

# Marginal patch alignment for dimensionality reduction

Jie Xu<sup>1,2</sup> · Shengli Xie<sup>1</sup> · Wenkang Zhu<sup>3</sup>

Published online: 28 November 2015  
© Springer-Verlag Berlin Heidelberg 2015

**Abstract** Patch alignment (PA) framework provides us a useful way to obtain the explicit mapping for dimensionality reduction. Under the PA framework, we propose the marginal patch alignment (MPA) for dimensionality reduction. MPA performs the optimization from the part to the whole. In the phase of the patch optimization, the marginal between-class and within-class local neighborhoods of each training sample are selected to build the local marginal patches. By performing the patch optimization, on the one hand, the contributions of each sample for optimal subspace selection are distinguished. On the other hand, the marginal structure information is exploited to extract discriminative features such that the marginal distance between the two different categories is enlarged in the low transformed subspace. In the phase of the whole alignment, a trick is performed to unify all of the local patches into a globally linear system and make MPA obtain the whole optimization. The experimental results on the Yale face database, the UCI Wine dataset, the Yale-B face database, and the AR face database, show the effectiveness and efficiency of MPA.

**Keywords** Patch alignment framework · Dimensionality reduction · Margin · Classification

---

Communicated by V. Loia.

---

✉ Jie Xu  
njxujie@aliyun.com

<sup>1</sup> Faculty of Automation, Guangdong University of Technology, Guangzhou 510006, People's Republic of China

<sup>2</sup> College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China

<sup>3</sup> School of mathematics and statistics, Shaoguan University, Shaoguan 512000, Guangdong, China

## 1 Introduction

Dimensionality reduction (DR) is an effective method of avoiding the “curse of dimensionality”. It has achieved remarkable success in the fields of computer vision and pattern recognition. Linear discriminant analysis (LDA; [Belhumeur et al. 1997](#)) is one of the most famous DR techniques. It is also called the parametric discriminant analysis (PDA) in ([Fukunaga 1990](#)) since it uses the parametric form of the scatter matrix. LDA aims to preserve information between the data of different classes. To achieve this goal, LDA maximizes the between-class scatter meanwhile minimizing the within-class scatter. However, the excellent performance of LDA is based on the assumption that the samples in each class satisfy the Gaussian distribution. In real world, this assumption is hard to realize. Thus, they suffer performance degradation in cases of non-Gaussian distribution. Besides this, the available features of LDA are limited because the rank of the between-class matrix is at most  $c - 1$ , where  $c$  is the number of classes. It is often insufficient to separate the classes well with only  $c - 1$  features, especially in high-dimensional spaces. In addition, LDA only uses the centers of classes to compute the between-class scatter matrix, and thus fails to capture the marginal structure of classes effectively, which has been proven to be essential in classification.

Even with the above limitations, LDA still performs well and is one of representative linear dimensionality reduction methods. However, LDA, as a global method, exploits the linear global Euclidean structure and fails to use the nonlinear local information. [Liu et al. \(2004\)](#) and [Müller et al. \(2001\)](#) applied the kernel trick to handle the nonlinearities of data structure. These kernel methods map data points from the original feature space into another higher dimensional one such that the data structure becomes linear.

The well-known nonlinear dimensionality reduction (NLDR) methods, e.g., locally linear embedding (LLE; Roweis and Saul 2000), Isomap (Tenenbaum et al. 2000), and Laplacian eigenmap (Belkin and Niyogi 2003), are proposed based on the hypothesis that data in a complex high-dimensional data space may reside more specifically on or nearly on a low-dimensional manifold embedded within the high-dimensional space (Yang et al. 2011). All these NLDR methods are implemented restrictedly on training data and cannot provide us with explicit maps on new test data. Bengio et al. (2003) presented a kernel solution to solve this problem, in which LE, Isomap, and Laplacian Eigenmaps are integrated into the same eigen-decomposition framework, and then they were considered as learning eigenfunctions of a kernel. Their methods achieve the satisfactory results. After that, He et al. proposed locality preserving projections (LPP; He et al. 2005; He and Niyogi 2003) and neighborhood preserving embedding (NEP; He et al. 2005) to obtain the linear projections and achieved good classification performance.

