Graph Embedding and Extensions: A General Framework for Dimensionality Reduction

Shuicheng Yan, *Member*, *IEEE*, Dong Xu, Benyu Zhang, Hong-Jiang Zhang, *Fellow*, *IEEE*,
Qiang Yang, *Senior Member*, *IEEE*, and Stephen Lin

Abstract—Over the past few decades, a large family of algorithms—supervised or unsupervised; stemming from statistics or geometry theory—has been designed to provide different solutions to the problem of dimensionality reduction. Despite the different motivations of these algorithms, we present in this paper a general formulation known as graph embedding to unify them within a common framework. In graph embedding, each algorithm can be considered as the direct graph embedding or its linear/kernel/tensor extension of a specific intrinsic graph that describes certain desired statistical or geometric properties of a data set, with constraints from scale normalization or a penalty graph that characterizes a statistical or geometric property that should be avoided. Furthermore, the graph embedding framework can be used as a general platform for developing new dimensionality reduction algorithms. By utilizing this framework as a tool, we propose a new supervised dimensionality reduction algorithm called Marginal Fisher Analysis in which the intrinsic graph characterizes the intraclass compactness and connects each data point with its neighboring points of the same class, while the penalty graph connects the marginal points and characterizes the interclass separability. We show that MFA effectively overcomes the limitations of the traditional Linear Discriminant Analysis algorithm due to data distribution assumptions and available projection directions. Real face recognition experiments show the superiority of our proposed MFA in comparison to LDA, also for corresponding kernel and tensor extensions.

Index Terms—Dimensionality reduction, manifold learning, subspace learning, graph embedding framework.

1 Introduction

Techniques for dimensionality reduction [1] in supervised or unsupervised learning tasks have attracted much attention in computer vision and pattern recognition. Among them, the linear algorithms Principal Component Analysis (PCA) [11], [14], [22] and Linear Discriminant Analysis (LDA) [8], [14], [32], [33] have been the two most popular because of their relative simplicity and effectiveness. Another linear technique called Locality Preserving Projections (LPP) [10] has been proposed for dimensionality reduction that preserves local relationships within the data set and uncovers its essential manifold structure. For conducting nonlinear dimensionality reduction on a data set that lies on or around a lower dimensional manifold, ISOMAP [20], LLE [18], and Laplacian Eigenmap [3] are three algorithms that have recently been developed. In addition, the kernel trick [16] has also been widely applied to extend linear dimensionality reduction algorithms to nonlinear ones by performing linear operations on other higher or even infinite dimensional features transformed by a kernel mapping function. Recently, a number of algorithms [25], [26], [27], [29], [30], [31] have been proposed to conduct dimensionality reduction on objects encoded as matrices or tensors of arbitrary order.

In this paper, we present two linked innovations to dimensionality reduction. First, we present a general framework called graph embedding, along with its linearization, kernelization, and tensorization, that offers a unified view for understanding and explaining many of the popular dimensionality reduction algorithms such as the ones mentioned above. The purpose of direct graph embedding is to represent each vertex of a graph as a low-dimensional vector that preserves similarities between the vertex pairs, where similarity is measured by a graph similarity matrix that characterizes certain statistical or geometric properties of the data set. The vector representations of the vertices can be obtained from the eigenvectors corresponding to the leading eigenvalues of the graph Laplacian matrix with certain constraints. While direct graph embedding only presents the mappings for the graph vertices in the training set, its extensions provide mappings for all samples, such as new test data, in the original feature space. The linearization of graph embedding assumes that the vector representation of each vertex is linearly projected from the original feature vector representation of the graph vertex and the kernelization of graph embedding applies the kernel trick on the linear graph embedding algorithm to handle data with nonlinear distributions. Finally, in the tensorization of graph embedding, the original vertex is encoded as a general tensor of arbitrary order and the multilinear algebra approach is applied to extend the direct graph embedding to multilinear cases based on tensor representations. As we show in this paper, the above-mentioned algorithms, such as PCA, LDA, LPP,

- *S. Yan is with the Beckman Institute, University of Illinois at Urbana-Champaign, 405 North Mathews Avenue, Urbana, IL 61801. E-mail: scyan@ifp.uiuc.edu.*
- *D. Xu is with the Department of Electrical Engineering, Columbia University, 500 West 120th Street, New York, NY 10027. E-mail: dongxu@ee.columbia.edu.*
- *B. Zhang, H.-J. Zhang, and S. Lin are with Microsoft Research Asia, Beijing 100080, China. E-mail: {byzhang, hjzhang, stevelin}@microsoft.com.*
- *Q. Yang is with the Department of Computer Science, Hong Kong University of Science and Technology, Hong Kong. E-mail: qyang@cse.ust.hk.*

Manuscript received 3 Aug. 2005; revised 21 Feb. 2006; accepted 1 June 2006; published online 13 Nov. 2006.
Recommended for acceptance by A. Srivastava.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0415-0805.

ISOMAP, LLE, Laplacian Eigenmap, and the recently proposed tensor based algorithms, can all be reformulated within this common framework. In graph embedding, the underlying merits and shortcomings of different dimensionality reduction schemes, existing or new, are revealed by differences in the design of their intrinsic and penalty graphs and their types of embedding.

Our second contribution is to show that the graph embedding framework can be used as a general platform for developing new dimensionality reduction algorithms. We accomplish this task by designing graphs according to specific motivations. In particular, we will focus on formulating a variant of LDA using graph embedding. We observe that, despite the success of the LDA algorithm in many applications, its effectiveness is still limited since, in theory, the number of available projection directions is lower than the class number. Furthermore, class discrimination in LDA is based upon interclass and intraclass scatters, which is optimal only in cases where the data of each class is approximately Gaussian distributed, a property that cannot always be satisfied in real-world applications. While many efforts [32], [33], including the popular null subspace algorithm [24], have been devoted to improving the performance of LDA, the fundamental issues and limitations of LDA are still unsolved in theory.

Using the graph embedding framework as a platform, we develop a novel dimensionality reduction algorithm, Marginal Fisher Analysis (MFA), to overcome these limitations of LDA. In MFA, the intrinsic graph is designed to characterize intraclass compactness, and the penalty graph is formulated for interclass separability. In the intrinsic graph, a vertex pair is connected if one vertex is among the k_1-nearest neighbors of the other and the elements of the pair belong to the same class. In the penalty graph, for each class, the k_2-nearest vertex pairs in which one element is in-class and the other is out-of-class are connected. Based on the graph embedding framework, we develop MFA, Kernel MFA, and Tensor MFA to preserve the characteristics of the intrinsic graph and at the same time suppress the characteristics of the penalty graph. In comparison to LDA, MFA has the following advantages: 1) The number of available projection directions is much larger than that of LDA, 2) there is no assumption on the data distribution, thus it is more general for discriminant analysis, and 3) without a prior assumption on data distributions, the interclass margin can better characterize the separability of different classes than the interclass scatter in LDA.

The rest of the paper is structured as follows: We introduce in Section 2 the unified graph embedding formulation along with its linearization, kernelization, and tensorization for general dimensionality reduction. We then utilize the graph embedding framework as a general platform for dimensionality reduction to design Marginal Fisher Analysis along with its kernelization and tensorization in Section 3. We experimentally evaluate the proposed schemes in a series of face recognition experiments as well as a synthetic data experiment in Section 4. Finally, we give concluding remarks and a discussion of future work in Section 5.

2 Graph Embedding: A General Framework for Dimensionality Reduction

Many approaches have been proposed for the dimensionality reduction task. Although the motivations of all these algorithms vary, their objectives are similar, that is, to derive a lower dimensional representation and facilitate the subsequent classification task. A natural question that arises is whether they can be reformulated within a unifying framework and whether this framework assists the design of new algorithms. In this section, we give positive answers to these questions. We present the novel formulation of graph embedding along with its linearization, kernelization, and tensorization to provide a common perspective in understanding the relationship between these algorithms and to design new algorithms.

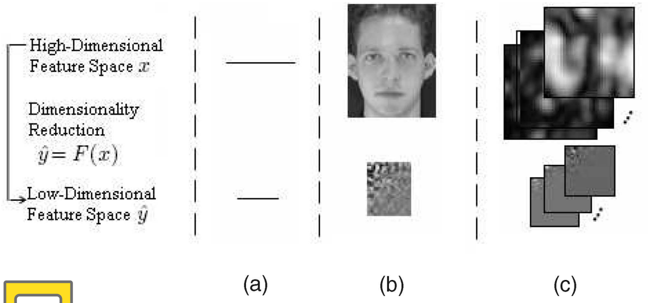

High-Dimensional Feature Space x
Dimensionality Reduction
$\hat{y} = F(x)$
Low-Dimensional Feature Space $\hat{y}$
(a)
(b)
(c)

Fig. 1. Illustration of dimensionality reduction for data of different forms. Note that the third-order tensor data are the Gabor filtered images.

