Graph Embedding: A General Framework for Dimensionality Reduction*

[1]Shuicheng Yan, [2]Dong Xu, [3]Benyu Zhang and [3]Hong-Jiang Zhang

[1]Department of Information Engineering, the Chinese University of Hong Kong, Shatin, Hong Kong
[2] MOE-Microsoft Key Laboratory of Multimedia Computing and Communication & Department of EEIS,
University of Science and Technology of China, Hefei, Anhui, P. R. China
[3]Microsoft Research Asia, Beijing, P.R. China
Contact: scyan@ie.cuhk.edu.hk

Abstract

In the last decades, a large family of algorithms—supervised or unsupervised; stemming from statistic or geometry theory—have been proposed to provide different solutions to the problem of dimensionality reduction. In this paper, beyond the different motivations of these algorithms, we propose a general framework, graph embedding along with its linearization and kernelization, which in theory reveals the underlying objective shared by most previous algorithms. It presents a unified perspective to understand these algorithms; that is, each algorithm can be considered as the direct graph embedding or its linear/kernel extension of some specific graph characterizing certain statistic or geometry property of a data set. Furthermore, this framework is a general platform to develop new algorithm for dimensionality reduction. To this end, we propose a new supervised algorithm, Marginal Fisher Analysis (MFA), for dimensionality reduction by designing two graphs that characterize the intra-class compactness and inter-class separability, respectively. MFA measures the intra-class compactness with the distance between each data point and its neighboring points of the same class, and measures the inter-class separability with the class margins; thus it overcomes the limitations of traditional Linear Discriminant Analysis algorithm in terms of data distribution assumptions and available projection directions. The toy problem on artificial data and the real face recognition experiments both show the superiority of our proposed MFA in comparison to LDA.

1. Introduction

The techniques for dimensionality reduction in supervised or unsupervised manner have attracted much attention in computer vision and pattern recognition. Among the linear algorithms, *i.e.* the subspace learning algorithms, *principal component analysis* (PCA) [7][9] and *linear discriminant analysis* (LDA) [4][9][19] are the two most popular ones and both were proposed with Gaussian assumptions on data distributions. Recently, He *et al.* [6] proposed the *locality preserving projections* (LPP) to pursue a subspace that preserves the local information and detect the essential manifold structure. ISOMAP [14], LLE [12], Laplacian Eigenmap [2] are three recently developed algorithms to conduct nonlinear dimensionality reduction for the data set lying on or nearly on a lower dimensional manifold. On the other hand, the kernel-trick [10] has been widely applied to extend the linear dimensionality reduction algorithms into nonlinear ones by performing the linear algorithm on higher or even infinite dimensional features transformed from the implicit mapping function involved in the kernel function.

In this paper, beyond the different motivations of the above mentioned algorithms, a general framework, graph embedding along with its linearization and kernelization, is proposed to provide a unified view to understand and explain these algorithms. The purpose of the *direct graph embedding* is to represent each vertex of a graph as a one dimensional vector by preserving the similarities of vertex pairs measured by the graph similarity matrix, which characterizes certain statistic or geometry property of the data set. As described later in this paper, the vector representation of the vertices can be derived as the eigenvectors corresponding to the leading eigenvalues of the graph Laplacian matrix with certain constraints. 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; while the *kernelization of the graph embedding* applies the kernel trick on the linear graph embedding algorithm to obtain the nonlinear embedding. These three types of graph embeddings consist of the unified framework from direct graph embedding to its linearization and kernelization. The *direct graph embedding* only presents the mappings for the graph vertices, while its extensions provide the mapping for all samples in the original feature space. As justified later, the above mentioned algorithms, such as PCA, LDA, LPP, ISOMAP, LLE and Laplacian Eigenmap, can all be reformulated in this common framework; and their differences lie in the strategy to design the graph and the embedding type. This framework

* This work was performed at Microsoft Research Asia.

1063-6919/05 $20.00 © 2005 **IEEE**

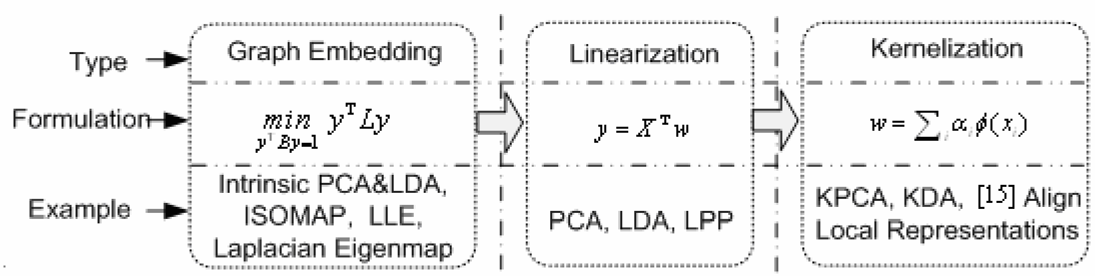


Figure 1. Graph Embedding along with its linearization and kernelization: a unified framework for general dimensionality reduction. The top row is the graph type; the middle row is the corresponding objective function and the third row lists the sample algorithms.

reveals the underlying objective shared by all these algorithms.

