A Parameter-Free Framework for General Supervised Subspace Learning

Shuicheng Yan, *Member, IEEE*, Jianzhuang Liu, *Senior Member, IEEE*, Xiaoou Tang, *Senior Member, IEEE*, and Thomas S. Huang, *Life Fellow, IEEE*

***Abstract*—Supervised subspace learning techniques have been extensively studied in biometrics literature; however, there is little work dedicated to: 1) how to automatically determine the subspace dimension in the context of supervised learning, and 2) how to explicitly guarantee the classification performance on a training set. In this paper, by following our previous work on unified subspace learning framework in our earlier work, we present a general framework, called parameter-free graph embedding (PFGE) to solve the above two problems by posing a general supervised subspace learning task as a semidefinite programming problem. The semipositive feature Gram matrix, namely the product of the transformation matrix and its transpose, is derived by optimizing a trace difference form of an objective function extended from that in our earlier work with the constraints that guarantee the class homogeneity within the neighborhood of each datum. Then, the subspace dimension and the feature weights are simultaneously obtained via the singular value decomposition of the feature Gram matrix. In addition, to alleviate the computational complexity, the Kronecker product approximation of the feature Gram matrix is proposed by taking advantage of the essential matrix form of image pixels. The experiments on simulated data and real-world data demonstrate the capability of the new PFGE framework in estimating the subspace dimension for supervised learning as well as the superiority in classification performance over traditional algorithms for subspace learning.**

***Index Terms*—Semidefinite programming, subspace dimension determination, subspace learning.**

I. Introduction

TECHNIQUES for subspace learning [10], [17], [31], [23], [28] have been actively studied for decades. Most of them, such as principal component analysis (PCA) [12], [22], linear discriminant analysis (LDA) [2], [9], [31], and marginal fisher analysis (MFA) [29], are solved with the spectral-analysis [6], [9] methods. The supervised techniques often optimize objective functions characterizing the discriminative power in the sense of expectation or with certain assumptions on data distribution, and cannot ensure that the training samples are best classified with the nearest neighbor method in an obtained low-dimensional feature space, especially when the number of training samples is small.

How to automatically determine the dimension of the desired low-dimensional feature space is seldom discussed in previous algorithms for supervised dimensionality reduction. Hence, the dimension is often intuitively set, or all possible subspace dimensions are explored in order to obtain the optimal one for classification, which is impractical and easily overfits the specific testing data. In the literature of unsupervised learning, intrinsic data dimension estimation [13], [16], [11] has been widely discussed in past decades. Kegl [13] utilized the geometric properties of the data to estimate the intrinsic data dimension in a nonparametric way. Hu [11] studied the automatic subspace dimension determination under the framework of Bayesian Ying–Yang (BYY) harmony learning. Lin *et al.* [16] estimated the intrinsic data dimension by constructing a Riemannian manifold in the form of a simplicial complex, and the dimension is defined as the maximal dimension of its simplices. Brito *et al.* [4] treated as a random variable the average reach of vertices in a k-nearest-neighbors graph associated with the interpoint distance matrix, and showed that this variable can be used to accurately (from a probabilistic viewpoint) identify the unknown dimension at low computational cost. Brito [5] discussed the application of linear combinations of the degree frequencies in the minimal spanning tree to the problem of identifying the appropriate dimension for a data set from its interpoint distance matrix. Costa [7] and Yang [30] studied the data dimension estimation problem by using trees to approximate manifold structures. All of these methods focus on unsupervised learning, and do not utilize the information of data-class labels that are available in supervised subspace learning.

Motivated by the above observations, we present a parameter-free framework for general supervised subspace learning by following our previous work on graph embedding as a unified framework for subspace learning [29]. The new framework searches for a low-dimensional feature space where the neighboring points of each datum share the same class label, which is optimal in the sense of the nearest neighbor classification.

The whole framework, referred to as parameter-free graph embedding (PFGE), consists of the following steps. First, instead of directly computing the transformation matrix for dimensionality reduction, we search for the feature Gram matrix (i.e., the product of the transformation matrix and its transpose). Then, the ratio form of the objective function in the graph embedding framework [29] is transformed into a difference form in PFGE. After that, the feature Gram matrix is learned by posing the supervised subspace learning

Manuscript received May 22, 2006; revised September 19, 2006. This work was supported in part by DTO under Contract NBCHC060160 and in part by the Research Grants Council of the Hong Kong SAR under Project CUHK 414306. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Vijaya Kumar Bhagavatula

S. Yan and T. S. Huang are with the Beckman Institute, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: scyan@ifp.uiuc.edu; huang@ifp.uiuc.edu).

J. Liu and X. Tang are with the Department of Information Engineering, the Chinese University of Hong Kong, Shatin, NT, Hong Kong, China (e-mail: jzliu@ie.cuhk.edu.hk; xtang@ie.cuhk.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2006.890313

problem as a semidefinite programming problem [26], [27], [25], with the constraints of neighborhood homogeneity and scale normalization. The neighborhood homogeneity property guarantees the classification accuracy on the training set and is expressed as a set of linear inequalities with the variables from the feature Gram matrix. Finally, the optimal transformation matrix encoding the projection directions and their weights is obtained from the singular value decomposition of the derived feature Gram matrix.

It is worthy of highlighting some aspects of our proposed PFGE framework.

1) PFGE guarantees the classification accuracy on the training set by the hard constraints of the semidefinite programming problem. It is different from traditional subspace learning methods, which are often irrelevant to the classification accuracy, such as PCA, or the objective functions of which do not directly optimize the classification accuracy such as LDA and MFA.
2) PFGE can automatically determine the optimal feature dimension by retaining the positive components of the feature Gram matrix. In traditional subspace learning methods, no criterion is provided to guide the selection of the feature dimension.
3) PFGE finds a transformation matrix, instead of a projection matrix, for dimensionality reduction. It simultaneously selects the projection directions and determines the weights to fuse them, which directly leads to its superiority over traditional algorithms for subspace learning in terms of classification capability.