2.1 Graph Embedding

For a general classification problem, the sample set for model training is represented as a matrix $X = [x_1, x_2, \ldots, x_N]$, $x_i \in \mathbb{R}^m$, where N is the sample number and m is the feature dimension. For supervised learning problems, the class label of the sample x_i is assumed to be $c_i \in \{1, 2, \ldots, N_c\}$, where N_c is the number of classes. We also let π_c and n_c denote the index set and number of the samples belonging to the cth class, respectively.

In practice, the feature dimension m is often very high and, thus, it is necessary and beneficial to transform the data from the original high-dimensional space to a low-dimensional one for alleviating the curse of dimensionality [8]. The essential task of dimensionality reduction is to find a mapping function $F : x \rightarrow \hat{y}$ that transforms $x \in \mathbb{R}^m$ into the desired low-dimensional representation $\hat{y} \in \mathbb{R}^{m'}$, where, typically, $m \gg m'$:

$$\hat{y} = F(x). \tag{1}$$

The function F may be explicit or implicit, linear, or nonlinear in different cases. An intuitive illustration of dimensionality reduction is displayed in Fig. 1 for different types of data, i.e., vectors, matrices, and the general tensors, as introduced later in this paper.

We now introduce the dimensionality reduction problem from the new point of view of graph embedding. Let $G = \{X, W\}$ be an undirected weighted graph with vertex set X and similarity matrix $W \in \mathbb{R}^{N \times N}$. Each element of the real symmetric matrix W measures, for a pair of vertices, its similarity, which may be negative. The matrix can be formed using various similarity criteria, such as Gaussian similarity from Euclidean distance as in [3], local neighborhood relationship as in [18], and prior class information in supervised learning algorithms as in [14]. The diagonal matrix D and the Laplacian matrix L of a graph G are defined as

$$L = D - W, \quad D_{ii} = \sum_{j \neq i} W_{ij}, \quad \forall\, i. \tag{2}$$

In this work, the graph embedding of the graph G is defined as an algorithm to find the desired low-dimensional vector representations relationships among the vertices of G that best characterize the similarity relationship between the vertex pairs in G. To simplify exposition, we explain the one-dimensional case and represent the low-dimensional representations of the vertices as a vector $y = [y_1, y_2, \dots, y_N]^T$, where y_i is the low-dimensional representation of vertex x_i. We define an intrinsic graph to be the graph G itself and a penalty graph $G^p = \{X, W^p\}$ as a graph whose vertices X are the same as those of G, but whose edge weight matrix W^p corresponds to the similarity characteristics that are to be suppressed in the dimension-reduced feature space. For a dimensionality reduction problem, we require an intrinsic graph G and, optionally, a penalty graph G^p as input. Our graph-preserving criterion is given as follows:

$$y^* = \arg \min_{y^T By=d} \sum_{i \neq j} \|y_i - y_j\|^2 W_{ij} = \arg \min_{y^T By=d} y^T L y, \tag{3}$$

where d is a constant and B is the constraint matrix defined to avoid a trivial solution of the objective function. B typically is a diagonal matrix for scale normalization and may also be the Laplacian matrix of a penalty graph G^p. That is, $B = L^p = D^p - W^p$, where D^p is the diagonal matrix as defined in (2). We note that a similar graph preserving criterion could alternatively be formulated with the constraint $\sum_i \|y_i\|^2 B_{ii} = d$ for scale normalization or $\sum_{i \neq j} \|y_i - y_j\|^2 W_{ij}^p = d$ for the penalty matrix G^p, when y_i is of multiple dimensions.

The similarity preservation property from the graph preserving criterion has a two-fold explanation. For larger (positive) similarity between samples x_i and x_j, the distance between y_i and y_j should be smaller to minimize the objective function. Likewise, smaller (negative) similarity between x_i and x_j should lead to larger distances between y_i and y_j for minimization.

The graph preserving criterion provides the direct graph embedding for all the vertices. To offer mappings for data points throughout the entire feature space, we present three approaches.

Linearization. Assuming that the low-dimensional vector representations of the vertices can be obtained from a linear projection as $y = X^T w$, where w is the unitary projection vector, the objective function in (3) becomes

$$w^* = \mathop{\arg\min}_{\substack{w^T XBX^T w=d \\ \text{or } w^T w=d}} \sum_{i \neq j} \|w^T x_i - w^T x_j\|^2 W_{ij} = \mathop{\arg\min}_{\substack{w^T XBX^T w=d \\ \text{or } w^T w=d}} w^T XLX^T w. \tag{4}$$

Note that, in the linearization case, scale normalization of the low-dimensional representations may be transformed onto the projection direction as in (4).

Commonly, the linearization extension of graph embedding is computationally efficient for both projection vector learning and final classification; however, its performance may degrade in cases with nonlinearly distributed data and we introduce the kernel extension of graph embedding to handle nonlinearly distributed data as follows.

Kernelization. A technique to extend methods for linear projections to nonlinear cases is to directly take advantage of the kernel trick [16]. The intuition of the kernel trick is to map the data from the original input space to another higher dimensional Hilbert space as $\phi : x \rightarrow \mathcal{F}$ and then perform the linear algorithm in this new feature space. This approach is well-suited to algorithms that only need to compute the inner product of data pairs $k(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$. Assuming that the mapping direction $w = \sum_i \alpha_i \phi(x_i)$ and K is the kernel Gram matrix with $K_{ij} = \phi(x_i) \cdot \phi(x_j)$, we have the following objective function from (4):

$$\alpha^* = \mathop{\arg\min}_{\substack{\alpha^T KBK\alpha=d \\ \text{or } \alpha^T K\alpha=d}} \sum_{i \neq j} \|\alpha^T K_i - \alpha^T K_j\|^2 W_{ij} = \mathop{\arg\min}_{\substack{\alpha^T KBK\alpha=d \\ \text{or } \alpha^T K\alpha=d}} \alpha^T KLK^T \alpha. \tag{5}$$

Here, K_i indicates the ith column vector of the kernel Gram matrix K.

The solutions of (3), (4), and (5) are obtained by solving the generalized eigenvalue decomposition problem [6],

$$\tilde{L} v = \lambda \tilde{B} v, \tag{6}$$

where $\tilde{L} = L$, XLX^T or KLK, and $\tilde{B} = I, B, K, XBX^T$, or KBK. For the problem in (3), there is a trivial solution with all elements being the same and corresponding to eigenvalue zero of the Laplacian matrix L. We generally omit it as in [3].

Tensorization. The above linearization and kernelization of graph embedding both consider a vector representation of vertices. However, the extracted feature from an object may contain higher-order structure. For example, an image is a second-order tensor, i.e., a matrix, and sequential data such as video sequences used in event analysis is in the form of a third-order tensor. In uncovering the underlying structure for data analysis, it is undesirable to mask the underlying high-order structure by transforming the input data into a vector as done in most algorithms, which often leads to the curse of dimensionality problem. Thus, a natural further extension of the above linearization and kernelization of graph embedding is to conduct dimensionality reduction with vertices encoded as general tensors of an arbitrary order.

Before introducing the tensorization of graph embedding, we review some terminology on tensor operations [23]. The inner product of two tensors $\mathbf{A} \in \mathbb{R}^{m_1 \times m_2 \times \dots \times m_n}$ and $\mathbf{B} \in \mathbb{R}^{m_1 \times m_2 \times \dots \times m_n}$ of the same dimensions is defined as

$$< \mathbf{A}, \mathbf{B} >= \sum_{i_1=1,\dots,i_n=1}^{i_1=m_1,\dots,i_n=m_n} \mathbf{A}_{i_1,\dots,i_n} \mathbf{B}_{i_1,\dots,i_n},$$

the norm of a tensor $\mathbf{A}$ is $\|A\| = \sqrt{< \mathbf{A}, \mathbf{A} >}$, and the distance between tensors $\mathbf{A}$ and $\mathbf{B}$ is $\|\mathbf{A} - \mathbf{B}\|$. In the second-order tensor case, the norm is called the Frobenius norm and written as $\|\mathbf{A}\|_F$. The k-mode product of a tensor $\mathbf{A}$ and a matrix $U \in \mathbb{R}^{m_k \times m_k'}$ is defined as $\mathbf{B} = \mathbf{A} \times_k U$, where $\mathbf{B}_{i_1,\dots,i_{k-1},j,i_{k+1},\dots,i_n} = \sum_{I=1}^{m_k} \mathbf{A}_{i_1,\dots,i_{k-1},i,i_{k+1},\dots,i_n} \times U_{ij}, j = 1, \dots, m_k'$. In this paper, bold upper case letters represent general tensors, italic upper case letters denote matrices, italic lower case letters represent vectors, and plain-text lower case letters denote scalars.