Recently, Zhang et al. (2009) proposed the locality-based patch alignment (PA) framework. Many dimensionality reduction methods can be unified into the PA framework, even that they are based on different embedding criteria, such as PCA, LDA, LLE, NPE, ISOMAP, LE, LPP, and local tangent space alignment (LTSA; Zhang and Zha 2004). The PA framework helps us to better understand the common properties and intrinsic difference in different algorithms. The PA framework consists of two stages: part optimization and whole alignment. In the phase of part optimization, the local patches are built by each sample and the related ones and aim to capture the local geometry (locality) of data. In the second phase, all part optimizations are integrated to form the final global coordinate for all independent patches based on the alignment trick (Zhang et al. 2009). Different algorithms were shown to construct whole alignment matrices in an almost identical way, but vary with patch optimization. Under the PA framework, Zhang et al. (2009) proposed discriminative locality alignment (DLA) to overcome the mentioned limitations of LDA. In DLA, a local nearest patch is built by one sample associated with its related nearest samples. In the phase of patch optimization, the within-class compactness is represented as the sum of distances between each point and its  $K1$ -nearest neighbors of the same class; while the separability of different classes is characterized as the sum of distances between each sample and its  $K2$ -nearest neighbors of the different classes. By performing the patch optimization, the samples with the same class-label should cluster better than before, but it does not guarantee that the marginal distance can be enlarged larger than before.

The marginal structure information has been proved to be essential in classification and has attracted more and more attention. Many margin maximization-based algorithms have been developed, e.g., (MMC; Li et al. 2006), (MFA; Yan

et al. 2007), and (MMDA; Yang et al. 2009). These methods all exploit the marginal samples, which contain much discriminative information than other samples, to extract the discriminative features such that the extracted features are more suitable for classification. This also motivates us to exploit the local marginal discriminative information to develop a novel nonparametric DR technique, called marginal patch alignment (MPA). In MPA, we select the between-class and within-class local marginal samples of each training sample to build the marginal patches. It is obvious that the marginal patches can contain most discriminative information than other patches. Based on these local marginal patches, the within-class compactness is represented as the sum of distances between each sample and its  $K1$  within-class marginal samples associated with the nearest between-class samples; while the separability of different classes is characterized as the sum of distances between the two local marginal means. By performing marginal patch optimization, the marginal structure can be found and the discriminative information is exploited. As a result, the marginal distances between the two different categories can be enlarged in a direct way.

Unlike DLA, MPA focuses on marginal samples rather than the nearest samples. MPA selects the marginal samples to build patches, and then performs the patch optimization to push them away, such that the marginal distances become larger than before. The way taken by MPA is more direct than that of DLA in enlarging the marginal distance. More concretely, MPA first finds the  $K2$  nearest between-class neighbors in each other's class, and then turns back to find the  $K1$  within-class marginal samples. By such a way, the selected within-class samples near the margin of different classes are maybe not the nearest samples of the given sample. While this does not affect us to make the samples with the same class-labels clustered better than before.

The remainder of this paper is organized as follows: Section 2 outlines the PA framework. Section 3 describes the proposed MPA algorithm. Section 4 presents the advantages of MPA. Section 5 verifies the effectiveness of MPA through experiments on the Wine dataset from UCI, the Yale face database, the Yale-B face database, and the AR face database. Finally, our conclusions are drawn in Sect. 6.

## 2 Patch alignment (PA) framework

In the PA framework (Zhang et al. 2009), if considering a dataset with  $N$  measurements, we can build  $N$  patches for each training sample. Each patch consists of a training sample and its associated ones, which depend on both characteristics of the dataset and the objective of an algorithm. With these built patches, optimization can be imposed on them based on an objective function, and then the alignment trick (Zhang

et al. 2009) can be utilized to unify all the patches into a global coordinate. The specific procedure of the PA framework is listed as follows:

### 2.1 Part optimization

Considering an arbitrary training sample  $\vec{x}_i$  and its  $K$  associated samples (e.g., nearest neighbors)  $\vec{x}_{i1}, \dots, \vec{x}_{iK}$ , we build the patch  $\vec{X}_i = [\vec{x}_i, \vec{x}_{i1}, \dots, \vec{x}_{iK}]$ . For  $\vec{X}_i$ , we have a part mapping  $f_i : \vec{X}_i \mapsto \vec{Y}_i$ , by which we can obtain  $\vec{Y}_i = [\vec{y}_i, \vec{y}_{i1}, \dots, \vec{y}_{iK}]$ , the low-dimensional representation of  $\vec{X}_i$ . Let us perform the part optimization on  $\vec{Y}_i$  as follows:

$$\arg \min_{\vec{Y}_i} \text{tr} \left( \vec{Y}_i \vec{L}_i \vec{Y}_i^T \right), \tag{1}$$

where  $\text{tr}(\cdot)$  is the trace operator;  $\vec{L}_i \in R^{(K+1) \times (K+1)}$  is the patch optimization matrix and varies with the different algorithms.

### 2.2 Whole alignment

For each patch  $\vec{X}_i$ , its low-dimensional representation  $\vec{Y}_i$  can be put into a global coordinate as follows:

$$\vec{Y}_i = \vec{Y} \vec{S}_i, \tag{2}$$

where  $\vec{Y} = [\vec{y}_1, \dots, \vec{y}_N]$  is the global coordinate,  $\vec{y}_i$  can be obtained by  $f_i: \vec{x}_i \mapsto \vec{y}_i$  for  $i = 1, \dots, N$ , and  $\vec{S}_i$  is a selection matrix. The entry of  $\vec{S}_i$  is defined as below:

$$(\vec{S}_i)_{pq} = \begin{cases} 1 & \text{if } p = F_i \{q\} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where  $F_i = \{i, i_1, \dots, i_K\}$  denotes the index set of the  $i^{th}$  patch, which is built by the training sample  $\vec{x}_i$  and its  $K$  associated ones. Then, Eq. (1) can be rewritten as follows:

$$\arg \min_{\vec{Y}} \text{tr} \left( \vec{Y} \vec{S}_i \vec{L}_i \vec{S}_i^T \vec{Y}^T \right). \tag{4}$$

Now, we can perform the whole alignment. Let  $i = 1, \dots, N$  and sum over all the part optimizations described as Eq. (4) as follows:

$$\arg \min_{\vec{Y}} \sum_{i=1}^N \text{tr} \left( \vec{Y} \vec{S}_i \vec{L}_i \vec{S}_i^T \vec{Y}^T \right). \tag{5}$$

Impose  $U^T U = I_d$  in Eq. (5). The optimal projection matrix  $U$  is obtained by solving the following objective function:

$$\begin{aligned} & \arg \min_U \sum_{i=1}^N \text{tr} \left( U^T \vec{X} \vec{S}_i \vec{L}_i \vec{S}_i^T \vec{X}^T U \right) \\ & \text{s.t. } U^T U = I_d \end{aligned} \tag{6}$$

## 3 Marginal patch alignment (MPA)

### 3.1 Motivations

Let us first consider a two-class problem.  $X = [x_1, x_2, \dots, x_N]$  is the training sample set, where  $N$  is the total training number. Assumed that  $x_i$  is the sample from Class 1. Now, we find its  $K2$ -nearest samples in Class 2. Collect these  $K2$ -nearest neighbors in  $X_i^2 = [x_{i1}, \dots, x_{iK2}]$  and calculate their local mean vector  $m_i^2$ . Now, we turn back to find the  $K1$ -nearest neighbors of  $m_i^2$  in Class 1 and collect them in  $X_i^1 = [x_i^1, \dots, x_{iK1}]$ . The mean of these  $K1$  samples is denoted by  $m_i^1$ . Collect the two parts of neighbors of  $x_i$  together and build the  $i^{th}$  marginal patch as  $X_i = [x_i, x_{i1}, \dots, x_{iK1}, x_{i1}, \dots, x_{iK2}]$ .

Now, we define the between-class marginal distance of  $x_i$  as follows:

$$d_b(x_i) = \left\| m_i^2 - m_i^1 \right\|^2. \tag{7}$$

The within-class marginal distance of  $x_i$  is defined as follows:

$$d_w(x_i) = \sum_{j=1}^{K1} \left\| x_i - x_{ij} \right\|^2 (w_i)_j \tag{8}$$

where  $(w_i)_j = \exp(-\|x_i - x_{ij}\|^2 / \partial)$ ,  $\partial$  is a parameter. Usually, we can choose the parameter  $\partial$  as the square of the average Euclidean distance between  $x_i$  and all its within-class marginal data points. More details above the selection of the parameter can refer (Belkin and Niyogi 2001).

