Exploiting Side Information in Locality Preserving Projection

Senjian An, Wanquan Liu and Svetha Venkatesh
Dept. of Computing, Curtin University of Technology
GPO Box U1987, Perth, WA 6845, Australia.
s.an, w.liu, s.venkatesh@curtin.edu.au

Abstract

Even if the class label information is unknown, side information represents some equivalence constraints between pairs of patterns, indicating whether pairs originate from the same class. Exploiting side information, we develop algorithms to preserve both the **intra-class and inter-class local structures**. *This new type of locality preserving projection (LPP), called LPP with side information (LPPSI), preserves the data's local structure in the sense that the close, similar training patterns will be kept close, whilst the close but dissimilar ones are separated. Our algorithms balance these conflicting requirements, and we further improve this technique using kernel methods. Experiments conducted on popular face databases demonstrate that the proposed algorithm significantly outperforms LPP. Further, we show that the performance of our algorithm with partial side information (that is, using only small amount of pair-wise similarity/dissimilarity information during training) is comparable with that when using full side information. We conclude that exploiting side information by preserving both similar and dissimilar local structures of the data significantly improves performance.*

1. Introduction

Finding the optimal discriminant subspace is one of the most important topics in computer vision and pattern recognition. It has been extensively studied and widely applied in face recognition, document indexing and text categorization, where the data is usually represented by vectors of high dimensionality. The high dimensionality may incur computational difficulty and classification deficiency. The classical linear discriminant analysis (LDA)[7] tries to maximize the class separability by maximizing the Fisher criterion

$$J(G) = trace\{(G^T S_w G)^{-1}(G^T S_b G)\} \qquad (1)$$

where G is the projection matrix, S_w is the within-class scatter matrix and S_b is the between-class scatter matrix. The solution of G is a set of leading eigenvectors of $S_w^{-1}S_b$ if S_w is nonsingular. In face recognition, the number of training images is often less than the data dimensionality and thus S_w is singular. To overcome this so-called *small sample* problem, several variants of LDA have been developed [1, 24, 22, 23, 6, 17, 28, 18, 16]. Since LDA and its variants use class means to represent the class, the local structure of data is ignored. A number of recent research efforts have shown that the face images possibly reside on a nonlinear sub-manifold [5, 11]. To overcome the drawbacks of LDA and its variants, locality preserving projection (LPP) seeks projections to preserve the local structure of the data and has been successfully applied in face recognition [11, 2] and document indexing [3]. LPP can be applied for both supervised and unsupervised learning. In the supervised case, one simple way to use the label information is to set the weights to be zero on the dissimilar training pairs [10, 15] and thus LPP just preserves the intra-class local structures of the training patterns. That is, close patterns with same label will be kept close after projection. However, the local structure of the *inter-class* training pairs was ignored. Recently, a method called locality discriminating indexing [12], utilizes label information by minimizing the ratio of the close intra-class and inter-class distances.

There is a large class of applications for which although sample labels are unknown, information exists as to whether individual samples belong to the same class or not. This information is called *side information* and represents some equivalence constraints between pair of patterns, indicating whether a pair of patterns originate from the same class (similar patterns) or from different classes (dissimilar patterns). This is a weaker condition than that required for full supervision, but more than that required for unsupervised training. For example, in applications of access control or attendance management, images from both the access group and outside the group can be included. Although we may not have information on who the people outside the group are, the similarity and dissimilarity information between the images from within and outside the group can be calculated. In designing such access control systems, we need to ensure

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

the inter-class pairs between the people in and outside the group are kept separate.

Side information has been exploited and successfully applied in metric learning [26, 25] and video object classification with support vector machine and Kernel Logistic Regression [27]. To incorporate side information in LPP, [4] suggest minimizing the modified LPP cost function by adding intra-class distances and subtracting inter-class distances according to side information.

Our major contribution in this paper is to systematically exploit side information and develop algorithms to balance the preservation of *intra-class and inter-class local structures*. We preserve the inter-class local structure in the sense that the close patterns with different labels are kept *separate*. This new type of locality preserving projection is called LPP with side information (LPPSI). In the supervised case, where the labels of training patterns are available, LPPSI exploits the label information in a more efficient way than LPP and potentially improves classification performance. In the unsupervised case (say clustering, image or document retrieval), where some side information is available, the proposed method can utilize this information to improve the clustering or retrieval performance.We successfully apply the proposed method to face recognition and experiments demonstrate that LPPSI significantly outperforms LPP and that LPPSI performs quite well when only a small amount (say, 2%) of side information is available. Further, a kernel version of LPPSI is developed to utilize the nonlinear structure of face images and we demonstrate its superior performance to LPPSI.