We express the training sample set in tensor form as $\{\mathbf{X_i} \in \mathbb{R}^{m_1 \times m_2 \times \dots \times m_n}, i = 1, 2, \dots, N\}$. Similar to the linearization of graph embedding, we assume that the low dimensional representation of a vertex is a tensor of a smaller size which is projected from the original tensor with projection matrices. A one-dimensional case can be represented as

$$y_i = \mathbf{X}_i \times_1 w^1 \times_2 w^2 \dots \times_n w^n. \tag{7}$$

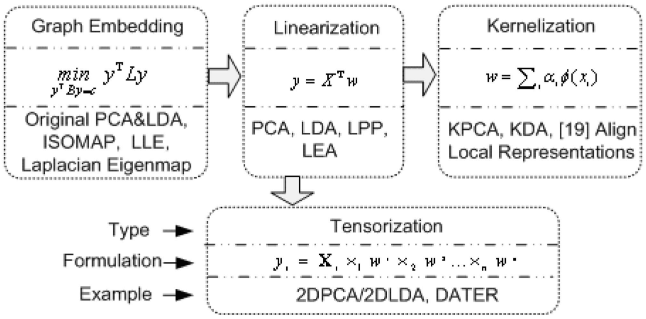

Graph Embedding
$\min_{y^T By = c} y^T L y$
Original PCA&LDA, ISOMAP, LLE, Laplacian Eigenmap
Linearization
$y = X^T w$
PCA, LDA, LPP, LEA
Kernelization
$w = \sum_i \alpha_i \phi(x_i)$
KPCA, KDA, [19] Align Local Representations
Type
Formulation
Example
Tensorization
$y_i = \mathbf{X}_i \times_1 w^1 \times_2 w^2 \ldots \times_n w^n$
2DPCA/2DLDA, DATER

Fig. 2. Graph embedding along with its linearization, kernelization, and tensorization: A unified framework for dimensionality reduction. The top row is the graph embedding type, the middle row is the corresponding objective function, and the third row lists the sample algorithms.

Then, the objective function of (3) is expressed as

$$(w^1, \ldots, w^n)^* = \underset{f(w^1,\ldots,w^n)=d}{\arg\min} \sum_{i \neq j} \|\mathbf{X_i} \times_1 w^1 \times_2 w^2 \ldots \times_n w^n - \mathbf{X_j} \times_1 w^1 \times_2 w^2 \ldots \times_n w^n\|^2 W_{ij}. \tag{8}$$

Here, if the matrix B is computed from scale normalization, then

$$f(w^1, \ldots, w^n) = \sum_{i=1}^{n} \|\mathbf{X_i} \times_1 w^1 \times_2 w^2 \ldots \times_n w^n\|^2 B_{ii}, \tag{9}$$

and, if B comes from the penalty graph, i.e., $B = L^p = D^p - W^p$, then

$$f(w^1, \ldots, w^n) = \sum_{i \neq j} \|\mathbf{X_i} \times_1 w^1 \times_2 w^2 \ldots \times_n w^n - \mathbf{X_j} \times_1 w^1 \times_2 w^2 \ldots \times_n w^n\|^2 W^p_{ij}. \tag{10}$$

In many cases, there is no closed-form solution for the objective function of (8). However, for each projection vector w_o, $o = 1, 2, \ldots, n$, if $(w^1, \ldots, w^{o-1}, w^{o+1}, \ldots, w^n)$ are known, then the objective function is the same as that of (4) if we set $x_i = \mathbf{X}_i \times_1 w^1 \ldots \times_{o-1} w^{o-1} \times_{o+1} w^{o+1} \ldots \times_n w^n$. Therefore, we can obtain the solution in a closed-form manner by fixing the other projection vectors and the local optimum of the objective function (8) can be obtained by optimizing different projection vectors iteratively.

Compared with the linearization of graph embedding, the feature dimension considered in each iteration of tensorization is much smaller which effectively avoids the curse of dimensionality issue and leads to a significant reduction in computational cost.

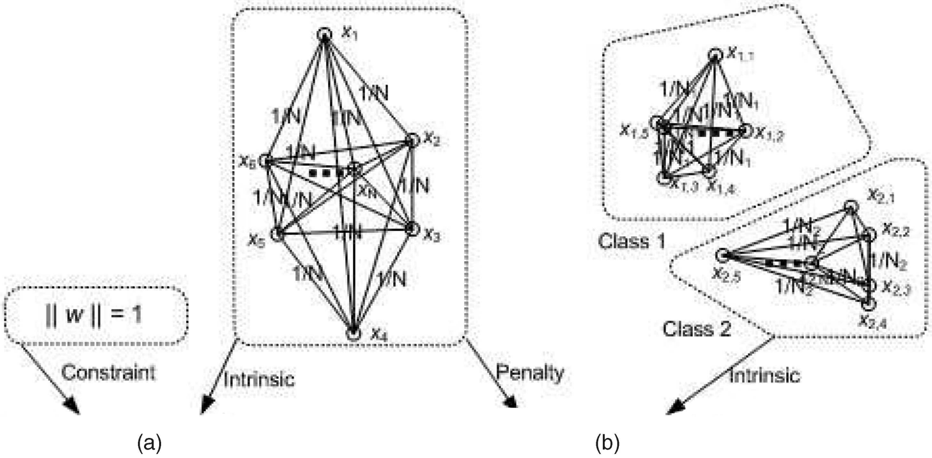
2.2 General Framework for Dimensionality Reduction

In this section, we show that the previously mentioned dimensionality reduction algorithms can be reformulated within the presented graph embedding framework. The differences between these algorithms lie in the computation of the similarity matrix of the graph and the selection of the constraint matrix. Fig. 2 provides an illustration of the graph embedding framework and also lists example algorithms for different types of graph embeddings. In the following, we give an overview of these algorithms.

PCA [11], [22] seeks projection directions with maximal variances. In other words, it finds and removes the projection direction with minimal variance, i.e.,

$$w^* = \underset{w^T w = 1}{\arg\min}\, w^T C w \qquad \text{with}$$
$$C = \frac{1}{N} \sum\nolimits_{i=1}^{N} (x_i - \bar{x})(x_i - \bar{x})^T = \frac{1}{N} X \left(I - \frac{1}{N} e e^T \right) X^T. \tag{11}$$

Here, e is an N-dimensional vector and I is an identity matrix, C is the covariance matrix, and $\bar{x}$ is the mean of all samples. It is clear that PCA follows the linearization of graph embedding with the intrinsic graph connecting all the data pairs with equal weights and constrained by scale normalization on the projection vector. Fig. 3a illustrates the intrinsic graph of PCA. KPCA [16] applies the kernel trick on PCA, hence it is a kernelization of graph embedding. 2DPCA [29] is a simplified second-order tensorization of PCA and only optimizes one projection direction, while [30] and [25] are full formulations of the second-order tensorization of PCA. Note that [10] qualitatively notes that PCA can be related to LPP by connecting the graph as in Fig. 3a. However, it does not completely and formally justify that PCA is a special case of their LPP framework since PCA utilizes a maximization criterion, while LPP is based on



(a) (b)

Fig. 3. The adjacency graphs for PCA and LDA. (a) Constraint and intrinsic graph in PCA. (b) Penalty and intrinsic graphs in LDA.

TABLE 1
The Common Graph Embedding View for the Most Popular Dimensionality Reduction Algorithms

Algorithm	W&B Definition	Type
PCA/KPCA/2DPCA	$W_{ij} = 1/N, i \neq j; B = I$	L/K/T
LDA/KDA/2DLDA/DATER	$W_{ij} = \delta_{c_i,c_j}/n_{c_i}; B = I - 1/Nee^T$	L/K/T/T
ISOMAP	$W_{ij} = \tau(D_G)_{ij}; B = I$	D
LLE/LEA/work [21]	$W = M + M^T + M^T M; B = I$	D/L/K
Laplacian Eigenmap/LPP	$W_{ij} = exp\{-\|x_i - x_j\|^2/t\}$, if $i \in N_k(j)\text{or}j \in N_k(i); B = D$	D/L

Note that the type D stands for direct graph embedding, while L, K, and T indicate the linearization, kernelization, and tensorization of the graph embedding, respectively.

minimization. With graph embedding, the intrinsic graph characterizes the properties of the projections that need to be be found and discarded, namely, the directions of small variance in PCA.

