Introduction to statistical pattern recognition. *Academic Press, second edition*, 1991. 1

[4] T. Hazan, S. Polak, and A. Shashua. Sparse image coding using a 3d non-negative tensor factorization. *ICCV*, vol. 1, pp. 50-57, 2005. 1

[5] C. Hu, B. Zhang, S. Yan, Q. Yang, J. Yan, Z. Chen, and Wei-Ying Ma. Mining Ratio Rules Via Principal Sparse Non-Negative Matrix Factorization. *ICDM*, pp. 407-410, 2004. 1

[6] A. Hyvärinen, J. Karhunen, and E. Oja. Independent component analysis. *John Wiley & Sons*, 2001. 1

[7] I. Joliffe. Principal component analysis. *Springer-Verlag, New York*, 1986. 1

[8] I. Kotsia, S. Zafeiriou, and I. Pitas. A Novel Discriminant Non-Negative Matrix Factorization Algorithm With Applications to Facial Image Characterization Problems. *IEEE Transactions on Information Forensics and Security*, pp. 588-595, 2007. 1, 7

[9] D. Lee and H. Seung. Learning the parts of objects by nonnegative matrix factorization. *Nature*, vol. 401, pp. 788–791, 1999. 1, 4

[10] H. Li, T. Jiang, and K. Zhang. Efficient and robust feature extraction by maximum margin criterion. *NIPS*, 2004. 1, 7

[11] S. Li, X. Hou, H. Zhang, and Q. Cheng. Learning spatially localized, parts-based representation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol.1, pp. 207–212, 2001. 1, 6, 7

[12] A. Martinez and A. Kak. Pca versus lda. *IEEE Transactions on Pattern Analysis and Ma-chine Intelligence*, vol. 23, no. 2, pp. 228–233, 2001. 1

[13] H. Tao, R. Crabb, and F. Tang. Non-orthogonal binary subspace and its applications in computer vision. *Proceedings of Computer Vision and Pattern Recognition Conference*, vol. 1, pp. 864–870, 2005. 1

[14] M. Turk and A. Pentland. Face recognition using eigenfaces. *IEEE Conference on Computer Vision and Pattern Recognition, Maui, Hawaii*, pp. 586–561, 1991. 1

[15] H. Wang, S. Yan, D. Xu, X. Tang, and T. Huang. Ratio vs. Ratio Trace for Dimensionality Reduction. *IEEE Conference on Computer Vision and Pattern Recognition*, pp.1–8, 2007. 2

[16] Y. Wang, Y. Jiar, C. Hu, and M. Turk. Fisher non-negative matrix factorization for learning local features. *Asian Conference on Computer Vision*, 2004. 1, 7

[17] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang and S. Lin. Graph Embedding and Extensions: A General Framework for Dimensionality Reduction. *Proc. IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 40-51, Jan. 2007. 1, 2, 7

[18] S. Yan, T. Yuan, and X. Tang. Learning Semantic Patterns with Discriminant Localized Binary Projections. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 168–174, 2006. 1