The novelty of the techniques relies in combining side information with locality preserving projection and demonstrating its application to face recognition. The significance of this approach is as follows: First, in exploring the local structures of the data, LPPSI exploits side information and considers both inter-class and intra-class structures which are more complete in describing the data's local properties. Second, the proposed algorithms are developed based on side information and thus are applicable to the case only pairwise similarity/dissimilarity is available from the training patterns. In the case where it is expensive to label the training patterns (say, a large amount of images), one can select a small amount of the training pairs and identify their pairwise similarity. Also, in a broad family of applications, side information can be obtained partially without supervision. We can get unsupervised equivalence constraints using temporal continuity in data such as video sequences. Sometimes side information is the natural form of supervision. For example, in image retrieval, there is no concept of category but there is the notion of similarities between the query and retrieved images.

The layout of the rest in this paper is as follows. In Section 2, we briefly review the formulation of LPP and its properties. Section 3 addresses LPP with side information. In Section 4, we address the nonlinear extension by kernels, i.e. kernel LPP with side information. Section 5 provides experimental results on some popular face databases to illustrate the performance of the proposed algorithms with comparison to LPP.

2. A Brief Review of Locality Preserving Projection

Let $x_i, i = 1, 2, \cdots, n$, denote the training patterns of m classes. We use $X = [x_1, x_2, \cdots, x_n]$ to denote the data matrix and use $l(x_i)$ to denote the label of x_i, say, $l(x_i) = k$ implies that x_i belongs to class k.

Locality Preserving Projection (LPP) aims to preserve the local structure of the data and can be obtained by solving the following minimization problem

$$\begin{array}{rcl} g_{opt} & = & \arg\min_g \sum_{i,j} [g^T(x_i - x_j)]^2 S_{ij} \\ & = & \arg\min_g g^T X L X^T g \end{array} \quad (2)$$

with the constraint

$$g^T X D X^T g = 1. \quad (3)$$

where $L = D - S$ is the *graph Laplacian*, D is a diagonal matrix with $D_{ii} = \sum_j S_{ij}$ which measures the locality density around x_i, and S is the similarity matrix. A typical way of defining S is as follows: $S_{ij} = exp(-\|x_i - x_j\|^2/\sigma^2)$ if $\|x_i - x_j\| > \epsilon$, and $S_{ij} = 0$ otherwise. In the supervised case, where the labels of the training patterns are known, one simple way to use the label information is to let S_{ij} equal 0 if x_i, x_j belong to different classes.

The optimal solution g_{opt} is the minimum eigenvector of the generalized eigenvalue problem

$$X L X^T g = \gamma X D X^T g. \quad (4)$$

For multi-dimensional projection, the LPP uses the d largest eigenvectors, say $g_1, g_2, \cdots, g_d$, as the columns of the projection matrix $G = [g_1, g_2, \cdots, g_d]$.

The heavy penalty on the close training pairs will force them to keep close in the reduced subspace.

3. Locality Preserving Projection with Side Information

3.1. Motivation

In this section, we consider the cross-validation errors of nearest neighbor classifiers. It will be shown that the cross-validation errors are dominated by the intra- and inter-class differences with relatively smaller distances and this motivates us to develop algorithms to find a projective map

which preserves the neighborhood of close intra-class patterns, while preserving the separability of close inter-class patterns.

Cross-validation is a typical way to estimate the generalization performance of learning algorithms [19]. In l-fold cross-validation, one divides the data into l subsets of (approximately) equal size and trains the classifier l times, each time leaving out one of the subsets from training, but using the omitted subset to compute the classification errors. If l equals the sample size, this is called leave-one-out cross-validation (LOO-CV).

Nearest neighbor is the simplest, but also most popular classifiers in pattern recognition. One often applies nearest neighbor classifiers to identify the test patterns after dimension reduction using LDA or LPP. Now we consider the LOO errors of the nearest neighbor classifier with training patterns $x_i, i = 1, 2, \cdots, n$, in the original or projected subspaces. Suppose x_k is left-out for testing. The nearest neighbor classifier compares the distances of x_k to all the other training patterns and identifies x_k to belong to the same class as its nearest neighbor. Let

$$\begin{array}{rcl} d_I(x_k) & = & \min_j\{\|x_k - x_j\|, j \neq k, l(x_k) = l(x_j)\} \\ d_E(x_k) & = & \min_j\{\|x_k - x_j\|, j \neq k, l(x_k) \neq l(x_j)\}. \end{array} \tag{5}$$

x_k is correctly identified in the LOO procedure, if and only if, $d_I(x_k) < d_E(x_k)$. This is true for all $x_k, k = 1, 2, \cdots, n$. Hence the generalization performance of the nearest neighbor classifier estimated by cross-validation is dominated by the close intra-class and inter-class training patterns. More precisely, we have the following observation