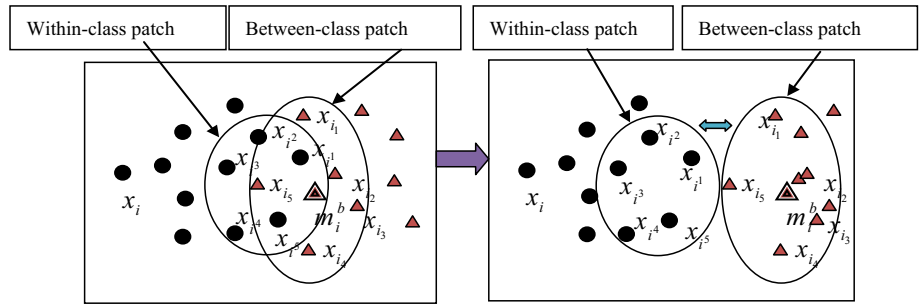
Our goal is to find a linear transform matrix  $P$ , such that the transformed marginal distance between the two difference classes can be enlarged, i.e., the transformed samples with different class-labels are push away. Meanwhile, the within-class samples are pulled closer. Figure 1 shows the optimized result.

### 3.2 Marginal patch optimization

In the transformed space, the within-class marginal distances  $d_w(y_i)$  generally characterizes the compactness of the marginal patch  $Y_i = [y_i, y_{i1}, \dots, y_{iK1}, y_{i1}, \dots, y_{iK2}]$ . By performing algebraic operation,  $d_w(y_i)$  can be deduced as follows:

$$\begin{aligned} d_w(y_i) &= \sum_{j=1}^{K1} \left\| y_i - y_{ij} \right\|^2 (w_i)_j \\ &= \text{tr} \left\{ \begin{pmatrix} y_i - y_{i1} \\ \vdots \\ y_i - y_{iK1} \end{pmatrix} \text{diag} (w_i^t) \begin{bmatrix} y_i - y_{i1}, \dots, y_i - y_{iK1} \end{bmatrix} \right\} \\ &= \text{tr} \left( Y_i L_{w(i)} Y_i^T \right) \end{aligned} \tag{9}$$

**Fig. 1** Illumination of the optimized result



where  $L_{w(i)} = \begin{pmatrix} I_{K1+1} \\ 0 \end{pmatrix} \begin{pmatrix} -e_{K1}^T \\ I_{K1} \end{pmatrix} \text{diag}(w_i)[-e_{K1}, I_{K1}]$   
 $L_{w(i)} = \begin{pmatrix} I_{K1+1} \\ 0 \end{pmatrix}^T$ ,  $e_{K1} = [1, \dots, 1]^T \in R^{K1 \times 1}$ , and  
 $I_{K1} = \text{diag}(1, \dots, 1) \in R^{K1 \times K1}$

Similarly, the between-class marginal distance  $d_b(y_i)$  in the transformed space characterizes the separability of the marginal samples with different class-labels.  $d_b(y_i)$  can be deduced as follows:

$$\begin{aligned} d_b(y_i) &= \left\| \hat{m}_i^1 - \hat{m}_i^2 \right\|^2 \\ &= \left\| \frac{1}{K1} \sum_{j=1}^{K1} y_{ij} - \frac{1}{K2} \sum_{s=1}^{K2} y_{is} \right\|^2 \\ &= \text{tr} \left( Y_i \Omega_i \Omega_i^T Y_i^T \right) \\ &= \text{tr} \left( Y_i L_{b(i)} Y_i^T \right) \end{aligned} \tag{10}$$

where  $L_{(b)i} = \Omega_i \Omega_i^T$  and  $\Omega_i = \begin{bmatrix} 0 & 0 \\ 0 & I_{K1+K2} \end{bmatrix}$   
 $[0, \underbrace{\frac{1}{K1}, \dots, \frac{1}{K1}}_{K1}, \underbrace{-\frac{1}{K2}, \dots, -\frac{1}{K2}}_{K2}]^T$ .