LDA [14] searches for the directions that are most effective for discrimination by minimizing the ratio between the intraclass and interclass scatters:

$$w^* = \arg \min_{w^T S_B w = d} w^T S_W w = \arg \min_w \frac{w^T S_W w}{w^T S_B w} = \arg \min_w \frac{w^T S_W w}{w^T C w},$$
$$S_W = \sum_{i=1}^{N} (x_i - \bar{x}^{c_i})(x_i - \bar{x}^{c_i})^T = X\left(I - \sum_{c=1}^{N_c} \frac{1}{n_c} e^c e^{cT}\right) X^T,$$
$$S_B = \sum_{c=1}^{N_c} n_c (\bar{x}^c - \bar{x})(\bar{x}^c - \bar{x})^T = NC - S_W. \tag{12}$$

Here, $\bar{x}^c$ is the mean of the cth class and e^c is an N-dimensional vector with $e^c(i) = 1$ if $c = c_i$; 0 otherwise. Note that, for the first line of (12), the second equality is guaranteed to be satisfied when $d \neq 0$. When $d = 0$, it will still be satisfied given that it is valid to minimize $w^T S_W w/d$ with respect to w and the optimal solution is obtained by minimizing $w^T S_W w$.

We can observe that LDA follows the linearization of graph embedding in which the intrinsic graph connects all the pairs with same class labels and the weights are in inverse proportion to the sample size of the corresponding class. The intrinsic graph of PCA is used as the penalty graph of LDA. Note that, although [10] discusses the relationship between LDA and LPP, $B = D$ in LPP, which implies that LDA cannot be described as a special case of the LPP algorithm. In contrast, with the graph embedding formulation in (4) and the constraint from a penalty graph, LDA can naturally be reformulated within the graph embedding framework. Fig. 3b exhibits these two graphs for LDA. The Kernel Discriminant Analysis (KDA) [9] algorithm is the kernel extension of LDA. 2DLDA [31] is the second-order tensorization of LDA, and the algorithm known as DATER [26] is the tensorization of LDA in arbitrary order.

ISOMAP was proposed in [20] to find the low-dimensional representations for a data set by approximately preserving the geodesic distances of the data pairs. Let D_G be the obtained approximated geodesic distance matrix. The function $\tau(D_G) = -HSH/2$, where $H = I - 1/Nee^T$ and $S_{ij} = D_G^2(i,j)$, converts the distance matrix into the corresponding inner product matrix. The MDS [20] algorithm is designed to obtain low-dimensional representations for all data points. ISOMAP follows the direct graph embedding formulation, as proven in Appendix A.

LLE [18] maps the input data to a lower dimensional space in a manner that preserves the relationship between the neighboring points. First, the sparse local reconstruction coefficient matrix M is calculated such that $\sum_{j \in N_k(i)} M_{ij} = 1$, where the set $N_k(i)$ is the index set of the k nearest neighbors of the sample x_i and the objective function $\|x_i - \sum_{j \in N_k(i)} M_{ij} x_j\|$ is minimized, and then the low dimensional representation y is obtained by minimizing $\sum_i \|y_i - \sum_{j \in N_k(i)} M_{ij} y_j\|^2$. LLE follows the direct graph embedding formulation, which we prove in Appendix B. Teh and Roweis [21] proposed a procedure to align disparate locally linear representations into a globally coherent coordinate system by preserving the relationship between neighboring points as in LLE. As demonstrated in [28], it is actually a special Geometry-Adaptive-Kernel-based LLE, that is, it is a kernel extension of LLE. The linearization of LLE, called LEA, was recently discussed in [7].

Laplacian Eigenmap (LE) [3] preserves the similarities of the neighboring points. Its objective function is similar to that in (3) and the adjacency matrix is calculated from the Gaussian function $W_{ij} = exp\{-\|x_i - x_j\|^2/t\}$ if $i \in N_k(j)$ or $j \in N_k(i)$; 0 otherwise. It naturally follows the direct graph embedding formulation. The newly proposed LPP [10] algorithm is its direct linear approximation, i.e., its linearization.

The above algorithms were proposed with different motivations. However, they in fact share the common formulation of (3), (4), (5), and (8). From the above analysis and the proofs given in the Appendices, Table 1 lists the similarity and constraint matrices for all of the above-mentioned methods. Their corresponding graph embedding types are also given.

2.3 Related Works and Discussions

We present a detailed discussion on the relationship between the graph embedding framework and some well-known related works.

2.3.1 Kernel Interpretation [13] and Out-of-Sample Extension [4]

Ham et al. [13] proposed a kernel interpretation of KPCA, ISOMAP, LLE, and Laplacian Eigenmap and demonstrated that they share a common KPCA formulation with different kernel definitions. Our framework and Ham's work present two intrinsically different perspectives to interpret these algorithms in a unified framework and they are different in many aspects. First, Ham's work was proposed by considering the normalized similarity matrix of a graph as a kernel matrix, whereas our work discusses the Laplacian matrix derived from the similarity matrix of a graph. Second, Ham et al.'s work analyzes only unsupervised learning algorithms, while our proposed framework can more generally analyze both unsupervised and supervised learning algorithms as described above. Moreover, as described later, our proposed framework can be utilized as a general platform and tool for developing new algorithms for dimensionality reduction. Bengio et al. [4] presented a method for computing the low dimensional representation of out-of-sample data. This method is also based on the kernel interpretation of ISOMAP, LLE, and Laplacian Eigenmap, and, similar to Ham's work, it cannot be directly generalized for supervised dimensionality reduction algorithms such as LDA.

2.3.2 Brand's Work [5]

Brand [5] also mentioned the concept of graph embedding, which defines an optimization problem as

$$y^* = \arg \max_{y^T Dy=1} y^T W y,$$

where the matrices W and D are defined as in (2). This optimization problem is equivalent to

$$y^* = \arg \min_{y^T Dy=1} y^T (D - W) y.$$

Brand's work is different from the graph embedding framework proposed in this work in several aspects. First, in [5], the constraint matrix B is fixed as the diagonal matrix D with $D_{ii} = \sum_j W_{ij}$ as in (2); hence, it is a special case of the general graph embedding framework. Despite the sound mathematical justification for this definition of the constraint matrix, it does not provide the level of generality and flexibility as our proposed framework in helping to understand and advance the dimensionality reduction literature; in our proposed graph embedding framework, the constraint matrix is flexible and can be defined in different ways: One is for scale normalization of which a special case is D as in [5] and another is the Laplacian matrix from a penalty graph as described above.

Second, our proposed graph embedding framework unifies most popular dimensionality reduction algorithms, while the proposed graph embedding in [5] does not cover many important algorithms in which the constraint matrix B is computed from the penalty matrix, such as in LDA and its variants. Moreover, [5] does not encompass the algorithms ISOMAP and LLE since the constraint matrix B in both methods is set as an identity matrix I, instead of D. We can also prove that the matrix D in ISOMAP and LLE are not equal to the matrix I, which can be seen from the proofs in Appendices A and B.

Finally, besides the basic direct graph embedding, we comprehensively extend our work to include linearization, kernelization, and tensorization. Brand's work [5] can be considered as a special case of our graph embedding framework.

2.3.3 Laplacian Eigenmap [3] and LPP [10]

Laplacian Eigenmap aims to find a low-dimensional representation that preserves the local properties of the data lying on a low-dimensional manifold. Drawing on the correspondence between the graph Laplacian, the Laplace Beltrami operator on the manifold, and the connections to the heat equation, it also imposes the optimization problem

$$y^* = \arg \min_{y^T Dy=1} y^T (D - W) y.$$

The entries in the similarity matrix W are defined as $W_{ij} = exp\{-\|x_i - x_j\|^2/t\}$ if one element of vertex pair (x_i, x_j) is among the k nearest neighbors of the other element; the entry is set to 0 otherwise. As described in Section 2.2, LPP is the linear approximation of Laplacian Eigenmap.

Similarly to Brand's work [5], the Laplacian Eigenmap and LPP algorithms are essentially different from our work. First, similarly to [5], the works of [3] and [10] assume that $B = D$; hence, they work with only a single graph, i.e., the intrinsic graph, and cannot be used to explain algorithms such as ISOMAP, LLE, and LDA.

Second, the works of [3] and [10] use a Gaussian function to compute the nonnegative similarity matrix, while, in our work, the similarity matrix is flexible, allowing the elements to even be negative.

Finally, as previously discussed, the works of [3] and [10] are special cases of our proposed graph embedding framework. Although [10] attempts to use LPP to explain PCA and LDA, this explanation is incomplete. The constraint matrix B is fixed to D in LPP, while as described in (12), the constraint matrix of LDA is not diagonal and comes from a penalty graph that connects all samples with equal weights; hence, LDA cannot be explained by LPP. Also, as discussed in Section 2.2, LPP, being a minimization algorithm, does not explain why PCA maximizes the objective function.

3 Marginal Fisher Analysis

In addition to encompassing most popular dimensionality reduction algorithms, the proposed framework can also be used as a general platform to design new algorithms for dimensionality reduction. The straightforward byproducts of the preceding analysis are the linear, kernel, and tensor extensions of the ISOMAP, LLE, and Laplacian Eigenmap algorithms. With this framework, we develop a new dimensionality reduction algorithm to avoid certain limitations of the traditional Linear Discriminant Analysis in terms of the data distribution assumption and available projection directions.