1. *Among the intra-class differences in $\{x_i - x_j, l(x_i) = l(x_j)\}$, the ones with smaller distances are more dominant in determining the generalization performance of the nearest neighbor classifier;*

2. *Among the inter-class differences in $\{x_i - x_j, l(x_i) \neq l(x_j)\}$, the ones with smaller distances are more dominant in determining the generalization performance of the nearest neighbor classifier.*

Hence, a good projective map needs to ensure that: 1) close intra-class pairs remain close after projection; and 2) close but dissimilar pairs, are kept separate after projection. For the first requirement 1), we need to minimize the weighted sum of the intra-class distances with heavy weights on the close intra-class pairs in the original space. On the other hand, for 2), we need to maximize the weighted sum of the inter-class distances with heavy weights on the close inter-class pairs in the original space. The heavy weights on the close inter-class pairs will force them to be kept separate, while the heavy weights on the close intra-class pairs will force them to remain close.

However, these two tasks may be conflicting. Next, we will develop an algorithm to balance these two tasks.

3.2. The Objective Function

Let Ω_s and Ω_d denote the sets of available similar and dissimilar training pairs respectively and let S_{ij} denote the similarity of a training pair (x_i, x_j) in Ω_s (or Ω_d). In supervised learning, we have the labels of all training patterns and thus the full side information is available. In this case

$$\begin{array}{rcl} \Omega_s & = & \{(x_i, x_j), l(x_i) = l(x_j)\} \\ \Omega_d & = & \{(x_i, x_j), l(x_i) \neq l(x_j)\} \end{array} \tag{6}$$

In the case only partial side information is available, Ω_s and Ω_d represent some subsets of similar and dissimilar training pairs respectively. For notational convenience, we assume that (x_j, x_i) belongs to Ω_s (or Ω_d) as well if $(x_i, x_j) \in \Omega_s$ (or Ω_d respectively).

The similarity S_{ij} can be defined as

$$S_{ij} = e^{\frac{-\|x_i - x_j\|^2}{\sigma^2}} \tag{7}$$

or

$$S_{ij} = \frac{|x_i^T x_j|}{\|x_i\|\|x_j\|}. \tag{8}$$

The latter computes the cosine of the angle between vectors x_i and x_j. It is often called *cosine similarity* and is widely used to measure similarities of images and documents.

Different objective functions yield different algorithms with different properties. While LPP aims to ensure close patterns remain close by minimizing the intra-class distances, we aim to preserve both intra- and inter-class local structures by minimizing the following objective function

$$\begin{array}{rcl} J_1(g) & = & \lambda \sum\limits_{(x_i, x_j) \in \Omega_s} [g^T(x_i - x_j)]^2 W_{ij}^{(s)} + (1 - \lambda) g^T g \\ & = & \lambda g^T C_s g + (1 - \lambda) g^T g \end{array} \tag{9}$$

with the constraint

$$g^T C_d g = 1. \tag{10}$$

where

$$\begin{array}{rcl} C_s & = & \sum\limits_{(x_i, x_j) \in \Omega_s} (x_i - x_j)(x_i - x_j)^T W_{ij}^{(s)} \\ C_d & = & \sum\limits_{(x_i, x_j) \in \Omega_d} (x_k - x_l)(x_k - x_l)^T W_{kl}^{(d)}. \end{array} \tag{11}$$

and the weights $W_{ij}^{(s)}$ and $W_{kl}^{(d)}$ are determined by the sim-

ilarities and can be defined as

$$\begin{aligned} W_{ij}^{(s)} &= \begin{cases} S_{ij} & \text{if } S_{ij} > \epsilon_s \text{ and } (x_i, x_j) \in \Omega_s \\ 0; & \text{Otherwise.} \end{cases} \\ W_{kl}^{(d)} &= \begin{cases} S_{kl} & \text{if } S_{kl} > \epsilon_d \text{ and } (x_k, x_l) \in \Omega_d \\ 0; & \text{Otherwise.} \end{cases} \end{aligned} \tag{12}$$

The thresholds ϵ_s and ϵ_d control the similar and dissimilar neighborhoods, and may be different since the inter-class distances are usually larger than the intra-class distances.

It is easy to check that the optimal solution g_{opt} is equal to the optimal solution $\hat{g}$, up to a scale factor, of the following maximization problem

$$\max_g J_2(g) = g^T C_d g \tag{13}$$

with the constraint

$$\lambda g^T C_s g + (1-\lambda) g^T g = 1. \tag{14}$$

From (13), one can see that the optimal solution maximizes the weighted sum of inter-class distances when $\lambda = 0$. And from (9), one can see that the optimal solution minimizes the weighted sum of intra-class distances when $\lambda = 1$. Hence, the objective function can be interpreted as a balance for two possibly conflicting tasks: minimizing intra-class distances and maximizing inter-class distances and the balance is controlled by the parameter $\lambda \in [0, 1]$.