Besides presenting a common framework to unify the previous algorithms for dimensionality reduction, the graph embedding framework can also be used as a general platform to develop new algorithm for dimensionality reduction by designing graphs with special motivations. Thereby, by using this framework as platform, we design a novel algorithm for dimensionality reduction from the following observations.

Despite the success of LDA in many applications, its application is still limited due to: 1) in theory, the number of available projection directions is lower than the class number; and 2) LDA characterizes the discriminating capability with the inter-class and intra-class scatters; it is optimal only in the cases that the data for each class is approximately Gaussian distributed, which can not always be satisfied in real world applications. Many efforts have been devoted to improve the performance of LDA. Among them, Null subspace [17] algorithm is the most popular one and it effectively applied the null subspace of the intra-class scatter matrix. However, the fundamental problem and limitations in LDA are still not solved in theory.

In this paper, we propose a novel dimensionality reduction, *Marginal Fisher Analysis (MFA)*, to overcome the limitations of LDA. *MFA* is developed by using the graph embedding framework as a platform. Two graphs are designed to characterize the intra-class compactness and the inter-class separability, respectively. In the first graph, the vertex pair is connected if one is among the k_1-nearest neighbors of the other and they belong to the same class. In the later one, for each class, the k_2-nearest in-class and out-class vertex pairs are connected. *MFA* and the kernel-MFA aim at providing the graph embeddings that preserve the characteristic of the first graph and at the same time suppress the characteristic of the second graph. Compared with LDA, MFA has the following advantages: 1) the number of the available projection directions is much higher than that of LDA; and 2) there is no assumption on the data distribution, thus it is more general for discriminant analysis; and 3) without the prior assumption on data distributions, the inter-class margin can better characterize the separability of different classes than the inter-class scatter in LDA.

The rest of the paper is structured as follows. A unified formulation, graph embedding along with its linearization and kernelization, for general dimensionality reduction is introduced in Section 2. In Section 3, by using graph embedding as a general platform, the Marginal Fisher Analysis is proposed. The toy problem and face recognition experiments are showed in Section 4 to demonstrate the effectiveness of MFA/KMFA. Finally, we give the conclusion remarks in Section 5.

2. Graph Embedding: A General Framework for Dimensionality Reduction

In the past decades, many approaches have been proposed for dimensionality reduction. Among them, the most popular ones include PCA, LDA KPCA and KDA [10]. Recently, a large family of algorithms, such as ISOMAP, LLE and its extensive work [15], Laplacian Eigenmap and LPP, were developed based on the geometrical assumption that the data set approximately lies on a lower dimensional manifold embedded in the original higher dimensional feature space. Although their motivations are different, their final objectives are similar: to derive lower dimensional representations; thereby a natural question is whether they can be formulated in a similar way.

We give a position answer to this question. In this section, we present a novel formulation, graph embedding along with its linearization and kernelization, commonly shared by all the above mentioned algorithms and provide a unified perspective to understand these algorithms.

2.1 Graph Embedding

Denote the sample set as $X = [x_1, x_2, \cdots, x_N], x_i \in \mathbb{R}^m$. In the supervised learning problem, the class labels are assumed as $l_i \in \{1, 2, \cdots, N_c\}$ and denote π_c as the index set of samples belonging to class *c*. Let *G*={*X, W*} be an undirected weighted graph with vertex set *X* and the similarity matrix $W \in \mathbb{R}^{N \times N}$. The element of real symmetry matrix W measures the similarity, maybe *negative,* of the vertex pair, and can be computed in different ways, such as the Gaussian similarity from Euclidean distance as in [2], local neighborhood relationship as in [12], and prior class information in supervised learning algorithm as in [9].

The diagonal matrix *D* and the Laplacian matrix *L* of the graph *G* are defined as

$$L = D - W, \quad D_{ii} = \sum\nolimits_{j \neq i} W_{ij} \quad \forall i \tag{1}$$

In this work, the *graph embedding* of the graph *G* is defined as the optimal low dimensional vector representations for the vertices of graph *G* that best characterize the similarity relationship between the data pairs. For simplicity, we consider the one dimensional case and represent low dimensional representations of the vertices as vector y. It is easy to extend to multi-dimensional cases. A starting point to preserve the similarity of a graph is to minimize the *graph preserving criterion* as follows

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

where *c* is a constant, *B* is the constraint matrix and it may be also a Laplacian matrix, such as in LDA, of some graph of which the similarity characteristics should be avoided and thrown away. The similarity preservation property from *the graph preserving criterion* can be explained in two-fold: *if the similarity (positive) between the x_i and x_j is higher, then the distance between y_i and y_j will be smaller for the minimum; on the other hand, if the similarity (negative) between the vertex x_i and x_j is lower, the distance between y_i and y_j will be larger for the minimum.*

The *graph preserving criterion* provides the direct graph embedding for the data set; however, it can not directly present the low dimensional representation for new data. In the following, we present two methods to solve this issue.