3.1 Marginal Fisher Analysis

The linear discriminant analysis algorithm is developed with the assumption that the data of each class is of a Gaussian distribution, a property that often does not exist in real-world problems. Without this property, separability of the different classes cannot be well characterized by interclass scatter. This limitation of LDA may be overcome by developing new criteria that characterizes intraclass compactness and interclass separability. Toward this end, we propose a novel algorithm, called Marginal Fisher Analysis (MFA), using the graph embedding framework. We design an intrinsic graph that characterizes the intraclass compactness and another penalty graph which characterizes the interclass separability, both shown in Fig. 4. In this figure, the intrinsic graph illustrates the intraclass point adjacency relationship, and each sample is connected to its k_1-nearest neighbors of the same class. The penalty graph illustrates the interclass marginal point adjacency relationship and the marginal point pairs of different classes are connected.

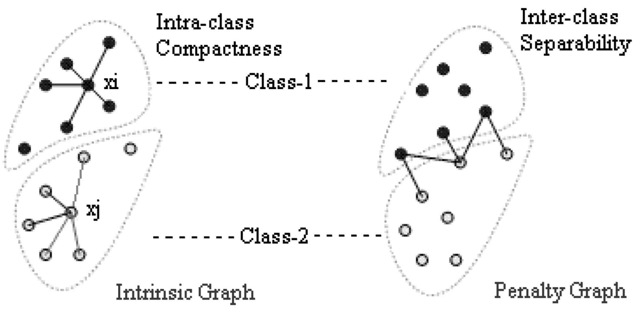


Fig. 4. The adjacency relationships of the intrinsic and penalty graphs for the Marginal Fisher Analysis algorithm. Note that the left adjacency graph only includes the edges for one sample in each class for greater clarity.

By following the graph embedding formulation, intraclass compactness is characterized from the intrinsic graph by the term

$$\begin{aligned}\tilde{S}_c &= \sum_i \sum_{i \in N_{k_1}^+(j) \text{ or } j \in N_{k_1}^+(i)} \|w^T x_i - w^T x_j\|^2 \\ &= 2w^T X(D-W)X^T w, \qquad (13)\\ W_{ij} &= \begin{cases} 1, & \text{if } i \in N_{k_1}^+(j) \text{ or } j \in N_{k_1}^+(i) \\ 0, & \text{else.} \end{cases}\end{aligned}$$

Here, $N_{k_1}^+(i)$ indicates the index set of the k_1 nearest neighbors of the sample x_i in the same class.

Interclass separability is characterized by a penalty graph with the term

$$\begin{aligned}\tilde{S}_p &= \sum_i \sum_{(i,j) \in P_{k_2}(c_i) \text{ or } (i,j) \in P_{k_2}(c_j)} \|w^T x_i - w^T x_j\|^2 \\ &= 2w^T X(D^p - W^p)X^T w, \qquad (14)\\ W_{ij}^p &= \begin{cases} 1, & \text{if } (i,j) \in P_{k_2}(c_i) \text{ or } (i,j) \in P_{k_2}(c_j) \\ 0, & \text{else.} \end{cases}\end{aligned}$$

Here, $P_{k_2}(c)$ is a set of data pairs that are the k_2 nearest pairs among the set $\{(i,j), i \in \pi_c, j \notin \pi_c\}$.

The algorithmic procedure of Marginal Fisher Analysis algorithm is formally stated as follows:

1. **PCA projection**. We first project the data set into the PCA subspace by retaining $N - N_c$ dimensions or a certain energy. Let W_{PCA} denote the transformation matrix of PCA.
2. **Constructing the intraclass compactness and interclass separability graphs**. In the intraclass compactness graph, for each sample x_i, set the adjacency matrix $W_{ij} = W_{ji} = 1$ if x_i is among the k_1-nearest neighbors of x_j in the same class. In the interclass separability graph, for each class c, set the similarity matrix $W_{ij}^p = 1$ if the pair (i,j) is among the k_2 shortest pairs among the set $\{(i,j), i \in \pi_c, j \notin \pi_c\}$.
3. **Marginal Fisher Criterion**. From the linearization of the graph embedding framework (4), we have the Marginal Fisher Criterion

$$w^* = \arg\min_w \frac{w^T X(D-W)X^T w}{w^T X(D^p - W^p)X^T w}, \qquad (15)$$

which is a special linearization of the graph embedding framework with

$$B = D^p - W^p.$$

4. **Output the final linear projection direction** as

$$w = W_{PCA} w^*.$$

In comparison to LDA, MFA has the following advantages: 1) The available projection directions are much greater than that of LDA and the dimension size is determined by k_2, the selected number of shortest pairs of in-class and out-of-class sample pairs. 2) There is no assumption on the data distribution of each class and the intraclass compactness is characterized by the sum of the distances between each data and its k_1-nearest neighbors of the same class. Thus, it is more general for discriminant analysis. 3) Without prior information on data distributions, the interclass margin can better characterize the separability of different classes than the interclass variance in LDA.

3.2 Kernel Marginal Fisher Analysis

The kernel trick is widely used to enhance the separation ability of a linear supervised dimensionality reduction algorithm. Marginal Fisher Analysis can be further improved by using the kernel trick. Assume that the kernel function $k(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$ is applied and the kernel Gram matrix is K with $K_{ij} = K(x_i, x_j)$. Let the projection direction be $w = \sum_{i=1}^N \alpha_i \phi(x_i)$, then the optimal α can be obtained as

$$\alpha^* = \arg\min_\alpha \frac{\alpha^T K(D-W)K\alpha}{\alpha^T K(D^p - W^p)K\alpha}. \qquad (16)$$

Note that the graphs for Kernel Marginal Fisher Analysis (KMFA) may be different from MFA as the k_1-nearest neighbors for each sample in KMFA may be different from that in MFA. The k_1 nearest in-class neighbors of each sample and the k_2 closest out-of-class sample pairs for each class are measured in the higher dimensional Hilbert space mapped from the original feature space with the kernel mapping function $\phi(x)$. The distance between sample x_i and x_j is obtained as

$$D(x_i, x_j) = \sqrt{k(x_i, x_i) + k(x_j, x_j) - 2k(x_i, x_j)}.$$

For a new data point x, its projection to the derived optimal direction is obtained as

$$\begin{aligned} F(x, \alpha^*) &= \lambda \sum_{i=1}^n \alpha_i^* k(x, x_i), \\ \lambda &= (\alpha^{*T} K \alpha^*)^{-1/2}. \end{aligned} \qquad (17)$$

3.3 Tensor Marginal Fisher Analysis

When the objects are represented as tensors of arbitrary order, from the tensorization of the graph embedding framework, we have the following formulation:

$$(w^1, \ldots, w^n)^* = \arg\min_{w^k, k=1,\ldots,n} \frac{\sum_{i \neq j} \|\mathbf{X_i} \times_1 w^1 \times_2 w^2 \ldots \times_n w^n - \mathbf{X_j} \times_1 w^1 \times_2 w^2 \ldots \times_n w^n\|^2 W_{ij}}{\sum_{i \neq j} \|\mathbf{X_i} \times_1 w^1 \times_2 w^2 \ldots \times_n w^n - \mathbf{X_j} \times_1 w^1 \times_2 w^2 \ldots \times_n w^n\|^2 W_{ij}^p}. \qquad (18)$$

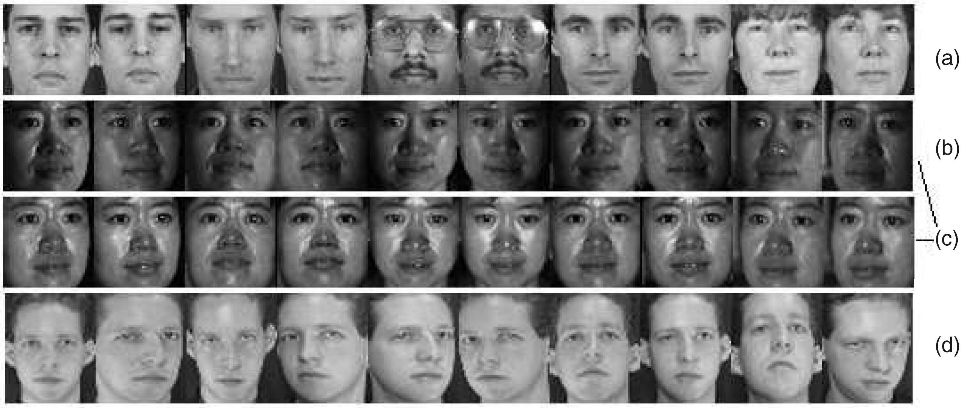


Fig. 5. The sample images cropped from the face database XM2VTS, PIE-1, PIE-2, and ORL, respectively. Note that the set PIE-1 is a subset of PIE-2. (a) XM2VTS, (b) PIE-1, (c) PIE-2, and (d) ORL.