The rest of the paper is structured as follows. The graph embedding framework is reviewed and then the PFGE framework is introduced in Section II. In Section III, the Kronecker product approximation of the feature Gram matrix is discussed. In Section IV, experiments on some toy problems and on face recognition are showed to demonstrate the effectiveness of our framework. Finally, we give concluding remarks in Section V.

II. Parameter-Free Graph Embedding

Let $X = [x_1, x_2, \ldots, x_N]$ with $x_i \in \mathbb{R}^m$ be a set of N sample points where the corresponding class labels are denoted as $\{c_i | c_i \in \{1, \ldots, N_c\}\}_{i=1}^N$. Denote the sample number of the cth class as n_c. Since in practice the dimension m is often very large, it is usually necessary to transform the data from the input high-dimensional space to a low-dimensional one for alleviating the curse of dimensionality [9].

A. Review of Graph Embedding [29]

Let $G = \{X, W\}$ be an undirected similarity graph, called an intrinsic graph, with the vertex set X and the similarity matrix $W \in \mathbb{R}^{N \times N}$ which characterizes the similarities among all sample pairs. The corresponding diagonal matrix D and the Laplacian matrix L [1] are defined as

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

The purpose of graph embedding [29] is to determine a low-dimensional representation of the vertex set X that preserves the similarities between pairs of data characterized in the original high-dimensional space. Denote the low-dimensional embedding of the vertices as $Y = [y_1, y_2, \ldots, y_N]$, where the column vector y_i is the embedding for the vertex x_i. Direct graph embedding [29] aims to maintain similarities among vertex pairs by following the graph preserving criterion [29]:

$$\begin{aligned} Y^* &= \arg \min_{\mathrm{Tr}(YBY^T)=c} \sum_{i \neq j} \| y_i - y_j \|^2 W_{ij} \\ &= \arg \min_{\mathrm{Tr}(YBY^T)=c} \mathrm{Tr}(YLY^T) \end{aligned} \tag{2}$$

where c is a constant, $\mathrm{Tr}(\cdot)$ is the trace of an arbitrary square matrix, and B is called the constraint matrix here. B may simply be a diagonal matrix and used for scale normalization [29], or may represent more general constraints among the vertices characterized by a penalty graph G' [29]. The penalty graph describes similarities between nodes that are unfavorable and should be avoided (i.e., $B = L^p = D^p - W^p$) where W^p is the similarity matrix of graph G', and B or L^p is the Laplacian matrix of G', defined similarly as (1).

The similarity preservation property of the graph preserving criterion works in two ways. If the similarity between samples x_i and x_j is greater (positive), then the distance between y_i and y_j should be smaller to minimize (2); on the other hand, if the similarity between x_i and x_j is lower (negative), the distance between y_i and y_j should instead be larger. Hence, the similarities and differences among vertex pairs in a graph G are preserved in the embedding.

As shown in the embedding framework [29], (2) can be solved by converting it into the following Trace Ratio problem:

$$Y^* = \arg \min_Y \frac{\mathrm{Tr}(YLY^T)}{\mathrm{Tr}(YBY^T)}. \tag{3}$$

If the constraint matrix represents only scale normalization, this ratio formulation can be directly solved by eigenvalue decomposition [6]. However, for a more general constraint matrix, it can only be approximately solved with generalized eigenvalue decomposition (GED) by transforming the objective function into a more tractable approximate form $\max_Y \mathrm{Tr}[(YLY^T)^{-1}(YBY^T)]$ or $\max_Y \mathrm{Tr}[Y(B - L)Y^T]$, referred to as ratio trace or trace difference in the remainder of this paper.

While direct graph embedding computes a low-dimensional representation of the vertices in X, it does not determine how new out-of-sample high-dimensional data can be mapped to the low-dimensional space. To this end, linearization and kernelization as described in [29] are needed. Let us take linearization as an example. Assume that the low-dimensional vector representations of the vertices can be obtained from the linear projection $y_i = P^T x_i$, where $P = [p_1, p_2, \ldots, p_d] \in \mathbb{R}^{m \times d}$ is the projection matrix that we want to find for mapping out-of-sample data, and d is the expected dimension after dimensionality reduction. Then, the objective function (2) is changed to

$$P^* = \arg \min_P \frac{\mathrm{Tr}(P^T XLX^T P)}{\mathrm{Tr}(P^T XBX^T P)}. \tag{4}$$

In practical applications, the determination of the number d of the columns in P is significant for achieving the best classification capability on the low-dimensional feature space. In the following, we present a new framework in the context of supervised subspace learning that can automatically derive the optimal subspace dimension d.

B. Feature Gram Matrix Learning for PFGE

As described before, the direct learning of the transformation matrix for dimensionality reduction results in the feature dimension selection issue. In this work, instead of directly computing the transformation matrix P, we learn the feature Gram matrix S which characterizes the similarity of different features and is defined as

$$S = PP^T. \tag{5}$$

We search for S by solving a semidefinite programming problem [3], [26].