Linearization, Assume the low dimensional vector representation can be obtained from linear projections as $y = X^{\mathrm{T}} w$; then, the objective function (2) is changed to

$$w^* = arg \min_{\substack{w^{\mathrm{T}} XBX^{\mathrm{T}} w = c \\ or \ w^{\mathrm{T}} w = c}} w^{\mathrm{T}} XLX^{\mathrm{T}} w \tag{3}$$

Kernelization, A direct way to extend the linear projections to nonlinear cases is to utilize the kernel trick. The intuition of the kernel trick is to map the data from the original input space to a higher dimensional Hilbert space as $\phi : x \to \mathcal{F}$ and then the algorithm is performed in this new feature space. It is well applied to the algorithms that only need to compute the inner product of data pairs $k(x, y) = \phi(x) \cdot \phi(y)$. Assume that the mapping function is $w = \sum\nolimits_i \alpha_i \phi(x_i)$ and K is the kernel Gram matrix with $K_{ij} = k(x_i, x_j)$, then we have

$$a^* = arg \min_{\substack{\alpha^{\mathrm{T}} KBK\alpha = c \\ or \ \alpha^{\mathrm{T}} K\alpha = c}} \alpha^{\mathrm{T}} KLK\alpha \tag{4}$$

The solutions of (2-4) can all be obtained by solving the generalized eigenvalue decomposition problem as

$$\tilde{L} v = \lambda \tilde{B} v \tag{5}$$

where $\tilde{L} = L, XLX^{\mathrm{T}} \ or \ KLK$, $\tilde{B} = I, B, K, XBX^{\mathrm{T}} \ or \ KBK$. For problem in Eq. (2), there is a trivial solution with all elements being the same and corresponding to eigenvalue zero. We omit it mostly as in [2].

Note that the formulation in Eq. (2) is very similar to that of Laplacian Eigenmap algorithm. However, as justified in the following subsection, besides Laplacian Eigenmap, most previous algorithms for dimensionality algorithms can all be reformulated as (2-4), and Laplacian Eigenmap is just a special case of the formulation (2).

2.2 General Framework for Dimensionality Reduction

In this subsection, we prove that all above mentioned dimensionality reduction algorithms share a common formulation *i.e.* graph embedding along with its linearization and kernelization. The differences of these algorithms lie in the ways to compute the similarity matrix of the graph and select the constraint matrix. Figure 1 provides an illustration on the graph embedding framework and also demonstrates the sample algorithms for different types of graph embeddings. In the following, we firstly give an overview on these algorithms.

PCA [7][16] pursues the projection direction with maximal variance. In other words, it finds and removes the projection direction with minimal variance, *i.e.*

$$w^* = arg \min_{w^{\mathrm{T}} w = 1} w^{\mathrm{T}} C w \quad \text{with} \tag{6}$$

$$C = \frac{1}{N}\sum_i (x_i - \overline{x})(x_i - \overline{x})^{\mathrm{T}} = \frac{1}{N}X(I - \frac{1}{N}ee^{\mathrm{T}})X^{\mathrm{T}}$$

where e is a N dimensional vector with $e = [1,1,...,1]^{\mathrm{T}}$, C is the covariance matrix and $\overline{x}$ is the mean of all samples. KPCA [10] applies the kernel trick on PCA.

LDA [9] searches for the directions that are most effective in discriminating. It minimizes the ratio between the intra-class scatter and the inter-class scatter as

$$w^* = \arg\min_w \frac{w^{\mathrm{T}} S_W w}{w^{\mathrm{T}} S_B w} = \arg\min_w \frac{w^{\mathrm{T}} S_W w}{w^{\mathrm{T}} C w}$$

$$S_W = \sum_{i=1}^{N} (x_i - \overline{x}^{l_i})(x_i - \overline{x}^{l_i})^{\mathrm{T}} = X(I - \sum_{c=1}^{N_c} \frac{1}{n_c} e^c e^{c\mathrm{T}})X^T \quad (7)$$

$$S_B = \sum_{c=1}^{N_c} n_c (\overline{x}^c - \overline{x})(\overline{x}^c - \overline{x})^{\mathrm{T}} = NC - S_W$$