As introduced in Section 2.1, the optimal projection vector can be obtained in an iterative manner.

4 Experiments

To evaluate the proposed Marginal Fisher Analysis (MFA) algorithm, we systematically compare it with the LDA algorithm on real-world and artificial data. On the real-world data, three benchmark databases, XM2VTS [12], CMU PIE [19], and ORL [17], are used to evaluate the separability of the lower dimensional representation derived from MFA in comparison to LDA. In addition, we compare on the face recognition problem the proposed Kernel MFA to Kernel Discriminant Analysis and second-order Tensor Marginal Fisher Analysis to DATER, respectively. Finally, we present a synthetic data set for clearly comparing performance in deriving the optimal discriminating direction when the data do not follow a Gaussian distribution. This additional test helps to support our observation that the MFA algorithm is general in nature.

4.1 Face Recognition

In our experiments, we use the XM2VTS, CMU PIE, and ORL databases for face recognition to evaluate our proposed MFA, Kernel MFA (KMFA), Tensor MFA (TMFA) algorithms. In all the experiments, all the images are aligned by fixing the locations of the two eyes. Histogram equalization is applied as a preprocessing step and after applying dimensionality reduction, the nearest neighbor classifier is used for final classification.

The XM2VTS database contains 295 people, where each person has four frontal face images taken in four different sessions. In this experiment, the samples in the first three sessions are used for training and the samples in the first session and the last session are used, respectively, as the gallery and probe sets. The size of each image is 64×64. The CMU PIE (Pose, Illumination and Expression) database contains more than 40,000 facial images of 68 people. The images were acquired across different poses, under variable illumination conditions, and with different facial expressions. In this experiment, two subdatabases are chosen for the evaluation of our proposed algorithms. In the first subdatabase, referred to as PIE-1, five near frontal poses (C27, C05, C29, C09, and C07) and illumination indexed as 10 and 13 are used such that each person has 10 images. Another subdatabase, PIE-2, consists of the same five poses as in PIE-1, but the illumination conditions indexed as 01 and 11 are additionally used, giving each person 20 images. The ORL database contains 400 images of 40 individuals. The images were captured at different times and with different variations including expression and facial details. The size of each image is 56×46. For the CMU PIE database and ORL database, the image set is partitioned into the different gallery and probe sets where Gm/Pn indicates that m images per person are randomly selected for training and the remaining n images are used for testing. Some cropped sample data are displayed in Fig. 5.

4.1.1 *MFA versus Fisherface [2]*

We first evaluate the performance of MFA in comparison to the Fisherface algorithm, i.e., PCA followed by LDA. For both algorithms, we retain $N - N_c$ dimensions in the PCA step. For a fair comparison with the Fisherface algorithm, we explore the performance on all possible feature dimensions in the LDA step and report the best result. Moreover, we also explore all PCA dimensions retaining energies of 90 to 99 percent along with all possible LDA dimensions, referred to as PCA + LDA in our results. The corresponding PCA + MFA is also evaluated.

As in algorithms such as ISOMAP, LLE, and Laplacian Eigenmap, how to set parameters is still an open problem. We therefore empirically set the parameters k_1 and k_2 of MFA in all face recognition experiments. Specifically, we sampled five values of k_1 between two and $(min_c\{n_c\} - 1)$ and chose the value with the best MFA performance. We similarly choose the best k_2 between 20 and $8N_c$ at sampled intervals of 20.

The experiments are conducted on both the XM2VTS and PIE-2 subdatabase. The face recognition results, listed in Tables 2 and 3, demonstrate that our proposed MFA consistently performs better than Fisherface and PCA + MFA outperforms PCA + LDA in most cases. All the results from the LDA and MFA related algorithms are better than that of the baseline algorithm PCA.

TABLE 2
Recognition Accuracies of PCA, Fisherface, and MFA, as Well as PCA + LDA and PCA + MFA on the XM2VTS Database (64×64)

PCA	69.8% (84)
Fisherface	81.7% (169)
MFA	84.8% (274)
PCA+LDA	94.6% (90, 83)
PCA+MFA	95.3% (90, 68)

Note that the numbers in parentheses are the corresponding feature dimensions with the best results after dimensionality reduction. For PCA + LDA and PCA + MFA, the first number is the percentage of energy retained in the PCA step.

TABLE 3
Recognition Accuracies of PCA, Fisherface, MFA, as Well as PCA + LDA and PCA + MFA on the PIE-2 Subdatabase (64×64)

	G4/P16	G5/P15	G6/P14
PCA	43.4% (252)	49.2% (264)	49.2% (341)
Fisherface	76.9% (62)	83.6% (61)	88.9% (62)
MFA	82.5% (70)	87.3% (117)	90.5% (69)
PCA+LDA	78.8% (94, 44)	84.3% (93, 36)	91.2% (91, 20)
PCA+MFA	83.9% (99, 71)	87.5% (99, 104)	90.8% (91, 32)

Note that the numbers in parentheses are the corresponding feature dimensions with the best results after dimensionality reduction. For PCA + LDA and PCA + MFA, the first number is the percentage of energy retained in the PCA step.

4.1.2 Comprehensive Evaluation of Graph Embedding Framework

In this section, we systematically evaluate Kernel Marginal Fisher Analysis (KMFA) in comparison with the traditional Kernel Discriminant Analysis (KDA) algorithm and the second-order Tensor Marginal Fisher Analysis (TMFA) to the recently proposed DATER of second-order, i.e., 2DLDA. For a comprehensive comparison, we also compare the above algorithms with baseline algorithms PCA, Fisherface, Bayesian Face, and the LPP algorithm. Bayesian Face is implemented as in [15], and LPP is implemented as in [10], with parameter k (number of neighboring points) of LPP selected in the same way for k_1 in MFA.

In all the experiments, the Gaussian Kernel $exp\{-\|x-y\|^2/\delta^2\}$ is used and parameter δ is set as $\delta = 2^{(n-10)/2.5}\delta_0$, $n = 0, 1, \ldots, 20$, where δ_0 is the standard derivation of the training data set. The reported result is the best one among the 21 configurations. Similar to observations made in [16], until now, it is still unclear how to choose the optimal kernel parameter. The PIE-1 subdatabase and the ORL database are used for the evaluation. We report five sets of experimental results, which are listed in Tables 4 and 5. From these results, we make several interesting observations:

1. The kernel trick can improve face recognition accuracy for both KDA and KMFA beyond the corresponding linear algorithms. The results demonstrate that the linearly inseparable data have the potential to be linearly separable in the Hilbert feature space transformed by the implicit kernel map of the defined kernel function. KMFA can be seen to outperform PCA, Fisherface, LPP, and MFA in most cases.

TABLE 4
Face Recognition Accuracies of PCA, Fisherface, LPP, MFA, Bayesian Face, PCA + LDA, MFA + LDA, KDA and KMFA, DATER, and TMFA on the ORL Database

	G3/P7	G4/P6	G5/P5
PCA	84.6% (116)	87.9% (82)	96.0% (44)
Fisherface	87.9% (31)	88.3% (28)	94.0% (39)
LPP	85.0% (60)	89.6% (141)	96.0% (158)
MFA	89.3% (39)	91.3% (46)	96.0% (42)
Bayesian Face	91.8%	90.0%	96.0%
PCA+LDA	90.7% (99, 36)	91.3% (99, 38)	98.0% (91, 38)
PCA+MFA	92.1% (99, 51)	91.7% (90, 42)	98.0% (90, 38)
KDA	87.5% (7, 27)	91.7% (6, 39)	98.5% (9, 51)
KMFA	88.6% (7, 26)	93.8% (7, 60)	98.5% (3, 66)
DATER	89.3% (18, 37)	92.0% (9, 5)	97.5% (31, 4)
TMFA	95.0% (41, 8)	96.3% (3, 45)	98.5% (14, 3)

Note that the numbers in parentheses are the corresponding feature dimensions with the best results after the dimensionality reduction. For PCA + LDA and PCA + MFA, the first number is the percentage of energy retained in the PCA step, for KDA and KMFA, the number is the kernel parameter sequence number and, for DATER and TMFA, the two numbers are the reduced row and column numbers, respectively.

TABLE 5
Face Recognition Accuracies of PCA, Fisherface, LPP, MFA, Bayesian Face, PCA + LDA, MFA + LDA, KDA and KMFA, DATER, and TMFA on the PIE-1 Subdatabase

	G4/P6	G3/P7
PCA	38.9% (234)	28.3% (178)
Fisherface	80.2% (58)	65.8% (58)
LPP	42.3% (77)	32.7% (150)
MFA	84.9% (68)	71.0% (54)
Bayesian Face	79.6%	75.1%
PCA+LDA	81.5% (95, 23)	77.1% (99, 62)
PCA+MFA	85.5% (99, 69)	79.6% (99, 61)
KDA	81.0% (2, 15)	70.0% (0, 26)
KMFA	85.2% (0, 47)	72.3% (0, 34)
DATER	82.3% (12, 6)	80.0% (11, 8)
TMFA	85.2% (9 , 7)	82.1% (17, 6)

Note that the numbers in parentheses are the corresponding feature dimensions with the best results after the dimensionality reduction. For PCA + LDA and PCA + MFA, the first number is the percentage of energy retained in the PCA step, for KDA and KMFA, the number is the kernel parameter sequence number and, for DATER and TMFA, the two numbers are the reduced row and column numbers, respectively.