1) Neighborhood Homogeneity Constraints: When there is no assumption on the data distribution, it is desirable that the neighboring samples of each point share the same class label, which is optimal in the sense of classification with the nearest neighbor method. Then, we have

$$\|P^T x_i - P^T x_{i\#}\|^2 \leq \|P^T x_i - P^T x_j\|^2, \quad c_i \neq c_j \tag{6}$$

where $x_{i\#}$ is the nearest neighbor of x_i belonging to the same class, measured with the Euclidean distance in the original feature space. Equation (6) can be expressed in another form

$$\begin{aligned}\mathrm{Tr}\{[(x_i - x_{i\#})(x_i - x_{i\#})^T - (x_i - x_j)(x_i - x_j)^T]S\} \\ = \mathrm{Tr}(A_{ij}S) \leq 0\end{aligned} \tag{7}$$

where $A_{ij} = (x_i - x_{i\#})(x_i - x_{i\#})^T - (x_i - x_j)(x_i - x_j)^T$. It is easy to verify that the matrix A_{ij} is symmetric.

2) Scale Normalization: To remove the degree of freedom of scale, we may constrain the sum of the column norms of the transformation matrix P to be 1, that is

$$\mathrm{Tr}(P^T P) = \mathrm{Tr}(PP^T) = \mathrm{Tr}(S) = 1. \tag{8}$$

3) Objective Function: As mentioned before, the objective function of graph embedding is often changed to the ratio trace or trace difference form for a more tractable solution. In this work, we utilize the trace difference formulation of graph embedding for semidefinite programming formulation. The objective function in this new formulation is defined as

$$\mathrm{Tr}[P^T X(L - B)X^T P] = \mathrm{Tr}[X(L - B)X^T S]. \tag{9}$$

With the above constraints and the objective function, the feature Gram matrix S can be obtained by optimizing the semidefinite programming problem given in Algorithm 1. The object function in Algorithm 1 is convex and the optimization does not suffer from the local optimum problem [26], [27], [25]. There are several general-purpose toolboxes and polynomial time solvers available for the semidefinite programming problem. In this paper, we utilize the solver SeDuMi and CSDP 4.9 toolbox in MATLAB [3].

Algorithm 1 Direct Feature Gram Matrix Learning for PFGE

Minimize $\mathrm{Tr}[X(L - B)X^T S]$

1) $S \succeq 0$;

2) $\mathrm{Tr}(S) = 1$;

3) $\mathrm{Tr}(A_{ij}S) \leq 0$, $c_i \neq c_j$.

C. Transformation Matrix From the Feature Gram Matrix

After obtaining the feature Gram matrix S by using the semidefinite programming approach, we can derive the transformation matrix P by preserving the positive components of the feature Gram matrix, which is similar to the multidimensional scaling (MDS) algorithm [8]. The singular value decomposition of S results in

$$S = \sum_{k=1}^{m} \lambda_k v_k v_k^T \tag{10}$$

where λ_k is the kth largest eigenvalue of S with the corresponding eigenvector v_k. Then, we have

$$P = [\lambda_1^{1/2} v_1, \lambda_2^{1/2} v_2, \ldots, \lambda_d^{1/2} v_d], \quad \lambda_d > \lambda_{d+1} = 0. \tag{11}$$

In practice, noise probably exists in the data, and there may be some λ_i of very small values. Hence, we only keep the dimensions that preserve sufficient information of the matrix S as PCA does; we retain 98% energy in all of the experiments.

III. Kronecker Production Approximation

When the original feature dimension m is too large, Algorithm 1 is impractical in both computation and memory requirements. As the constraint matrix A_{ij} is not sparse, we can mostly only handle the cases with less than 400 features. In this section, we discuss how to solve this computational problem.

A. Kronecker Production Approximation of the Feature Gram Matrix

In the real world, the extracted features of an object often have some special structures, and these structures are in the form of second- or even higher order tensors. For example, a captured image is a second-order tensor (i.e., a matrix), and a video sequence is in the form of a third-order tensor. It would be desirable to uncover the underlying structures in the problems for data analysis. In the following, we investigate how to utilize the latent data structure to solve the computational problem suffered by the PFGE framework.

Algorithm 2 Feature Gram Matrix Learning from Kronecker Approximation for PFGE

Minimize $\mathrm{Tr}[X(L - B)X^T(S_2 \otimes S_1)]$

1) $S_1 \succeq 0$, $S_2 \succeq 0$;

2) $\mathrm{Tr}(S_2 \otimes S_1) = 1$;

3) $\mathrm{Tr}[A_{ij}(S_2 \otimes S_1)] \leq 0$, $c_i \neq c_j$.

Assume that the training samples are denoted as second-order tensors (matrices) $\{X_i \in \mathbb{R}^{m_1 \times m_2}\}_{i=1}^{N}$ and its corresponding column-wise concatenated vector is still represented as $\{x_i | x_i \in \mathbb{R}^m\}_{i=1}^{N}$ with $m = m_1 \times m_2$. Assume that the low-dimensional representation of X_i is obtained from two transformation matrices P_1 and P_2. Then

$$Y_i = P_1^T X_i P_2, \ P_1 \in \mathbb{R}^{m_1 \times d_1}, \ P_2 \in \mathbb{R}^{m_2 \times d_2}. \tag{12}$$

It means that the transformation matrix P is approximated as the Kronecker product of two transformation matrices $P = P_2 \otimes P_1$, where $\otimes$ is the Kronecker product with $A \otimes B = [A_{ij}B]$.

Then, we have

$$y_i = \mathrm{vec}(Y_i) = P^T \mathrm{vec}(X_i) = (P_2 \otimes P_1)^T x_i. \tag{13}$$

where $\mathrm{vec}(\cdot)$ means the vectorization of a matrix by column-wise connecting all of the elements. Consequently, it follows that the feature Gram matrix $S = PP^T = (P_2 \otimes P_1)(P_2 \otimes P_1)^T = S_2 \otimes S_1$ with $S_1 = P_1 \otimes P_1^T$ and $S_2 = P_2 \otimes P_2^T$. Based on the above kronecker product approximation, the PFGE framework is modified as follows.