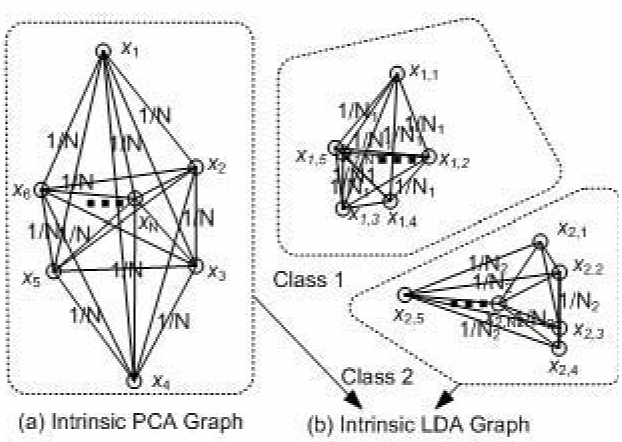
where $\overline{x}^c$ is the mean of the c-th class, and e^c is an N dimensional vector with $e^c(i) = 1, if\ c = l_i$; *0*, otherwise. KDA [10] is the kernel based discriminant analysis.

ISOMAP [14] aims to find the low dimensional representations for the data set that preserve the geodesic distances of the data pairs. Let D_G be the obtained approximated geodesic distance matrix, $\tau(D_G)$ =*-HSH/2*, where $H = I - \frac{1}{N}ee^{\mathrm{T}}$ and S_{ij}= $D_{G\,(i,j)}^{\,2}$, converts the distance matrix to the corresponding inner product matrix and the *MDS* [14] is conducted to obtain the low dimensionality representations for all the data points.

LLE [12] maps the input data to a lower dimensional space in a manner that preserves the relationship between the neighboring points. Firstly, 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 indices of the k nearest neighbors for the sample x_i and $\sum_{j \in N_k(i)} \| x_i - M_{ij} x_j \|^2$ is minimized; then, the low dimensional representation y is obtained by minimizing $\sum_i \sum_{j \in N_k(i)} \| y_i - M_{ij} y_j \|^2$. Roweis *et al.* [15] proposed a procedure to align disparate local linear presentations into a global coherent coordinate system by preserving the relationship between neighboring points as in LLE. As demonstrated in [18], it is actually a special Geometry-Adaptive-Kernel based LLE.

Laplacian Eigenmap (LE) [2] preserves the similarities of the neighboring points. As mentioned previously, its objective function is similar to that in Eq. (2) 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. The newly proposed LPP [6] is its linear approximation.

The above algorithms were proposed from different motivations; however, they in fact share a common formulation as in Eq. (2-4). Table 1 lists the similarity and constraint matrices for all above mentioned methods. And their corresponding graph embedding types are also demonstrated.



(a) Intrinsic PCA Graph (b) Intrinsic LDA Graph

Figure 2. The adjacency graphs for PCA and LDA.

From Eq. (6) and (7), we can easily have the listed formulations of the similarity matrix *W* and constraint matrix *B* for PCA/KPCA and LDA/KDA as listed in Table 1. Figure 2 plots the intrinsic graphs for PCA and LDA, respectively. It shows that PCA connects all the sample pairs with equal weights. LDA connects all the pairs with same class labels and the weights are in inverse proportion to the sample number of the corresponding class; meanwhile the Laplacian matrix of the graph in PCA is used as constraint matrix in LDA. LE and LPP directly follow the formulation for direct graph embedding and its linearization. The proofs of the W and B formulations for ISOMAP and LLE are presented as in appendix A and B.

2.3 Discussion

Ham *et al.* [8] also proposed a kernel view to interpret the KPCA, ISOMAP, LLE and Laplacian Eigenmap algorithm and demonstrated that they share the common KPCA formulation. Our framework and Ham's work present two intrinsically different perspectives to interpret these algorithms in a unified framework; however, they are different in many aspects:

1) Ham's work was proposed by considering the normalized similarity matrix as a kernel matrix; while our work discusses the Laplacian matrix derived from similarity matrix of the graph.
2) Ham's work can only discuss the unsupervised learning algorithm; while our proposed framework is more general and can be applied to analyze both unsupervised and supervised learning algorithms as described above.
3) Moreover, as showed later, our proposed framework can be utilized as a general platform and

tool to develop new algorithm for dimensionality reduction.

Table 1. Common graph embedding view for the most popular dimensionality reduction algorithms. Note that type *D* means direct graph embedding, while *L* and *K* mean the linearization and kernelization of the graph embedding, respectively.