3.3. The Algorithms

The solution g_{opt} of (9) is the eigenvector associated with the smallest eigenvalue of the following generalized eigenvalue problem

$$[\lambda C_s + (1-\lambda) I] g = \gamma C_d g. \tag{15}$$

Suppose the projection matrix is of rank d. We need to find the d eigenvectors associated with the d smallest eigenvalues.

If C_d is not of full rank, one can use (13) and solve the following generalized eigenvalue problem

$$C_d g = \gamma [\lambda C_s + (1-\lambda) I] g \tag{16}$$

to find the d largest eigenvectors.

Note that the columns of the projection matrices obtained by solving (15) and (16) are identical up to a scale factor which can be computed using the constraints. In our experiments, we normalize the eigenvectors to be of unit norm.

In summary, the proposed algorithm includes the following four steps:

1. *Compute the similarity matrix S, and then set the weight matrices $W^{(s)}$ and $W^{(d)}$ for intra-class and inter-class training patterns respectively;*
2. *Compute the similar and dissimilar weighted covariance matrices C_s and C_d;*
3. *Solve the generalized eigenvalue problem (16) (or (15)) to find the d largest (or smallest respectively) eigenvectors;*
4. *Project the training patterns and the test patterns into the selected eigenvector space and use nearest neighbor classifier to identify the test patterns.*

3.4. Comparison to LPP

Let us define the *similar and dissimilar graph Laplacians* as

$$\begin{aligned} L_s &= D^{(s)} - W^{(s)} \\ L_d &= D^{(d)} - W^{(d)} \end{aligned} \tag{17}$$

where $D^{(s)}$ is a diagonal matrix with diagonals $D_{ii}^{(s)} = \sum_j W_{ij}^{(s)}$, and $D^{(d)}$ is a diagonal matrix with diagonals $D_{ii}^{(d)} = \sum_j W_{ij}^{(d)}$. Then we have $C_s = X L_s X^T$ and $C_d = X L_d X^T$, and (9) can be described as

$$\min_g \lambda g^T X L_s X^T g + (1-\lambda) g^T g \tag{18}$$

with the constraint

$$g^T X L_d X^T g = 1. \tag{19}$$

If we use the same similarity measure and use the same thresholds $\epsilon_s = \epsilon$, L_s is the same as the *graph Laplacian* L in the formulation (2) of LPP for supervised learning. For any training pattern x_i, LPP minimizes the distances of x_i to any similar patterns in its neighborhood which is determined by the threshold ϵ. This ensures the closeness of x_i to its similar neighbors but also raises risk in making x_i close to its dissimilar neighbors as well. This risk is avoided in LPPSI. By introducing *dissimilar graph Laplacian* L_d and the controlling parameter λ, LPPSI achieves a good balance between minimizing the distances to similar neighbors and maximizing the distances to dissimilar neighbors. In practice, one can use cross-validation [19] to find the optimal λ by minimizing the cross-validation errors.

4. Kernel LPP with Side Information

In this section, we present a kernel version of LPPSI, named as KLPPSI. Consider a nonlinear map $x_i \rightarrow \Phi_i = \Phi(x_i)$ induced by a kernel where $k(x_i, x_j) = \Phi(x_i)^T \Phi(x_j)$. Let $\Phi = [\Phi_1, \Phi_2, \cdots, \Phi_n]$ and let K denote the matrix with $K_{i,j} = k(x_i, x_j)$. The optimal LPPSI in the kernel induced feature space can be obtained by solving

$$\begin{aligned} \min_g J(g) &= \lambda \textstyle\sum_{i,j} [g^T(\Phi_i - \Phi_j)]^2 W_{ij}^{(s)} + (1-\lambda) g^T g \\ &= \lambda g^T \Phi L_s \Phi^T g + (1-\lambda) g^T g \end{aligned} \tag{20}$$

with the constraint

$$g^T \Phi L_d \Phi^T g = 1 \tag{21}$$

where L_s, L_d are the *similar and dissimilar graph Laplacians* and are defined in (17).