**3.3 Whole alignment**

For better classification, the marginal distance between the two categories in the low-dimensional transformed space should be as large as possible. To achieve this goal, for each marginal patch, the distance from each sample to its corresponding within-class marginal points should be as small as possible, i.e., maximizing  $d_w(y_i)$ . At the same time, the distance between the two means of marginal patches with different class-labels should be as large as possible, i.e, maximizing  $d_b(y_i)$ . Unifying both two distance functions, we have

$$\begin{aligned} &\arg \max_{y_i} (d_b(y_i) - \beta d_w(y_i)) \\ &= \arg \max_{y_i} \left( \left\| \frac{1}{K1} \sum_{j=1}^{K1} y_{ij} - \frac{1}{K2} \sum_{s=1}^{K2} y_{is} \right\|^2 \right. \end{aligned}$$

$$\begin{aligned} &\left. - \beta \sum_{j=1}^{K1} \|y_i - y_{ij}\|^2 (w_i)_j \right) \\ &= \arg \max_{y_i} \text{tr} \left( Y_i L_{b(i)} Y_i^T \right) - \beta \text{tr} \left( Y_i L_{w(i)} Y_i^T \right) \end{aligned} \tag{11}$$

where the parameter  $\beta$  is a scale factor to adjust the tradeoff between the two measurements of the within-class distance and the between-class distance (i.e., the compactness and separability) (Zhang et al. 2009). The main factor that can cause the imbalance is the unequal numbers  $K1$  and  $K2$ , of the same-class and different-class marginal neighbors. In our paper, we simply set  $\beta = 1$  to place the same weight on two distances. In addition, we can set  $\beta$  in the range of  $[0, 1]$ . If  $\beta = 0$ , obviously, the within-class marginal distances will be neglected and the between-class marginal distances become very crucial. The marginal discriminative information can only be captured by exploring the between-class marginal structure. In such a case, an appropriate  $K2$  becomes very important. When  $K2 = T$ , where  $T$  is the number of training sample per class, the between-class local marginal mean is the class-mean. The class-mean contains less class marginal information. So  $K2$  should not be set too large. On the contrary, if we set  $K2$  too small, such as 1, that means only a very small amount of training samples are utilized in the learning the discriminative structure information, which may lead to suboptimal performance. Zhang et al. (2008) provides us a more simple way to choose parameters  $K1$  and  $K2$ . We can follow it to find the optimal parameters  $K1$  and  $K2$ . Based on the optimal parameters  $K1$  and  $K2$ , the sensitivity of objective function caused by  $\beta$  can be reduced to some extent.

Having finished the marginal patch optimization, we now perform the whole alignment. The low-dimensional representation of the  $i^{th}$  patch  $Y_i = [y_i, y_{i1}, \dots, y_{iK1}, y_{i1}, \dots, y_{iK2}]$  can be integrated into a global coordinate  $Y = [y_1^T, y_2^T, \dots, y_N^T]$  by a selection matrix  $S_i$ , i.e,

$$Y_i = Y S_i, \tag{12}$$

where  $Y = P^T X$  and

$$(S_i)_{pq} = \begin{cases} 1 & \text{if } p \in \Gamma_i(q) \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

and  $\Gamma_i = \{i, i^1, \dots, i^{K1}, i_1, \dots, i_{K2}\}$ .

Let  $i = 1, \dots, N$  and we have  $N$  sub-optimizations of Eq. (11). Sum up all of them and then perform the whole alignment as follows:

$$\begin{aligned} & \arg \max_Y \sum_{i=1}^N \left( \text{tr} \left( Y S_i L_{b(i)} S_i^T Y^T \right) - \beta \text{tr} \left( Y S_i L_{w(i)} S_i^T Y^T \right) \right) \\ &= \arg \max_Y \text{tr} \left( Y \sum_{i=1}^N \left( S_i L_{b(i)} S_i^T - \beta S_i L_{w(i)} S_i^T \right) Y^T \right) \\ &= \arg \max_Y \text{tr} (Y L Y^T) \end{aligned} \tag{14}$$

where  $L$  is the alignment matrix and

$$L = \sum_{i=1}^N \left( S_i L_{b(i)} S_i^T - \beta S_i L_{w(i)} S_i^T \right). \tag{15}$$

The optimal transformed matrix  $P_{\text{MPA}} = [p_1, \dots, p_d]$  is constructed by the eigenvectors of

$$X L X^T P = \lambda P \tag{16}$$

associating with  $d$  maximal eigenvalues  $\lambda$ .