Algorithm	W & B Definition	Type
PCA/KPCA	$W_{ij} = 1/N,\ i \neq j; B = I$	*L/K*
LDA/KDA	$W_{ij} = \delta_{l_i, l_j} / n_{l_i},\ B = I - \frac{1}{N} ee^{\mathrm{T}}$	*L/K*
ISOMAP	$W_{ij} = \tau(D_G)_{ij},\ i \neq j;\ \ B = I$	*D*
LLE/ [15]	$W = M + M^{\mathrm{T}} - M^{\mathrm{T}} M;\ B = I$	*D/K*
LE / LPP	$W_{ij} = \exp\{-\| x_i - x_j \|^2 / t\}$ if $i \in N_k(j)$ or $j \in N_k(i)$;*B*=*D*	*D/L*

3. Marginal Fisher Analysis: Using Graph Embedding as a Platform

As discussed above, most popular dimensionality reduction algorithms can be reformulated in the proposed unified framework, graph embedding along with its linear or kernel extensions. The proposed framework can be also used as a general platform to design new algorithms for dimensionality reduction. The straightforward byproducts of above analysis are the linear and kernel extensions of the very popular ISOMAP, LLE and Laplacian Eigenmap. In the following, we present a new dimensionality reduction algorithm based on the platform to avoid the limitations of LDA in data distribution assumption and available projection directions.

3.1. Marginal Fisher Analysis

LDA is motivated from the assumption that the data of each class is Gaussian distributed, which can not always be satisfied in real world problems. Moreover, the inter-class scatter can not well characterize the separability of the different classes of data without the Gaussian distribution assumption. A proper way to overcome the limitations of LDA is to design new criterion that characterizes the intra-class compactness and the inter-class separability. To this end, we propose a novel algorithm, called *Marginal Fisher Analysis* (MFA), in which the intra-class compactness is represented as the sum of distances between each point and its k_1-nearest neighbors of the same class; while the separability of different classes is characterized as the sum of distances between the margin points and their neighboring points of different classes. The adjacency graphs for *Marginal Fisher Analysis* are plotted in Figure 3. In this figure, Graph 1 illustrates the intra-class graph adjacency relationship, *i.e.* each sample is connected to its k_1-nearest neighbors, and Graph 2 connects the margin samples.

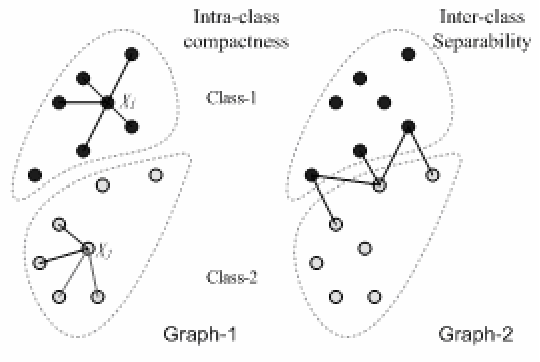


Figure 3: The two adjacency graphs for Marginal Fisher Analysis. Note that the left adjacency graph only plots the connection edges for one sample in each class for ease of understanding.

The intra-class compactness is characterized by

$$\begin{aligned}\tilde{S}_c &= \sum_i \sum_{i \in N_{k_1}^+(j) \, or \, j \in N_{k_1}^+(i)} \| w^{\mathrm{T}} x_i - w^{\mathrm{T}} x_j \|^2 \\ &= 2w^{\mathrm{T}} X (D^c - W^c) X^{\mathrm{T}} w \end{aligned} \tag{8}$$

$$W_{ij}^c = 1 \quad if\ j \in N_{k_1}^+(i)\ or\ i \in N_{k_1}^+(j);\ 0, else.$$

where $N_{k_1}^+(i)$ means the k_1 nearest neighbors in the same class of the sample x_i.

The inter-class separability is characterized by

$$\begin{aligned}\tilde{S}_m &= \sum_i \sum_{(i,j) \in P_{k_2}(l_i) \, or \, (j,i) \in P_{k_2}(l_j)} \| w^{\mathrm{T}} x_i - w^{\mathrm{T}} x_j \|^2 \\ &= 2w^{\mathrm{T}} X (D^m - W^m) X^{\mathrm{T}} w \end{aligned} \tag{9}$$

$$W_{ij}^m = 1\ \ if\ (i,j) \in P_{k_2}(l_i)\ or\ (j,i) \in P_{k_2}(l_j);\ 0, else.$$

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

The algorithmic procedure of *Marginal Fisher Analysis* is formally stated below:

1. **PCA projection**: We project the data set into the PCA subspace by retaining $N - N_c$ dimensions. Let W_{PCA} denote the transformation matrix of PCA.
2. **Constructing the intra-class compactness and inter-class separability graphs**: In the intra-class compactness graph, for each sample x_i, set the adjacency matrix $W_{ij}^c = W_{ji}^c = 1$ if x_j is among the k_1-nearest neighbors of x_i of the same class. In the inter-class separability graph, for each class *c*, set the similarity matrix $W_{ij}^m = W_{ji}^m = 1$ 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**: MFA aims at finding the optimal projection direction that optimizes the Marginal Fisher Criterion
$$w^* = \arg\min_w \frac{w^{\mathrm{T}} X(D^c - W^c) X^{\mathrm{T}} w}{w^{\mathrm{T}} X(D^m - W^m) X^{\mathrm{T}} w}$$
which is special linear graph embedding of (2) with
$$W = W^c, B = D^m - W^m$$
4. **Output** the final linear projection direction as
$$w = W_{PCA} w^* \qquad \blacksquare$$

In comparison with LDA, MFA has the following characteristics: 1) the available projection directions is much higher than that of LDA and the dimension number is determined by k_2, *i.e.* the selected number of shortest pairs of in-class and out-class samples; and 2) there is no assumption on the data distribution of each class and the intra-class compactness is characterized by the sum of the distance between each data and its k-nearest neighbors of the same class, thus it is more general for discriminant analysis; and 3) without the prior information on data distributions, the inter-class margin can better characterize the separability of different classes than the inter-class variance as in LDA.