2. The results demonstrate that, when the training set adequately characterizes the data distribution, such as the cases of G4/P6 and G5/P5 for the ORL database, LPP has the potential to outperform Fisherface and PCA as reported in [10]. However, when the data set distribution is complex and the training data cannot represent the data distribution well, such as for the PIE-1 subdatabase, where three or four samples of each person are not enough to characterize a data distribution that includes five poses, LPP appears to be less effective than Fisherface in these cases, though still better than the PCA algorithm. All the results show that LPP does not perform better than MFA.
3. The results show that the performance can be substantially improved by exploring a certain range of PCA dimensions before conducing LDA or MFA.

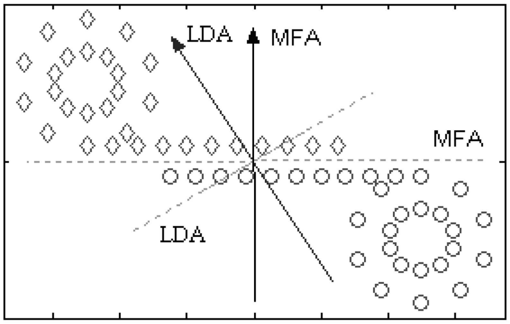


Fig. 6. A synthetic data problem: the comparative optimal projections from Marginal Fisher Analysis ($k_1 = 5$, $k_2 = 250$) and LDA. Note that the solid line and dashed line represent the optimal projection direction and optimal classification hyperline, respectively.

We can also see that the PCA + LDA (PCA + MFA) combinations are comparable with the kernel version of the LDA (MFA) algorithm.

4. The Bayesian Face algorithm performs better than PCA, Fisherface, and LPP in most cases and it is comparable to MFA. However, it is consistently worse than PCA + MFA in all cases.
5. The tensor representation brings encouraging performance improvements compared to the corresponding vector-based algorithms. TMFA (DATER) is shown to be much better than MFA (Fisherface) in all cases. Moreover, in most cases, the TMFA (DATER) algorithm performs better than KMFA (KDA) and PCA + MFA (PCA + LDA).
6. Another interesting observation is that, when the training sample size is large enough to sufficiently characterize the data distribution, such as the case for the G5/P5 ORL database, all algorithms we discussed in this work can achieve similar performance. This fact shows that, for a real-world application, it is critical to collect sufficient samples for all subjects.

4.2 A Non-Gaussian Case

Generally, MFA can work well when the marginal sample pairs are sufficient to characterize the separability of different classes. In this artificial problem, a two-class classification problem is designed. The objective of this artificial problem is to justify that when the data do not follow a Gaussian distribution, the original LDA algorithm may fail to find the optimal projection direction for classification, whereas the MFA algorithm can find a much better direction.

The data for each class are distributed as shown in Fig. 6. Here, the circles and the diamonds represent samples of different classes, which obviously do not exhibit a Gaussian distribution for each class. The solid lines in Fig. 6 represent the learned optimal projection directions from Marginal Fisher Analysis (MFA) and Linear Discriminant Analysis (LDA), respectively, and the dashed lines are the optimal classification lines for MFA and LDA. The results clearly demonstrate that LDA fails to find the optimal direction in the case with non-Gaussian distributed data. However, MFA successfully derives the discriminative direction because of its consideration of marginal points.

Although this example represents an extreme case of non-Gaussian data distributions that is not intended to represent the only situations or even the majority of situations where the MFA algorithm works, we utilize it to clearly justify the intuition that, when the data do not follow a Gaussian distribution, LDA may fail to find the optimal direction, whereas MFA can find an a more discriminative direction.

5 Conclusion and Future Work

In this paper, we aim to provide insights into the relationship among the state-of-the-art dimensionality reduction algorithms, as well as to facilitate the design of new algorithms. A general framework known as graph embedding, along with its linearization, kernelization, and tensorization, has been proposed to provide a unified perspective for the understanding and comparison of many popular dimensionality reduction algorithms. Moreover, the graph embedding framework can be used as a general platform to develop new algorithms for dimensionality reduction. As shown in this paper, we have proposed a novel dimensionality reduction algorithm called Marginal Fisher Analysis by designing two graphs that characterize the intraclass compactness and the interclass separability, respectively, and by optimizing their corresponding criteria based on the graph embedding framework. This new algorithm is shown to effectively overcome the data distribution assumption of the traditional LDA algorithm. Thus, MFA is a more general algorithm for discriminant analysis.

A byproduct of this paper is a series of linearization, kernelization and tensorization versions of the ISOMAP and LLE algorithms. One of our future works is to systematically compare all possible extensions of the algorithms mentioned in this paper. Another possible extension of this work is the combination of the kernel trick and tensorization. Although there have already been some attempts to address this issue [27], there still exists no theoretically reasonable way to map the tensor data to another higher or even infinite tensor space while simultaneously providing an efficient solution. We intend to further investigate this issue in both theory and practice.

An open problem in MFA is the selection of parameters k_1 and k_2, which is also an unsolved problem in algorithms such as ISOMAP, LLE, Laplacian Eigenmap, and LPP. Additional theoretical analysis is needed for this topic. Moreover, there are also cases under which MFA may fail. For example, when the value of k_2 is insufficiently large, a nonmarginal pair from different classes in the original feature space may be very close in the dimension-reduced feature space and degrade classification accuracy.

There are also certain limitations in the graph embedding framework. For example, this framework only considers the L_2 distance as a similarity measure, which means that it can only take into account the second-order statistics of the data set. How to utilize higher order statistics of the data set in the graph embedding framework is also an interesting direction for future study.

Appendix A

The ISOMAP algorithm can be reformulated as the direct graph embedding formulation in (3) with the similarity matrix $W_{ij} = \tau(D_G)_{ij}$ if $i \neq j$; 0 otherwise; and $B = I$.

Proof. With matrix $\tau(D_G) = -HSH/2$, we have, for any i,

$$\begin{aligned}
&\sum_j \tau(D_G)_{ij} \\
&= \sum_j (-HSH/2)_{ij} \\
&= \sum_j \left(-\left(I - \frac{1}{N} ee^T\right) S \left(I - \frac{1}{N} ee^T\right)/2 \right)_{ij} \\
&= \frac{1}{2} \sum_j (-S_{ij} + \frac{1}{N} \sum_{i'} S_{i'j} + \frac{1}{N} \sum_{j'e} \left(S_{ij'} - \frac{1}{N} \sum_{i'} S_{i'j'}\right) \\
&= \left(-\frac{1}{2} \sum_j S_{ij} + \frac{1}{2N} \sum_{jj'} S_{ij'} \right) + \left(\frac{1}{2N} \sum_{ji'} S_{i'j} - \frac{1}{2N^2} \sum_{jj'i'} S_{i'j'} \right) \\
&= 0 + 0 = 0.
\end{aligned}$$

□

Hence, the row sum of matrix $\tau(D_G)$ is zero, so it can be considered as the Laplacian matrix of a graph and if we define a graph by setting the nondiagonal element as $W_{ij} = \tau(D_G)_{ij}$ if $i \neq j$; 0 otherwise; then we have

$$\begin{aligned}
y^* &= \arg\max_{y^T y = \lambda} y^T \tau(D_G) y = \arg\max_{y^T y = \lambda} y^T (W - D) y \\
&= \arg\min_{y^T y \lambda} y^T (D - W) y.
\end{aligned}$$

Note that constant d is the corresponding eigenvalue of $\tau(D_G)$, which is different from the other algorithms in which d is mostly set to 1. Therefore, we can conclude that the ISOMAP algorithm can be reformulated in the graph embedding formulation in (3).

Appendix B

The LLE algorithm can be reformulated as the direct graph embedding formulation in (3) with similarity matrix $W_{ij} = (M + M^T - M^T M)_{ij}$ if $i \neq j$; 0 otherwise; and $B = I$.