1) Neighborhood Homogeneity Constraints: The hard constraint on neighborhood homogeneity in (6) is changed to

$$\begin{aligned} &\|P_1^T(X_i - X_{i\#})P_2\|^2 \le \|P_1^T(X_i - X_j)P_2\|^2, \ c_i \neq c_j \\ &\Rightarrow \mathrm{Tr}[A_{ij}(S_2 \otimes S_1)] \le 0, \ c_i \neq c_j. \end{aligned} \tag{14}$$

2) Scale Normalization: To remove the degree of freedom of scale, we can constrain the sum of the column norms of the transformation matrix to be 1, that is

$$\mathrm{Tr}(S) = \mathrm{Tr}(S_2 \otimes S_1) = 1. \tag{15}$$

3) Objective Function: From the objective function in (9), we have

$$\mathrm{Tr}[X(L-B)X^T S] = \mathrm{Tr}[X(L-B)X^T(S_2 \otimes S_1)]. \tag{16}$$

With the above constraint and the objective function, we can have the Kronecker product approximation-based algorithm for PFGE as shown in Algorithm 2.

The optimization problem formulated in Algorithm 2 is not a standard semidefinite programming problem, and commonly nonconvex. However, when S_1 or S_2 is known, the optimization problem with respect to the other one is a standard semidefinite programming problem and can be solved by using the semidefinite programming toolbox as in Algorithm 1. Therefore, we can optimize these two matrices in an iterative manner until convergence is reached. Finally, the transformation matrices P_1 and P_2 can be derived from the singular value decompositions of S_1 and S_2.

Based on the Kronecker product approximation of the feature Gram matrix, the optimization problem defined in each step runs on a much lower dimensional feature space. For example, when the image matrix is of size 100×100, the parameter number in the previous formulation is 10^8, making it impractical for Algorithm 1 to run on a PC with a common configuration. However, in the Kronecker product approximation version, there are only 10^4 parameters, which greatly alleviates the computational complexity of the optimization problem.

4) Discussions: To further reduce the complexity of the optimization problem, in Algorithm 1 or 2, we do not use all of the constraints for optimization in the beginning. In the first optimization step, for each sample, we only use the constraint from the nearest samples of different classes. After each optimization step, we add a certain percentage of the sample pairs that do not satisfy the neighborhood homogeneity constraints and then optimize again until all of the constraints are satisfied. Moreover, it is possible that not all of the constraints must be satisfied. Thus, in our experiments, we relax the constraints by adding two relaxation parameters ζ_{ij} and ξ_{ij}, and modify Algorithm 1 to be Algorithm 3. More details on the use of the relaxation parameters can be found from [3].

The optimization problem in Algorithm 3 can also be solved by using the solver SeDuMi and CSDP 4.9 toolbox in MATLAB. Note that if all of the constraints can be satisfied, the value of λ will not affect the final results and, thus, even in this case, there is no parameter to select. Similarly, we can use Algorithm 3 to optimize the step optimization problem of Algorithm 2.

Algorithm 3 Feature Gram Matrix Learning with Relaxation for PFGE

Minimize $\mathrm{Tr}[X(L-B)X^T S] + \lambda \sum_{c_i \neq c_j} \xi_{ij}, \quad \lambda \gg 0.$

1) $S \succeq 0$.

2) $\mathrm{Tr}(S) = 1$.

3) $\mathrm{Tr}(A_{ij}S) + \zeta_{ij} - \xi_{ij} = 0, \xi_{ij} \ge 0, \zeta_{ij} \ge 0, c_i \neq c_j$.

Algorithmic Analysis

In this subsection, we discuss some characteristics of the proposed PFGE framework and its relationship with other state-of-the-art algorithms for dimensionality reduction.

5) Training Accuracy is Guaranteed by Hard Constraints: The classification accuracy on the training set is guaranteed with the hard constraints imposed on the semidefinite programming problem; while in most traditional algorithms of dimensionality reduction, the training accuracy is not their direct targets, especially for the unsupervised ones.

6) Feature Dimension and Fusing Weights are Automatically Determined: The dimension of the low-dimensional feature space is automatically determined by preserving the information of the feature Gram matrix in (11); while in the previous algorithms for dimensionality reduction, the subspace dimension can only be experimentally set, or it is needed to explore all of the possible dimensions and select an optimal one for a specific data set. Moreover, PFGE derives a transformation matrix, instead of a projection matrix, for dimensionality reduction, which automatically determines the weight (importance) of each feature.

7) Relationship With Distance Metric Learning for Large Margin Nearest Neighbor Classification (LMNN) [24]: LMNN and our PFGE both utilize the semidefinite programming tool to formulate and solve their problems, and both use nearest

neighbor classifiers to help design effective algorithms; yet, they are essentially different in several aspects.

1) The motivation of PFGE is to provide some general constraints for supervised subspace learning, such that the required subspace dimension for classification can be automatically determined and the classification accuracy on the training set is guaranteed. The former target is not studied in LMNN. Moreover, the constraints of LMNN cannot be directly used for general supervised subspace learning since they do not provide a constraint to bound the scale of the matrix S, and PFGE will not have solutions if these constraints are imposed on PFGE. The constraints in PFGE do not suffer from those problems and can be used for all supervised subspace learning algorithms.
2) PFGE is a general framework for supervised subspace learning, while LMNN is only a specifically designed algorithm. Similarly to the original graph embedding framework [29], PFGE can help develop new algorithms for subspace learning without suffering from the above-mentioned problems.
3) PFGE has been extended to the Kronecker production approximation form, which is applicable for common images, while LMNN is impractical even for images of moderate size 64×64 pixels.