3.2. Kernel Marginal Fisher Analysis

Kernel trick is widely used to enhance the separability of the linear supervised dimensionality reduction algorithms. The Marginal Fisher Analysis can be further improved by using the kernel trick.

Assume that the kernel function $k(x,y) = \phi(x) \cdot \phi(y)$ is applied, and the kernel Gram matrix is K with $K_{ij} = k(x_i, x_j)$. Let $w = \sum_{i=1}^{N} \alpha_i \phi(x_i)$, then optimal α can be obtained as

$$\alpha^* = \arg\min_\alpha \frac{\alpha^{\mathrm{T}} K(D^c - W^c) K\alpha}{\alpha^{\mathrm{T}} K(D^m - W^m) K\alpha} \tag{10}$$

Note that the graphs for Kernel Marginal Fisher Analysis (KMFA) may be different from the marginal fisher analysis as the k_1-nearest neighbors for each data in KMFA may be different from that in MFA. Thus the k_1 nearest neighbors for each data and the k_2 shortest pairs of in-class and out-class samples 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

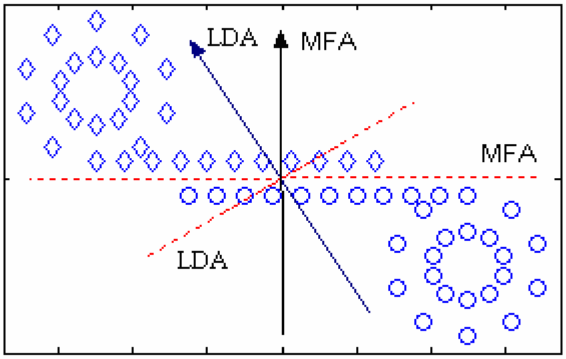


Figure 4. A Toy Problem: the comparative optimal projections from Marginal Fisher Criterion and LDA. Note that the solid line and dashed line represent the optimal projection direction and optimal classification hyperline, respectively.

$$D(x_i, x_j) = k(x_i, x_i) + k(x_j, x_j) - 2k(x_i, x_j)$$

For a new data x, its projection to the obtained optimal direction is obtained as
$$F(x, \alpha^*) = \tfrac{1}{\lambda} \sum_{i=1}^{N} \alpha_i^* k(x, x_i),$$
$$\text{where} \quad \lambda = (\alpha^{*}{}' K \alpha^*)^{1/2} \tag{11}$$

4. Experiments

In this section, we systematically evaluate the proposed Marginal Fisher Analysis (MFA) compared with LDA in both artificial and real world data. On the toy problem, we have compared their performance in deriving the optimal discriminant direction when the data is in non-Gaussian distribution. Two benchmark databases, ORL [11] and CMU PIE [13] have been used to evaluate the separability of the lower dimensional representation derived from MFA in comparison to LDA; also we have compared KMFC with KDA in the face recognition problem. All the experiments apply nearest neighbor approach as the final classifier.

4.1. A Toy Problem

In the toy problem, a two-class problem is discussed. The data for each class is non-Gaussian distributed as shown in Figure 4. The solid lines in Figure 4 represent the derived optimal projection directions for MFA and 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; while MFA successfully derived the most discriminating direction.

4.2. Face Recognition Problem

We have used the ORL and CMU PIE databases for face recognition to evaluate our proposed MFA and

KMFA algorithms. The ORL database contains 400 images of 40 individuals. Some images were captured at different times and have different variations including expression and facial details. In our experiments, all images are in grey scale and normalized to a resolution of 56*46 pixels by fixing the locations of two eyes. Histogram equilibrium was applied as the preprocessing step.

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 sub-databases are chosen for the evaluation of our proposed algorithm. In the first sub-database, referred as PIE-1, five near frontal poses (C27, C05, C29, C09 and C07) and illumination indexed as 08 and 11 are used, so each person has ten images. Another sub-database PIE-2 consists of the same five poses as in PIE-1, but the illumination indexed as 10 and 13 are also used, so each person has twenty images. All the images were aligned by fixing the locations of two eyes, and the images are resized to 64*64 pixels. Histogram equilibrium was applied as the preprocessing step similar to in the ORL database.