First, we show that $g_{opt} \in span\{\Phi\}$. It is trivial if Φ is of full column rank. Now assume that Φ is not of full column rank and let $\Phi^\perp$ denote a basis of its null space. Then g_{opt} can be represented as $g_{opt} = g_1 + g_2$ where $g_1 \in span(\Phi)$ and $g_2 \in span(\Phi^\perp)$. Note that $g_2^T \Phi = 0, g_2^T g_1 = 0$, we have $J(g) = J(g_2) + (1-\lambda) g_2^T g_2 \geq J(g_2)$ and g_2 also satisfies the constraint (21). Hence if g_{opt} is the optimal solution, then g_2 must be the zero vector and therefore $g_{opt} \in span\{\Phi\}$.

So there is h_{opt} such that $g_{opt} = \Phi h_{opt}$ and h_{opt} can be obtained by solving

$$\min_h J(h) \quad = \quad \lambda h^T K L_s K h + (1-\lambda) h^T K h \tag{22}$$

with the constraint

$$h^T K L_d K h = 1. \tag{23}$$

The solution is the minimum eigenvector of the following generalized eigenvalue problem

$$[\lambda L_s K + (1-\lambda) I] h = \gamma L_d K h. \tag{24}$$

Similarly, the objective function (13) can be kernelized as

$$\min_h J(h) \quad = \quad h^T K L_d K h \tag{25}$$

with the constraint

$$\lambda h^T K L_s K h + (1-\lambda) h^T K h = 1. \tag{26}$$

And its solution is the largest eigenvector of the following generalized eigenvalue decomposition problem

$$L_d K h = \gamma [\lambda L_s K + (1-\lambda) I] h. \tag{27}$$

Let $\{h_i\}_{i=1}^d$ be the eigenvectors of (27)(or (24)) associated with the d largest (or smallest respectively) eigenvalues and denote $H = [h_1, h_2, \cdots, h_d]$. Then the projection matrix will be ΦH and the projection of a pattern x will be

$$\begin{aligned} p(x) &= H^T \Phi^T \Phi(x) \\ &= H^T [k(x, x_1), k(x, x2), \cdots, k(x, x_n)]^T. \end{aligned} \tag{28}$$

Hence, in either training or testing stages, we don't need to access the nonlinear features $\Phi(x)$. With the kernel function $k(\cdot,\cdot)$, one can compute the kernel matrix K and solve the eigenvalue problems (27) or (24) to obtain H. Then the training and test patterns can be projected into the selected feature space using (28) directly without computing Φ. We summarize the KLPPSI algorithm as follows:

1. *Compute the weight matrices $W^{(s)}$ and $W^{(d)}$ for intra-class and inter-class training patterns respectively;*

2. *Compute the similar and dissimilar Laplacian matrices L_s and L_d;*

3. *Compute the kernel matrix K with $K_{ij} = k(x_i, x_j)$;*

4. *Solve the generalized eigenvalue problem (27) (or (24)) to find the d largest (or smallest respectively) eigenvectors and normalize them to be unit norm;*

5. *Compute the projections of the training patterns and the test patterns using (28)and use nearest neighbor classifier to identify the test patterns.*

Typical kernel functions $k(\cdot,\cdot)$ include linear kernel $k(x_i, x_j) = x_i^T x_j$, polynomial kernel $(x_i^T x_j + 1)^d$ and Gaussian kernel $k(x_i, x_j) = \exp\left(-\|x_i - x_j\|^2/\sigma^2\right)$. The performance of KLPPSI is governed by the kernel parameters and the controlling parameter λ. One can use cross-validation [19] to find the optimal hyper-parameters by minimizing the cross-validation errors.

5. Experimental Results

Experiments were conducted on two databases: CMU PIE [20, 21] and the original and Extended Yale Face Database B (Yale B) [9, 14] to test the performance of the proposed algorithms with comparisons to LPP. The CMU PIE face database contains 68 individuals with 41368 face images. The face images were captured by 13 synchronized cameras and 21 flashes, under varying pose, illumination and expression. The extended Yale Face Database B [14] contains 16128 images of 28 human subjects under 9 poses and 64 illumination conditions. The data format of this database is the same as the original Yale Face Database B [9] which contains 5760 images of 10 people under the same 9 poses and 64 illumination conditions.

We provide three experiment results. The first one compares the performance of LPPSI to the reported performance of LPP in the study by [2]. The second one aims to test the performance in robust face recognition across pose and lighting variations. And the third experiment tests the performance of LPPSI when only a small amount of side information is available.

We used cosine similarity through all the experiments since it is popular in measuring image similarities. For KLPPSI, we used the Gaussian kernel $k(x_i, x_j) = \exp\left(-\|x_i - x_j\|^2/(2\sigma^2)\right)$. The parameters for LPPSI and KLPPSI are provided in Tables 1-2. For convenience of comparison to LPP where all the similar pairs were used in supervised learning, we set the threshold ϵ_s to be zero in all the experiments.