### 3.4 Algorithm of MPA

In summary of the description above, the marginal patch alignment (MPA) algorithm is given below:

- Step 1: Project all the training samples into a PCA-transformed subspace with the projective matrix of  $P_{\text{PCA}}$ .
- Step 2: For each sample  $x_i$ , find its  $K2$  between-class marginal samples  $X_i^2 = [x_{i_1}, \dots, x_{i_{K2}}]$  and  $K1$  within-class marginal neighbors  $X_i^1 = [x_{i_1}, \dots, x_{i_{K1}}]$ . And then build the  $i^{\text{th}}$  marginal patch  $X_i = [x_i, x_{i_1}, \dots, x_{i_{K1}}, x_{i_1}, \dots, x_{i_{K2}}]$ .
- Step 3: Calculate the alignment matrix  $L$  by Eq. (15).
- Step 4: Solve the standard eigenequation of  $X L X^T P = \lambda P$  and obtain the transformed matrix  $P_{\text{MPA}}$ .
- Step 5: Output the final linear transformed matrix  $P^* = P_{\text{PCA}} P_{\text{MPA}}$ .

Once  $P^*$  is obtained, we can project all the samples into the optimal transformed space with the projective matrix of  $P^*$ , and then select the minimum distance classifier (MDC; Gonzalez and Woods 1997) for classification.

### 3.5 Extension to multi-class case

Let us consider the multi-class cases in the observation space. Suppose there are  $c$  pattern classes. Let  $X = \{X_i^{l_i}\}$  be the

training sample set, where  $i = 1, \dots, N$ , and  $l_i \in \{1, \dots, c\}$  is the class-label of  $x_i$ . Now, we need to adjust some steps of the MPA algorithm in two-class case. More concretely, in step 2, we turn to find the  $K2$  between-class marginal neighbors in Class  $s$  (where  $s = 1, \dots, c$  and  $s \neq l_i$ ), and collect them in  $(\chi_b)_i^s$ . Calculate the mean of these between-class marginal neighbors  $(m_b)_i^s$  and find  $K1$  nearest neighbors of  $(m_b)_i^s$  in Class  $l_i$ . The  $K1$  nearest neighbors is called the within-class marginal samples of  $x_i$  with respect to Class  $s$ . Collect them in  $(\chi_w)_i^s = \{x_{i_j}^s\}$  for  $j = 1, \dots, K1$ .

Now, we calculate the within-class marginal distance and between-class marginal distance of  $y_i$  in the transformed space, respectively, as follows:

$$D_w(y_i) = \sum_{l=1, l \neq l_i}^c \sum_{j=1}^{K1} \|y_i - y_{ij}^l\| \left( w_i^l \right)_j, \tag{17}$$

$$D_b(y_i) = \sum_{l=1, l \neq l_i}^c \left\| \frac{1}{K1} \sum_{j=1}^{K1} y_{ij}^l - \frac{1}{K2} \sum_{t=1}^{K2} y_{it}^l \right\|^2. \tag{18}$$

Perform the marginal patch optimization as follows:

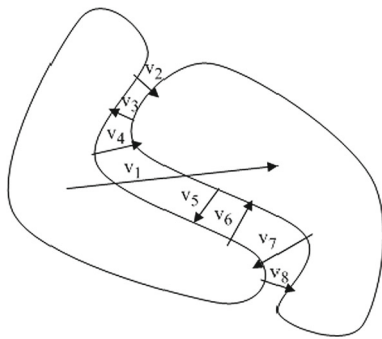
$$\arg \max_{y_i} (D_b(y_i) - \beta D_w(y_i)). \tag{19}$$

The remaining steps are the same as the ones in two-class case.