In each set of experiments, the image set is partitioned into the gallery and probe set with different numbers. For ease of representation, Gm/Pn means the *m* images per person are *randomly* selected for training and the remaining *n* images are for testing.

4.2.1 MFA vs. Fisherface

In this subsection, we evaluate the performance of the MFA compared with Fisherface [1] algorithm. For both algorithms, we retain $N - N_c$ dimension in the PCA step. For Fisherface, the final reduced dimension after LDA is $N_c - 1$ in [1]. To compare with the Fisherface fairly, similar to MFA, we explore the performance on all the feature dimensions and report the best result, which is referred to as *PCA+LDA* in all result tables. The experiments are conducted on both ORL and PIE-2 sub-database. The final results listed in Table 2 and 3 demonstrate that our proposed MFA consistently outperforms Fisherface and PCA+LDA. They also show that the best result in Fisherface is not in the dimension $N_c - 1$ and PCA+LDA improves the final accuracies significantly in most cases.

Table 2. Recognition accuracies (%) compared between MFA and Fisherface, PCA+LDA on the ORL database

	G3/P7	G4/P6	G5/P5
Fisherface	86.4	87.9	94.0
PCA+LDA	87.9	88.3	94.0
MFA	**89.3**	**91.3**	**96.0**

Table 3. Recognition accuracies (%) compared between MFA and Fisherface, PCA+LDA on the PIE-2 sub-database

	G4/P16	G5/P15	G6/P14
Fisherface	*71.1*	80.2	82.2
PCA+LDA	*76.9*	83.6	88.9
MFA	***82.5***	**87.3**	**90.5**

4.2.2 KMFA vs. KDA

In this subsection, we systematically evaluate the Kernel Marginal Fisher Analysis compared with the traditional Kernel Discriminant Analysis algorithm. In all the experiments, the Gaussian Kernel was used and parameter δ is set as $\delta = 2^{(n-10)/2.5}\delta_0, n = 1,...,20$, where δ_0 is the standard derivation of the data set. And the reported result is the best one among the 20 experiments. From the results listed in Table 4, we observe that: 1) the kernel trick improve the face recognition accuracy in both KDA and KMFA algorithms; and 2) KMFA consistently outperforms the other four algorithms in all the experiments.

Table 4. Face recognition accuracies (%) compared between MFA, KMFA and Fisherface, PCA+LDA, KDA on the PIE-1 sub-database and ORL database

	PIE-G4/P6	PIE-G3/P7	ORL-G4/P6
Fisherface	79.9	65.3	87.9
PCA+LDA	80.2	65.8	88.3
MFA	**84.9**	**71.0**	**91.3**
KDA	81.0	70.0	91.7
KMFA	**85.2**	**72.3**	**93.8**

5. Conclusion

The work presented in this paper has given an insight to the relationship among the state-of-the-art dimensionality reduction algorithms. A general framework, graph embedding along with its linearization and kernelization, has been proposed to provide a unified perspective to understand most traditional dimensionality reduction algorithms. Moreover, it can also be used as a platform to develop new algorithm for dimensionality reduction. For this, we have proposed a novel dimensionality reduction algorithm, called Marginal Fisher Analysis (MFA), by designing two graphs that characterize the intra-class compactness and the inter-class separability, respectively. This new algorithm effectively overcome the limitations of the traditional LDA algorithm in data distribution assumption and is a more general algorithm for discriminant analysis. The toy problem and the face recognition experiments have demonstrated the superiority of our proposed MFA and its kernelization compared with traditional Fisherface and KDA.

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*, vol. 19, No. 7, 1997, pp. 711-720.

[2] M. Belkin and P. Niyogi, “Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering”, *Advances in Neural Information Processing System* 15, Vancouver, British Columbia, Canada, 2001.

[3] F. Chung. “Spectral Graph Theory”, *Regional Conferences Series in Mathematics*, Number 92, 1997.

[4] K. Fukunnaga, “Introduction to Statistical Pattern Recognition”, Academic Press, second edition, 1991.

[5] D. Hand. “Kernel Discriminant Analysis”. Research Studies Press, Chichester, 1982

[6] X. He, P. Niyogi. "Locality Preserving Projections (LPP)". TR-2002-09, 29 October, 2002.

[7] I. Joliffe. “Principal Component Analysis”. Springer-Verlag, New York, 1986.

[8] J. Ham, D. Lee, S. Mika and B. Schölkopf: “A kernel view of the dimensionality reduction of manifolds”, in the Proceedings of ICML 2004.

[9] Martinez and A. Kak. "PCA versus LDA". IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 2, pp. 228-233, 2001.

[10] K. Mtiller, S. Mika, G. Riitsch,, K. Tsuda, B. Sch61kopf, "An Introduction to kernel-based learning algorithms", IEEE Transactions on Neural Networks, Vol. 12, pp.181 201, 2001.