Table 1. Parameters for LPPSI on CMU PIE and Yale B face databases.

Parameters	ϵ_d	ϵ_s	λ
CMU PIE	0.85	0	0.7
Yale B-Experiment 1	0.7	0	0.7
Yale B-Experiments 2&3	0.7	0	0.99

Table 2. Parameters for KLPPSI on CMU PIE and Yale B face databases.

Parameters	ϵ_d	ϵ_s	λ	σ
CMU PIE	0.85	0	0.7	0.7
Yale B-Experiment 1	0.7	0	0.7	1.2
Yale B-Experiments 2&3	0.7	0	0.99	0.5

5.1. Experiment 1

For convenience of comparison, this experiment adopts the same procedure as that in the study by [2]. From CMU PIE, we choose the five near frontal poses (C05,C07,C09,C27,C29) and use all the 11544 images under different illuminations, lighting and expressions, where each individual has 170 images except for a few bad images. From the Yale B Database B, we choose all the 2414 frontal images (except for a few bad images) for 38 people. All test image data used in the experiments are manually aligned, cropped, and then re-sized to 32x32 images.

A random subset with $l(= 5, 10, 20, 30)$ images per individual was taken with labels to form the training set, and the rest of the database was considered to be the testing set. For each l, we average the results over 50 random splits and we used the same splits and the same Matlab data files [1] which were used in [2].

Table 3. Performance (error rate) comparison on CMU PIE face database.

Method	5 Train	10 Train	20 Train	30 Train
LPP	30.8%(67)	21.1%(134)	14.1%(146)	7.13%(131)
LPPSI	23.52%(60)	11.39%(40)	5.77%(50)	4.13%(50)
KLPPSI	27.88%(20)	12.32%(30)	5.48%(30)	3.62%(60)

Table 4. Performance (error rate) comparison on the Yale B Database.

Method	5 Train	10 Train	20 Train	30 Train
LPP	24%(37)	11.4%(76)	7.1%(193)	7.5%(251)
LPPSI	20.44%(180)	9.40%(180)	3.86%(150)	1.92%(40)
KLPPSI	24.74%(50)	9.93%(60)	3.15%(50)	1.39%(40)

[1] which were downloaded from http://ews.uiuc.edu/ dengcai2/Data/data.html

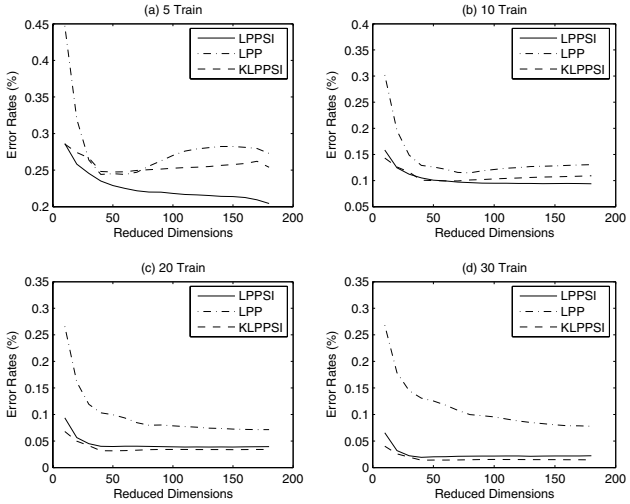

Figure 1. Performance vs Reduced Dimensions on Yale B Database.

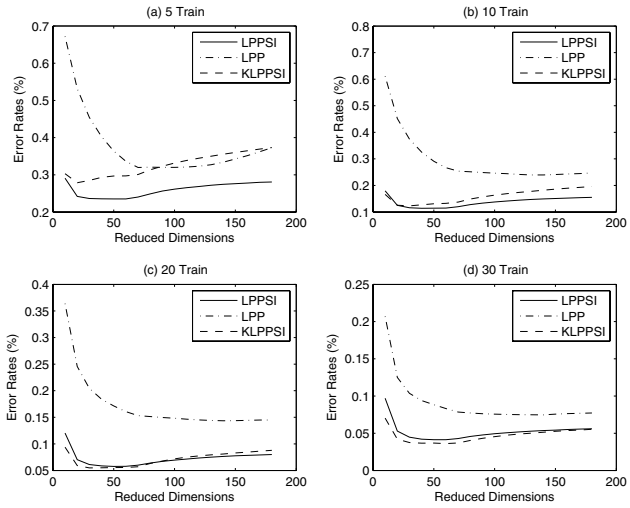

Figure 2. Performance vs Reduced Dimensions on PIE Database.