8) Relationship With Maximum Margin Criterion [15]: The objective function of MMC shares the similar formulation as that of PFGE. However, MMC directly optimizes the map matrix P and cannot guarantee that the neighborhood homogeneity constraints are satisfied. Moreover, MMC cannot automatically determine the subspace dimension.

9) Convergence of the Kronecker Production Approximation: It is easy to prove that the solution space

$$\chi = \{(S_1, S_2) | S_1 \succeq 0, S_2 \succeq 0, \mathrm{Tr}(S_2 \otimes S_1) = 1, \mathrm{Tr}[A_{ij}(S_2 \otimes S_1)] \leq 0, c_i \neq c_j\}$$

is closed and bounded. On one hand, the objective function $\mathrm{Tr}[X(L-B)X^T(S_2 \otimes S_1)]$ is nonincreasing in each iteration. On the other hand, as χ is closed and bounded, the objective function is also bounded. Therefore, Algorithm 2 converges to a local optimum.

IV. Experiments

In this section, we present two sets of experiments to evaluate the effectiveness of the proposed PFGE framework by comparing it with the Eigenfaces [2] and Fisherfaces [2]. In order to compare with the Fisherfaces algorithm [2] fairly, we apply the intrinsic and penalty graphs from Fisherfaces as demonstrated in [29] to define a specific algorithm from the PFGE framework, and this specific algorithm is referred to as Fisherfaces–PFGE (F–PFGE) in the following. For the simulated data, we demonstrate the effectiveness of PFGE in determining the optimal feature dimension for classification and in satisfying the neighborhood homogeneity constraints. For the real face data, we evaluate the classification capability of the low-dimensional representations derived from the F–PFGE algorithm and from the traditional Eigenfaces, and Fisherfaces. In our experiments, we utilize the Kronecker product approximation version of F–PFGE to conduct dimensionality reduction for face recognition task since the vector-based version is impractical.

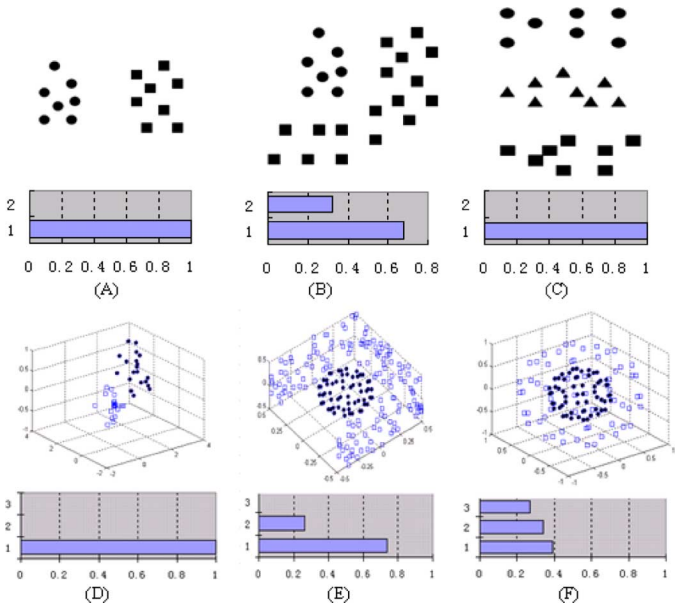


Fig. 1. Subspace dimensions automatically determined by F–PFGE. Note that from (A) to (F), the top plot is the multiclass data distributions, and the lower plot is the eigenvalue distribution computed from (11). For the lower plots, the horizontal axes denote the value of $\sqrt{\lambda_i}$ in (11), and the vertical axes denote the index number of λ_i.

A. Toy Problems

1) Automatic Subspace Dimension Determination: We present two series of experiments, one is to demonstrate the capability of F–PFGE in determining the optimal subspace dimension for classification. Fig. 1 displays a series of experimental results on the derived eigenvalues from different data sets. The results show that F–PFGE successfully uncovers the required feature dimension for classification. Traditional supervised dimensionality reduction algorithms such as LDA cannot automatically determine which dimension is optimal, and for the sample sets in (B) and (C), the LDA algorithm will output one and two ($N_c - 1$) dimensions, respectively [2]. Actually, the required dimensions of these two problems are two and one, respectively, which are uncovered by F–PFGE successfully.

2) Neighborhood Homogeneity: The other series of experiments is to evaluate the capability of F–PFGE in ensuring that the neighboring points of each point share the same class label. We use two sets of data, one is plotted in Fig. 2(A), and the derived low-dimensional representations are plotted in Fig. 2(B). In Fig. 2(B), each sample is connected to its nearest sample measured in the derived representations, and we can see that the nearest neighbor of each sample shares the same label as it does. The other data set is the digital numbers from the MNIST database [14]. We select digital numbers 0–3 and use 39 samples for each number. The data distribution in the derived low-dimensional representations is displayed in Fig. 2(C), and the results show that the nearest neighbor of each sample also has the same class label as the sample does.

B. Face Recognition

In this subsection, the F–PFGE algorithm is compared with Eigenfaces [22] and Fisherfaces [2], for face recognition on three benchmark databases XM2VTS [18], CMU PIE [21], and

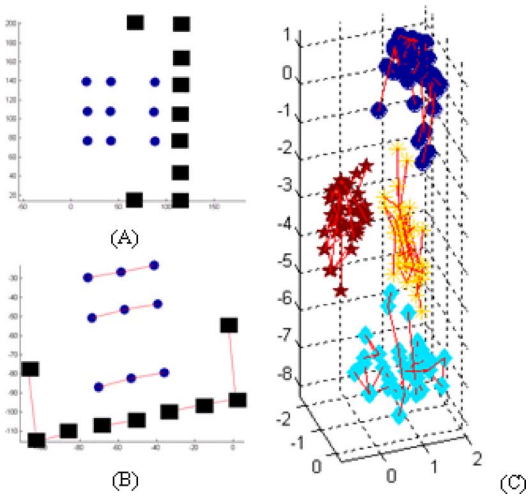


Fig. 2. Neighborhood homogeneity results. (a) Simulated data. (b) Rrepresentations derived from F–PFGE. (c) Low-dimensional representations of digital numbers obtained by F–PFGE.