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

[12] S. Roweis, and L. Saul, “Nonlinear Dimensionality Reduction by Locally Linear Embedding”, *Science*, vol 290, 22 December 2000.

[13] T. Sim, S. Baker, and M. Bsat. “The CMU Pose, Illumination, and Expression (PIE) Database of Human Faces”. Tech. Report CMU-RI-TR-01-02, Robotics Institute, Carnegie Mellon University, January, 2001.

[14] J. Tenenbaum, V. Silva, and J. Langford, “A Global Geometric Framework for Nonlinear Dimensionality Reduction”,*Science*, vol 290, 22 December 2000.

[15] Y. Teh and S. Roweis, “Automatic Alignment of Hidden Representations”, *Advances in Neural Information Processing System* 15 (NIPS'02). pp. 841-848.

[16] M. Turk and A. Pentland. “Face Recognition Using Eigenfaces”, *IEEE Conference on Computer Vision and Pattern Recognition*, Maui, Hawaii, 1991.

[17] X. Wang and X. Tang, “Dual-space linear discriminant analysis for face recognition,” Proceedings of CVPR’04, vol. 2, pp. 564–569, 2004.

[18] S. Yan, H. Zhang, Y. Hu, B. Zhang and Q. Cheng. "Discriminant Analysis on Embedded Manifold". Proceeding of The 8th European Conference on Computer Vision - ECCV 2004, Prague, May 11-14, 2004.

[19] 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.

Appendix

A. The ISOMAP algorithm can be reformulated as the direct graph embedding formulation in Eq. (2) with similarity matrix as $W_{ij} = \tau(D_G)_{ij}$, $i \neq j$; else 0. $B = I$.

Proof: As the matrix $\tau(D_G) = -HSH/2$, we have

$$\begin{aligned}\sum\nolimits_j \tau(D_G)_{ij} &= \sum\nolimits_j (-HSH/2)_{ij} \\ &= \sum\nolimits_j (-(I - \tfrac{1}{N}ee^{\mathrm{T}})S(I - \tfrac{1}{N}ee^{\mathrm{T}})/2)_{ij} \\ &= \tfrac{1}{2}\sum\nolimits_j (-S_{ij} + \tfrac{1}{N}\sum\nolimits_{i'} S_{i'j} + \tfrac{1}{N}\sum\nolimits_{j'} (S_{ij'} - \tfrac{1}{N}\sum\nolimits_{i'} S_{i'j'})) \\ &= (-\tfrac{1}{2}\sum\nolimits_j S_{ij} + \tfrac{1}{2N}\sum\nolimits_{jj'} S_{ij'}) + (\tfrac{1}{2N}\sum\nolimits_{ji'} S_{i'j} - \tfrac{1}{2N^2}\sum\nolimits_{jj'i'} S_{i'j'}) \\ &= 0 \qquad \forall i\end{aligned}$$

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

$$\begin{aligned}y^* &= arg \max_{y^{\mathrm{T}}y=\lambda} y^{\mathrm{T}}\tau(D_G)y = arg \max_{y'y=\lambda} y^{\mathrm{T}}(-D+W)y \\ &= arg \min_{y^{\mathrm{T}}y=\lambda} y^{\mathrm{T}}(D-W)y\end{aligned}$$

Note that c is the corresponding eigenvalue of $\tau(D_G)$, which is different from the other algorithms in which c is mostly set as 1. Therefore, we can conclude that ISOMAP algorithm can be reformulated in the graph embedding formulation in (2). ■

B. The LLE algorithm can be reformulated as the direct graph embedding formulation in Eq. (2) with similarity matrix $W_{ij} = (M + M^{\mathrm{T}} - M^{\mathrm{T}}M)_{ij}, i \neq j;$ else $0. B = I$.

Proof: With simple algebra computation, we have [2]

$$\sum\nolimits_i \sum\nolimits_j \| y_i - M_{ij}y_j \|^2 = y^{\mathrm{T}}(I - M^{\mathrm{T}})(I - M)y$$

On the other hand, $\sum\nolimits_j M_{ij} = 1 \ \ \forall i$ [2]; thus

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

Therefore, the matrices $(I - M^{\mathrm{T}})(I - M)$ can be considered as Laplacian matrix of a graph.

If $W_{ij} = (M + M^{\mathrm{T}} - M^{\mathrm{T}}M)_{ij}, i \neq j;$ else $0. B = I$, then

$$\begin{aligned}y^* &= arg \min_{y^{\mathrm{T}}y=1} y^{\mathrm{T}}(I - M^{\mathrm{T}})(I - M))y \\ &= arg \min_{y^{\mathrm{T}}y=1} y^{\mathrm{T}}(D - W)y\end{aligned}$$

That is, LLE algorithm can be reformulated as the direct graph embedding formulation as in Eq. (2) ■