The performance is shown in Tables 3-4 and Figures 1-2. The performance for LPP in Tables 3-4 are taken from [2] for CMU PIE database and from http://ews.uiuc.edu/ dengcai2/Data/data.html. for the Yale B Database. The numbers in the brackets are the best dimensions. The performance of LPPSI is consistently and significantly better than LPP. The kernel method further improves the performance when training sample size is large (20 Train and 30 Train). However, when training sample sizes are small (5 Train and 10 Train), KLPPSI is more likely to suffer overfitting and thus performs worse than LPPSI.

5.2. Experiment 2

In the second experiment, to demonstrate robust face recognition across pose and lighting variations, we use all

the images of the full Yale B database where each people have 576 images with 9 poses and 64 lighting conditions. We found that most of the excluded bad images in the studies [9, 14, 2] are identifiable after histogram equalization [13] and include all the images in our experiment. We manually find the positions of eyes and mouths for each person under each pose, and then align and crop all the images according to these positions. Then all the images are re-sized to 32x32 images and preprocessed by histogram equalization.

Our procedure is as follows: First, we choose 10 people in the original Yale B database to understand the critical configurations for varying pose and lighting conditions. We do this by applying affinity propagation clustering[8] and find a universal configuration of 35 cluster centers of the total 576 images for each person. These 35 cluster centers represent 35 critical viewing conditions among 9 poses and 64 lighting conditions. We consider these 35 critical viewing conditions are universal for each person in the full Yale B database. Next, for each subject, we take the 35 images associated with these 35 critical viewing conditions as training images to train the classifiers. Finally, we compare the distances of the test images to all the training images and identify them using the nearest neighbor method.

The performance is shown in Table 5 and Figure 3, which demonstrate the clear advantage of LPPSI and KLPPSI over LPP. Note that LPPSI achieves an error rate of 3.36% with dimension 40. It shows that robust face recognition under varying lighting and poses can be achieved in quite a low dimensional subspace. In the training procedure, we need images under the critical viewing conditions. In case these images are not available, one may apply face synthesis techniques to generate these images.

Table 5. Performance (error rate) comparison on the Yale B Database with 9 poses and 64 lighting conditions.

Method	LPP	LPPSI	KLPPSI
Error rates	19.63%(60)	3.36%(40)	1.43%(40)

5.3. Experiment 3

This experiment adopts the same procedure as experiment 2 but using only a small amount of side information. We used the similarity/dissimilarity information of randomly selected 1% of dissimilar training pairs and half of the similar pairs. The total used pairs of side information (6866+11305=18171) is 2.06% of the total 883785 pairs of the $(35 \times 38 = 1310)$ training images.

The performance is shown in Figure 4. With around 2% of the total side information, LPPSI achieves an error rate of 3.52% (averaged on 10 runs) which is very close to the error rate of 3.36% achieved with full side information. For higher dimensions than 40, LPPSI performs even better using partial rather than full side information.

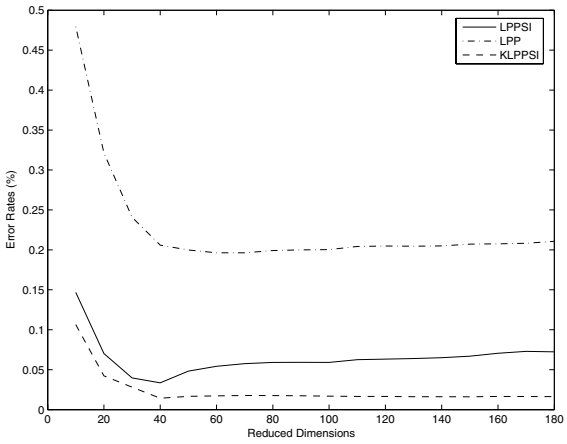

Figure 3. Performance vs Reduced Dimensions on the Yale B Database.

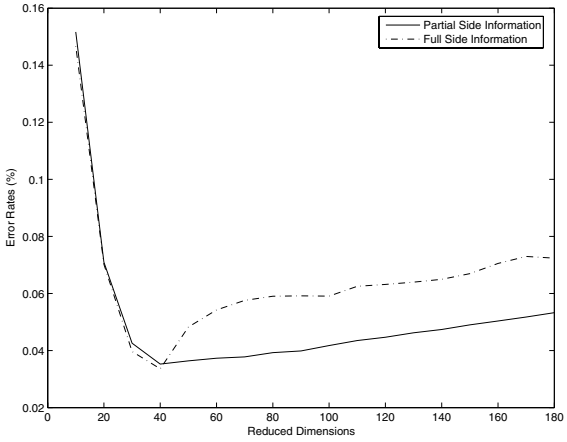

Figure 4. Performance of LPPSI on Partial and Full Side Information vs Reduced Dimensions.