Proof. With simple algebraic computation, we have [18]

$$\sum_i \|y_i - \sum_j M_{ij} y_j\|^2 = y^T (I - M)^T (I - M) y.$$

□

On the other hand, $\sum_j M_{ij} = 1, \forall\, i$, thus

$$\begin{aligned}
&\sum_j [(I - M)^T (I - M)]_{ij} \\
&= \sum_j I_{ij} - M_{ij} - M_{ji} + (M^T M)_{ij} \\
&= 1 - \sum_j M_{ij} - \sum_j M_{ji} + \sum_j \sum_k M_{ki} M_{kj} \\
&= 1 - 1 - \sum_j M_{ji} + \sum_k M_{ki} = 0.
\end{aligned}$$

Therefore, the matrices $(I - M^T)(I - M)$ can be considered as Laplacian matrices of a graph. If we set $W_{ij} = (M + M^T - M^T M)_{ij}$ when $i \neq j$, 0 otherwise, then

$$y^* = \arg\min_{y^T y = 1} y^T (I - M^T)(I - M) y = \arg\min_{y^T y = 1} y^T (D - W) y.$$

That is, the LLE algorithm can be reformulated as the direct graph embedding formulation as in (3).

References

[1] A. Batur and M. Hayes, "Linear Subspaces for Illumination Robust Face Recognition," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 296-301, Dec. 2001.

[2] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 7, pp. 711-720, July 1997.

[3] M. Belkin and P. Niyogi, "Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering," *Advances in Neural Information Processing System,* vol. 14, pp. 585-591, 2001.

[4] Y. Bengio, J. Paiement, P. Vincent, O. Delalleau, N. Roux, and M. Ouimet, "Out-of-Sample Extensions for LLE, ISPMAP, MDS, Eigenmaps, and Spectral Clustering," *Advances in Neural Information Processing Systems,* vol. 16, 2004.

[5] M. Brand, "Continuous Nonlinear Dimensionality Reduction by Kernel Eigenmaps," Technical Report 2003-21, Mitsubishi Electric Research Labs, Apr. 2003.

[6] F. Chung, "Spectral Graph Theory," *Regional Conf. Series in Math.,* no. 92, 1997.

[7] Y. Fu and T. Huang, "Locally Linear Embedded Eigenspace Analysis," IFP-TR, Univ. of Illinois at Urbana-Champaign, Jan. 2005.

[8] K. Fukunnaga, *Introduction to Statistical Pattern Recognition,* second ed. Academic Press, 1991.

[9] D. Hand, *Kernel Discriminant Analysis.* Research Studies Press, 1982.

[10] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang, "Face Recognition Using Laplacianfaces," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 3, pp. 328-340, Mar. 2005.

[11] I. Joliffe, *Principal Component Analysis.* Springer-Verlag, 1986.

[12] J. Luettin and G. Maitre, "Evaluation Protocol for the Extended M2VTS Database (XM2VTS)," *DMI for Perceptual Artificial Intelligence,* 1998.

[13] J. Ham, D. Lee, S. Mika, and B. Schölkopf, "A Kernel View of the Dimensionality Reduction of Manifolds," *Proc. Int'l Conf. Machine Learning,* pp. 47-54, 2004.

[14] A.M. Martinez and A.C. Kak, "PCA versus LDA," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 2, pp. 228-233, Feb. 2001.

[15] B. Moghaddam, T. Jebara, and A. Pentland, "Bayesian Face Recognition," *Pattern Recognition,* vol. 33, pp. 1771-1782, 2000.

[16] K. Muller, S. Mika, G. Riitsch, K. Tsuda, and B. Schölkopf, "An Introduction to Kernel-Based Learning Algorithms," *IEEE Trans. Neural Networks,* vol. 12, pp. 181-201, 2001.

[17] Olivetti & Oracle Research Laboratory, The Olivetti & Oracle Research Laboratory Face Database of Faces, http://www.cam-orl.co.uk/facedatabase.html, 1994.

[18] S. Roweis and L. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science,* vol. 290, no. 22, pp. 2323-2326, Dec. 2000.

[19] T. Sim, S. Baker, and M. Bsat, "The CMU Pose, Illumination, and Expression Database," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 12, pp. 1615-1618, Dec. 2003.

[20] J. Tenenbaum, V. Silva, and J. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science,* vol. 290, no. 22, pp. 2319-2323, Dec. 2000.

[21] Y. Teh and S. Roweis, "Automatic Alignment of Hidden Representations," *Advances in Neural Information Processing System,* vol. 15, pp. 841-848, 2002.

[22] M. Turk and A. Pentland, "Face Recognition Using Eigenfaces," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 586-591, 1991.

[23] M. Vasilescu and D. Terzopoulos, "TensorTextures: Multilinear Image-Based Rendering," *ACM Trans. Graphics,* vol. 23, no. 3, pp. 336-342, 2004.

[24] X. Wang and X. Tang, "Dual-Space Linear Discriminant Analysis for Face Recognition," *Proc. Computer Vision and Pattern Recognition,* vol. 2, pp. 564-569, 2004.

[25] D. Xu, S. Yan, L. Zhang, Z. Liu, and H. Zhang, "Coupled Subspaces Analysis," Microsoft Research Technical Report, MSR-TR-2004-106, Oct. 2004.

[26] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H. Zhang, "Discriminant Analysis with Tensor Representation," *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 526-532, 2005.

[27] S. Yan, D. Xu, L. Zhang, B. Zhang, and H. Zhang, "Coupled Kernel-Based Subspace Learning," *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 645-650, 2005.

[28] S. Yan, H. Zhang, Y. Hu, B. Zhang, and Q. Cheng, "Discriminant Analysis on Embedded Manifold," *Proc. Eighth European Conf. Computer Vision,* vol. 1, pp. 121-132, May 2004.

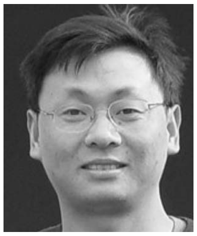
[29] J. Yang, D. Zhang, A. Frangi, and J. Yang, "Two-Dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 1, pp. 131-137, Jan. 2004.

[30] J. Ye, "Generalized Low Rank Approximations of Matrices," *Proc. Int'l Conf. Machine Learning,* pp. 895-902, 2004.

[31] J. Ye, R. Janardan, and Q. Li, "Two-Dimensional Linear Discriminant Analysis," *Advances in Neural Information Processing Systems,* vol. 17, pp. 1569-1576, 2005.

[32] J. Ye, R. Janardan, C. Park, and H. Park, "An Optimization Criterion for Generalized Discriminant Analysis on Undersampled Problems," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 8, pp. 982-994, Aug. 2004.

[33] H. Yu and J. Yang, "A Direct LDA Algorithm for High Dimensional Data-with Application to Face Recognition," *Pattern Recognition,* vol. 34, pp. 2067-2070, 2001.

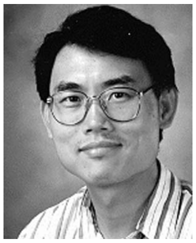


Shuicheng Yan received the BS and PhD degrees from the Applied Mathematics Department, School of Mathematical Sciences, Peking University, China, in 1999 and 2004, respectively. His research interests include computer vision and machine learning and he is a member of the IEEE.

Dong Xu received the BS and PhD degrees from the Electronic Engineering and Information Science Department, University of Science and Technology of China, in 2001 and 2005, respectively. His research interests include pattern recognition, computer vision, and machine learning.

Benyu Zhang received the bachelor's and master's degrees in computer science from Peking University in 1999 and 2002. He joined Microsoft Research, China, in July 2002 to pursue his wide-ranging research interests in machine learning, information retrieval, data mining, and artificial intelligence.

Hong-Jiang Zhang received the PhD degree from the Technical University of Denmark and the BS degree from Zhengzhou University, China, both in electrical engineering, in 1982 and 1991, respectively. From 1992 to 1995, he was with the Institute of Systems Science, National University of Singapore, where he led several projects in video and image content analysis and retrieval and computer vision. From 1995 to 1999, he was a research manager at Hewlett-Packard Labs, responsible for research and technology transfers in the areas of multimedia management, computer vision, and intelligent image processing. In 1999, he joined Microsoft Research and is currently the Managing Director of the Advanced Technology Center. Dr. Zhang has authored three books, more than 300 referred papers, eight special issues of international journals on image and video processing, content-based media retrieval, and computer vision, as well as more than 50 patents or pending applications. He currently serves on the editorial boards of five IEEE/ACM journals and a dozen committees of international conferences. He is a fellow of the IEEE.



Qiang Yang received the PhD degree from the University of Maryland, College Park. He is a faculty member in the Hong Kong University of Science and Technology's Department of Computer Science. His research interests are AI planning, machine learning, case-based reasoning, and data mining. He is a senior member of the IEEE and an associate editor for the *IEEE Transactions on Knowledge and Data Engineering* and *IEEE Intelligent Systems*.

Stephen Lin received the BSE degree in electrical engineering from Princeton University and the PhD degree in computer science and engineering from the University of Michigan. He is a researcher in the Internet Graphics Group of Microsoft Research Asia. His current research interests include physics-based computer vision and reflectance modeling in computer graphics.