Fig. 3. Cropped samples from the XM2VTS database.

TABLE I
BEST RECOGNITION ACCURACIES (%) COMPARED AMONG EIGENFACES, FISHERFACES, AND F–PFGE ON THE XM2VTS DATABASE

Algorithm	*Eigenfaces*	*Fisherfaces*	*F-PFGE*
Accuracy	69.8%	81.7%	86.4%
Training Time	≈4.01s	≈ 6.02s	≈ 601s

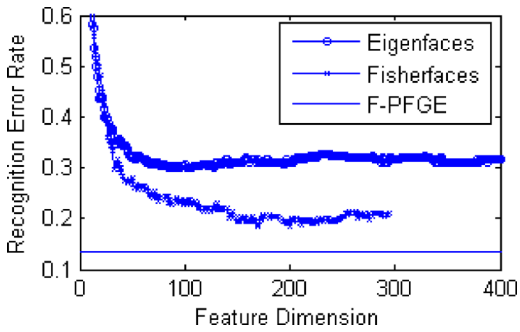


Fig. 4. Face recognition error rates versus feature dimension of Eigenfaces, Fisherfaces, and F–PFGE on the XM2VTS database.

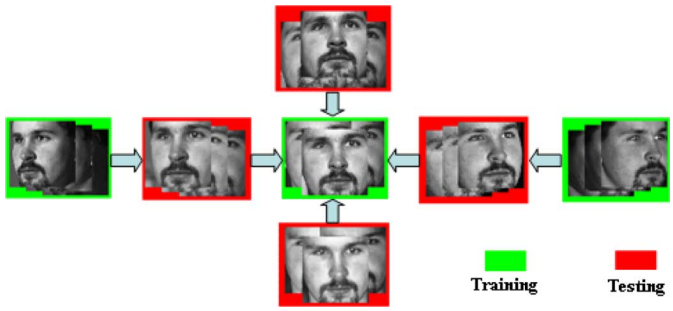


Fig. 5. Cropped samples from the CMU PIE database.

TABLE II
BEST RECOGNITION ACCURACIES (%) COMPARED AMONG EIGENFACES, FISHERFACES, AND F–PFGE ON THE PIE DATABASE

Algorithm	*Eigenfaces*	*Fisherfaces*	*F-PFGE*
Accuracy	69.4%	57.4%	72.8%
Training Time	≈4.11s	≈ 6.30s	≈ 620s

ORL [20]. In the experiments, the nearest neighbor method is used for final classification. For Fisherfaces, we first project the images into a PCA space of dimension $N - N_c$, and then explore all possible LDA dimensions and report the best results.

The XM2VTS database contains 295 persons 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. Some cropped samples are displayed in Fig. 3. Table I and Fig. 4 show the face recognition results of F–PFGE compared with Eigenfaces, and Fisherfaces. The results indicate that although there is no user-selectable parameter in F–PFGE, its performance is still better than Eigenfaces and Fisherfaces algorithms which need to explore all possible feature dimensions to obtain the best results. Note that since the result of F–PFGE is free from the final feature dimension, we plot it as a line in Fig. 4. The training times used by different algorithms are also given in Table I. From the results, we can see that the time complexity of F–PFGE is relatively higher than those of Eigenfaces and Fisherfaces.

The CMU Pose, Illumination, and Expression (PIE) database contains more than 40 000 facial images of 68 people. The images were acquired over different poses, under variable illumination conditions, and with different facial expressions. The gallery set and probe set are selected as in Fig. 5, and nine images of each person are used for training and 12 images of each person for testing. All of the images are aligned by fixing the locations of the eye centers and normalized to the size of 64 × 64 pixels, and 63 people are used in our experiments due to the data incompleteness of the other five people. Table II and Fig. 6 list the face recognition results of F–PFGE compared with Eigenfaces and Fisherfaces, which again show that F–PFGE is superior to the other two algorithms. An interesting observation is that the performance of Eigenfaces is even better than that of Fisherfaces, which has been reported in [19] and is caused by the data distribution inconsistence between the training and testing data.

The ORL database contains 400 images of 40 individuals. Some cropped sample images are displayed in Fig. 7. In the experiments, all of the images are in gray level and rescaled to the resolution of 56 × 46 pixels. Histogram equalization is applied as a preprocessing step. Half of the data are used for model training and the others are used for testing. The comparison results are given in Table III and Fig. 8, which show that the other two algorithms perform worse than F–PFGE does.

V. DISCUSSIONS AND FUTURE WORK

In this paper, we have presented a unified solution, called parameter-free graph embedding, for the following two problems for general supervised subspace learning: 1) how to directly

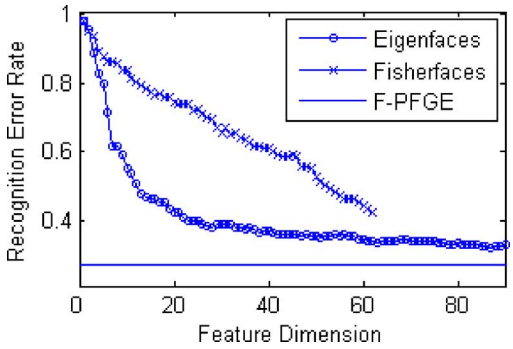


Fig. 6. Face recognition error rates versus feature dimension of Eigenfaces, Fisherfaces, and F–PFGE on the CMU PIE database.