6. Conclusion

By exploiting side information, we have presented a projection method to preserve both the intra-class and inter-class local structures of the data. The experiments in face recognition demonstrate that the proposed method significantly outperforms the previously developed locality preserving projection which ignored the inter-class local structure, and that robust face recognition across pose and lighting can be achieved in quite a low dimensional subspace. Although we focus on supervised learning and face recognition in our experiments, the proposed method has potential to improve performance over locality preserving projection in unsupervised learning with some side information, or in other pattern recognition problems such as digit recognition and document indexing.

References

[1] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 19(7):711–720, 1997. 1

[2] D. Cai, X. He, H. J., and Z. H.-J. Orthogonal laplacianfaces for face recognition. *IEEE Trans. Image Processing*, 15(11):3608–3614, 2006. 1, 5, 6, 7

[3] D. Cai, X. He, W. V. Zhang, and J. Han. Regularized locality preserving indexing via spectral regression. In *Proc. 2007 ACM Int. Conf. on Information and Knowledge Management (CIKM'07)*, 2007. 1

[4] H. Cevikalp, J. Verbeek, F. Jurie, and A. Klaser. Semi-supervised dimensionality reduction using pairwise equivalence constraints. In *International Conference on Computer Vision Theory and Applications*, 2008. 2

[5] Y. Chang, C. Hu, and M. Turk. Manifold of facial expression. In *Proc. of IEEE Int'l Workshop on Analysis and Modeling of Faces and Gestures*, 2003. 1

[6] L.-F. Chen, H.-Y. M. Liao, M.-T. Ko, J.-C. Lin, and G.-J. Yu. A new LDA-based face recognition system which can solve the small sample size problem. *Pattern Recognition*, 33(10):1713–1726, 2000. 1

[7] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley & Sons, 2000. 1

[8] B. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007. 7

[9] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition undervariable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001. 5, 7

[10] X. He and P. Niyogi. Locality preserving projections. In *Proc. Conf. Advances in Neural Information Processing Systems (NIPS'03)*, 2003. 1

[11] X. He, S. Yan, H. Y., N. P., and Z. H.-J. Face recognition using laplacianfaces. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 27(3):328–340, 2005. 1

[12] J. Hu, W. Deng, J. Guo, and W. Xu. Locality discriminating indexing for document classification. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007. 1

[13] B. Jahne. *Digital image processing*. Springer,Berlin, 2005. 7

[14] K. Lee, J. Ho, and D. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 27(5):684–698, 2005. 5, 7

[15] J.-B. Li, J.-S. Pan, and S.-C. Chu. Kernel class-wise locality preserving projection. *Inf. Sci.*, 178(7):1825–1835, 2008. 1

[16] D. Lin and X. Tang. Recognize high resolution faces: From macrocosm to microcosm. In *Proc. of CVPR'06*, 2006. 1

[17] C. Liu and H. Wechsler. Enhanced fisher linear discriminant moldels for face recognition. In *Proc. of ICPR'98*, 1998. 1

[18] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. Face recognition using LDA-based algorithms. *IEEE Trans. on Neural Networks*, 14(1):195–200, 2003. 1

[19] M. Plutowski. *Survey: Cross-validation in Theory and in Practice*. Research Report. Dept. of Computational Science Reserach, David Sarnoff Reserach Center, Princeton, New Jersey., 1996. 3, 4, 5

[20] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination, and expression (PIE) database. In *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, page 215, Washington, DC, USA, May 2002. IEEE Computer Society. 5

[21] T. Sim, S. Baker, and M. Bsat. The CMU pose, illumination, and expression (PIE) database. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 25(12):1615–1618, 2003. 5

[22] X. Wang and X. Tang. Dual-space linear discriminant analysis for face recognition. In *Proc. of CVPR'04*, 2004. 1

[23] X. Wang and X. Tang. Random sampling LDA for face recognition. In *Proc. of CVPR'04*, 2004. 1

[24] X. Wang and X. Tang. A unified framework for subspace face recognition. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 26(9):1222–1227, 2004. 1

[25] K. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classifiers. In *Proc. Conf. Advances in Neural Information Processing Systems (NIPS'05)*, 2006. 2

[26] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning with application to clustering with side information. In *Proc. Conf. Advances in Neural Information Processing Systems (NIPS'02)*, 2003. 2

[27] R. Yan, J. Zhang, J. Yang, and A. G. Hauptmann. A discriminative learning framework with pairwise constraints for video object classification. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 28(4):578–593, 2006. 2

[28] W. Zhao, R. Chellappa, and A. Krishnaswamy. Discriminant analysis of principal components for face recognition. In *Proc. of FGR'98*, 1998. 1