Fig. 7. Cropped samples from the ORL database.

TABLE III
BEST RECOGNITION ACCURACIES (%) COMPARED AMONG EIGENFACES, FISHERFACES, AND F–PFGE ON THE ORL DATABASE

Algorithm	*Eigenfaces*	*Fisherfaces*	*F-PFGE*
Accuracy	96%	96.5%	97.5%
Training Time	≈0.30s	≈ 0.50s	≈ 161s

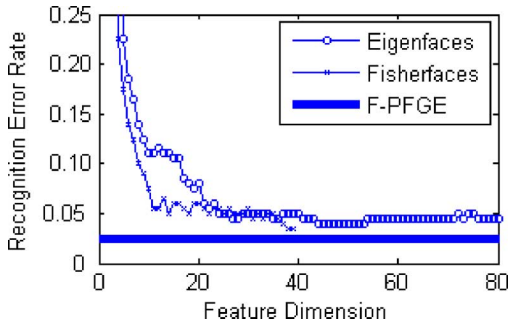
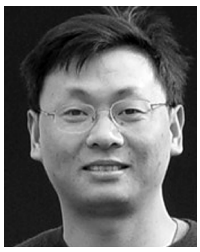


Fig. 8. Face recognition error rates versus feature dimension of Eigenfaces, Fisherfaces, and F–PFGE on the ORL database.

optimize the classification accuracy and 2) how to automatically determine the optimal subspace dimension and combine the selected features for final classification. More specifically, in our solution, the feature Gram matrix is learned by using the semidefinite programming method and ensuring that the neighboring points of each sample share the same class label; then, the optimal subspace dimension and feature fusing weights are automatically obtained from the singular value decomposition of the learned feature Gram matrix. Our proposed framework elicits some new research directions for further study, such as how to efficiently solve the semidefinite programming problem when the feature dimension is above 1000.

REFERENCES

[1] C. Alpert, A. Kahng, and S. Yao, "Spectral partitioning with multiple eigenvectors," *Discr. Appl. Math.*, vol. 90, no. 3, pp. 3–26, 1999.
[2] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 1997.
[3] B. Borchers, "CSDP, A C library for semidefinite programming," *Optim. Meth. Softw.*, vol. 11, no. 1, pp. 613–623, 1999.
[4] M. Brito, J. Quiroz, and V. Yukich, "Graph-theoretic procedures for dimension identification," *J. Multivar. Anal.*, vol. 81, no. 1, pp. 67–84, 2002.
[5] M. Brito and A. Quiroz, "Degree frequencies in the minimal spanning tree and dimension identification," *Commun. Statist.: Theory Methods*, vol. 33, no. 1, pp. 99–105, 2004.
[6] F. Chung, "Spectral graph theory," in *Reg. Conf. Series Math.*, 1997, vol. 92.
[7] J. Costa and A. Hero, "Geodesic entropic graphs for dimension and entropy estimation in manifold learning," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2210–2221, Aug. 2004.
[8] T. Cox and M. Cox, *Multidimensional Scaling*, 2nd ed. London, U.K.: Chapman & Hall, 2001.
[9] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. New York: Academic, 1991.
[10] X. He and P. Niyogi, Locality preserving projections 2002, Tech. Rep. 2002-09.
[11] X. Hu and L. Xu, "A comparative investigation on subspace dimension determination," *Neural Netw.*, vol. 17, pp. 1051–1059, 2004.
[12] I. Joliffe, *Principal Component Analysis*. New York: Springer-Verlag, 1986.
[13] B. Kegl, "Intrinsic dimension estimation using packing numbers," in *Proc. Neur. Inf. Process. Syst.*, Vancouver, BC, Canada, 2002, CD ROM.
[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[15] H. Li, T. Jiang, and K. Zhang, "Efficient and robust feature extraction by maximum margin criterion," *Adv. Neural Inf. Process. Syst.*, vol. 16, 2004.
[16] T. Lin, H. Zha, and S. Lee, "Riemannian manifold learning for nonlinear dimensionality reduction," in *Proc. 9th Eur. Conf. Computer Vision, Part I*, Graz, Austria, May 7–13, 2006, vol. 3951, pp. 44–55.
[17] H. Liu, C. Su, Y. Chiang, and Y. Hung, "Personalized face verification system using owner-specific cluster-dependent LDA-subspace," in *Proc. Int. Conf. Pattern Recognition*, 2004, pp. 344–347.
[18] J. Luettin and G. Maitre, "Evaluation protocol for the extended M2VTS database (XM2VTS)," *DMI Percept. Artif. Intell.*, 1998.
[19] A. Martinez and A. Kak, "PCA versus LDA," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 2, pp. 228–233, Feb. 2001.
[20] Olivetti and Oracle Res. Lab. The Olivetti Oracle Res. Lab. Fface Database of Faces ORL-Database [Online]. Available: http://www.cam-orl.co.uk/facedatabase.html.
[21] T. Sim, S. Baker, and M. Bsat, The CMU pose, illumination, and expression (PIE) database of human faces Robot. Instit., Carnegie-Mellon Univ., Pittsburgh, PA, 2001, Tech. Rep. CMU-RI-TR-01-02.
[22] M. Turk and A. Pentland, "Face recognition using eigenfaces," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Maui, HI, 1991, pp. 586–590.
[23] X. Wang and X. Tang, "A unified framework for subspace face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1222–1228, Sep. 2004.
[24] K. Weinberger, J. Blitzer, and L. Saul, "Distance metric learning for large margin nearest neighbor classification," in *Adv. Neural Inf. Process. Syst.*. Cambridge, MA: MIT Press, 2006, vol. 18.
[25] K. Weinberger, B. Packer, and L. Saul, "Nonlinear dimensionality reduction by semidefinite programming and kernel matrix factorization," in *Proc. 10th Int. Workshop Artificial Intelligence and Statistics*, 2005.
[26] K. Weinberger and L. Saul, "Unsupervised learning of image manifolds by semidefinite programming," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2004, pp. 988–995.
[27] K. Weinberger, F. Sha, and L. Saul, "Learning a kernel matrix for nonlinear dimensionality reduction," in *Proc. 21st Int. Conf. Machine Learning*, 2004, pp. 839–846.
[28] S. Yan, D. Xu, Q. Yang, X. Tang, and H. Zhang, "Discriminant analysis with tensor representation," in *Proc. Computer Vision and Pattern Recognition*, 2005, pp. 526–532.
[29] S. Yan, D. Xu, B. Zhang, and H. Zhang, "Graph embedding: A general framework for dimensionality reduction," in *Proc. Computer Vision and Pattern Recognition*, 2005, pp. 830–837.
[30] L. Yang, "Building k edge-disjoint spanning trees of minimum total length for isometric data embedding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1680–1683, Oct. 2005.
[31] H. Yu and J. Yang, *A Direct LDA Algorithm for High Dimensional Data with Application to Face Recognition*, 2nd ed. New York: Academic, 2001, vol. 34, pp. 2067–2070.



Shuicheng Yan (M'06) received the B.S. and Ph.D. degrees from the Applied Mathematics Department, School of Mathematical Sciences, Beijing University, Beijing, China, in 1999 and 2004, respectively.

His research interests include computer vision and machine learning.

Jianzhuang Liu (M'02–SM'02) received the B.E. degree from Nanjing University of Posts Telecommunications, Nanjing, China, in 1983, the M.E. degree from Beijing University of Posts Telecommunications, Beijing, China, in 1987, and the Ph.D. degree from the Chinese University of Hong Kong, Hong Kong, China, in 1997.

From 1987 to 1994, he was a Faculty Member in the Department of Electronic Engineering, Xidian University, Xi'an, China. From 1998 to 2000, he was a Research Fellow with the School of Mechanical and Production Engineering, Nanyang Technological University, Singapore. He was then a Postdoctoral Fellow with the Chinese University of Hong Kong for several years. Currently, he is an Assistant Professor in the Department of Information Engineering, the Chinese University of Hong Kong. His research interests include image processing, computer vision, pattern recognition, and graphics.

Xiaoou Tang (S'93–M'96–SM'02) received the B.S. degree from the University of Science and Technology of China, Hefei, in 1990, the M.S. degree from the University of Rochester, Rochester, NY, in 1991, and the Ph.D. degree fom the Massachusetts Institute of Technology, Cambridge, in 1996.

Currently, he is a Professor and the Director of Multimedia Lab in the Department of Information Engineering, the Chinese University of Hong Kong, Hong Kong, China. He is the Group Manager of the Visual Computing Group, Microsoft Research Asia, Beijing,China. His research interests include computer vision, pattern recognition, and video processing.

Dr. Tang is a Local Chair of the IEEE International Conference on Computer Vision (ICCV) 2005, an Area Chair of ICCV'07, a Program Chair of ICCV'09, and a General Chair of the ICCV International Workshop on Analysis and Modeling of Faces and Gestures 2005. He is a Guest Editor of the Special Issue on Underwater Image and Video Processing for the IEEE JOURNAL OF OCEANIC ENGINEERING and the Special Issue on Image- and Video-based Biometrics for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. He is an associate editor of IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (PAMI).

Thomas S. Huang (S'61–M'63–SM'76–F'79–LF'01) received the B.S. degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, R.O.C., and the M.S. and D.Sc. degrees in electrical engineering from the Massachusetts Institute of Technology (MIT), Cambridge.

He was on the Faculty of the Department of Electrical Engineering at MIT from 1963 to 1973 and on the Faculty of the School of Electrical Engineering and Director of its Laboratory for Information and Signal Processing at Purdue University, West Lafayette, IN, from 1973 to 1980. In 1980, he joined the University of Illinois, Urbana-Champaign, where he is William L. Everitt Distinguished Professor of Electrical and Computer Engineering, Research Professor at the Coordinated Science Laboratory, and Head of the Image Formation and Processing Group at the Beckman Institute for Advanced Science and Technology. He is Co-Chair of the Institute's major research theme Human Computer Intelligent Interaction. His research interests include the transmission and processing of multidimensional signals. He has published many books, and more than 500 papers on network theory, digital filtering, image processing, and computer vision.

Dr. Huang is a member of the National Academy of Engineering, a Foreign Member of the Chinese Academies of Engineering and Sciences, and a Fellow of the International Association of Pattern Recognition, IEEE, and the Optical Society of America. He received a Guggenheim Fellowship, an A.V. Humboldt Foundation Senior U.S. Scientist Award, and a Fellowship from the Japan Association for the Promotion of Science. He received the IEEE Signal Processing Society's Technical Achievement Award in 1987 and the Society Award in 1991. He was awarded the IEEE Third Millennium Medal and Honda Lifetime Achievement Award for "contributions to motion analysis" in 2000. In 2001, he received the IEEE Jack S. Kilby Medal and in 2002, he received the King-Sun Fu Prize from the International Association of Pattern Recognition, and the Pan Wen-Yuan Outstanding Research Award. In 2005, he received the Okawa Prize. In 2006, he was named by IS&T and SPIE as the Electronic Imaging Scientist of the year. He is a Founding Editor of the *International Journal of Computer Vision, Graphics, and Image Processing* and Editor of the Springer Series in Information Sciences, published by Springer-Verlag.