## 4 Advantages of MPA

The proposed MPA has the following advantages:

- The contributions of each sample for optimal subspace selection are distinguished in the phase of marginal patch optimization.
- The number of the available features is more than  $c-1$ , where  $c$  is the number of classes. In the high-dimensional space, it usually needs more features to separate the classes well.
- The marginal samples contain more discriminative information than those in other patches. MPA takes the marginal samples into account, and thus can well preserve marginal discriminability of classes.
- Without prior information on data distributions, ‘‘marginal patch optimization’’ can characterize the separability of different classes well and the ‘‘whole alignment’’ can help us to obtain the global optimization.
- Like the nonlinear dimensionality reduction method, MPA can find the nonlinear boundary structural information for different classes hidden in the high-dimensional data, even the data with non-Gaussian distribution. This



**Fig. 2** MPA's between-class scatter and LDA's between-class scatter.  $v_1$ : difference vector of the centers of two classes;  $\{v_2, \dots, v_8\}$ : difference vectors from the local means located in the classification boundary

can be explained by examining the vectors. As illustrated in Fig. 2, MPA has the advantage of utilizing the boundary information. More concretely, the MPA's between-class scatter matrix spans a space involving the subspace spanned by the vectors  $\{v_2, \dots, v_8\}$  where boundary structure is embedded. Therefore, the boundary information can be fully utilized. We compare MPA with LDA, and find LDA computes the between-class scatter matrix only using the vector  $v_1$ , which is merely the difference between the centers of the two classes. It is obvious that  $v_1$  fails to capture the boundary.

## 5 Experiments

In this section, the proposed MPA method is evaluated using the Yale face database, the UCI Wine dataset, the Yale-B face database, and the AR face database and compared with PCA, FLDA, LPP, maximum margin criterion (MMC; Li et al. 2006), and DLA. For LPP, we select the Gaussian kernel  $t = \infty$ . Note that, in our experiment, LPP and MPA use the  $K$ -nearest neighbor algorithm to select the neighbors. The neighbor parameters are set by the global searching strategy. For distinction, the neighbor parameter is denoted by  $K$  in LPP, the within-class and the between-class neighbor parameters in MPA are denoted as  $K_1$  and  $K_2$ , respectively. After extracting features, MDC is employed for classification. The experiments are executed on a computer system of Intel(R) Core(TM) i5-4440 CPU @ 3.10GHz 3.10GHz and 8.00 GB with Matlab R2010a.

### 5.1 Experiments on the Yale face database

The Yale face database is constructed at the Yale Center for Computational Vision and Control. It contains 165 grayscale images of 15 individuals. The images demonstrate the variations in lighting condition, facial expression, and

with/without glasses. The size of each cropped image is  $100 \times 80$ . Figure 3 shows some sample images of one individual. These images vary as follows: center-light, with glasses, happy, left-light, without glasses, normal, right-light, sad, sleep, surprised, and winking.

In the following section, we continue our experiments with random training samples.  $T$  ( $=3, 4, 5$ , and  $6$ ) images are randomly selected from the image gallery of each individual to form the training sample set. The remaining  $11-T$  images are used for test. The selection of neighbor parameters of LPP and MPA is similar to that in UCI Wine database. For FLDA, LPP, MMC, and MPA, we first project the data set into a 44-dimensional PCA subspace. We independently run the system 20 times. Table 1 shows the average recognition rates and their corresponding 95% intervals of each method. The maximal average recognition rates of six methods versus the variation of the size of training set are illustrated in Fig. 4. From Table 1 and Fig. 4, we can see that:

1. MPA significantly outperforms PCA, FLDA, LPP, MMC, and DLA, no matter with the variation of the size of training samples in each class.
2. By comparing with the column of Table 2 with respect to FLDA, we find its recognition rate has degraded a lot when the number of training sample increases to 4. The possible reason is that FLDA is sensitive to outliers. Compared with the "center-light" image, the "left-light" and the "right-light" images of each class can be treated as outliers. The outlier images may affect the class-mean and cause error in estimate of scatters. This will finally lead to the projection of FLDA inaccurate (Xu et al. 2014). In contrast, MPA builds the adjacency relationship of data points using  $K_2$  between-class nearest marginal neighbors and the  $K_1$  within-class marginal neighbors. Most outlier images of different persons lie on the margin of class manifold. MPA focuses on the classification margin. Thus, most outliers may be treated as the marginal points and used to calculate the between-class scatter (i.e., the between-class marginal distance). At the same time, the outliers may be treated as the within-class marginal samples. By minimizing the within-class marginal distance, the same-class samples cluster better than before, and the marginal distance between the different classes also become larger. From this point of view, MPA seems to be more robust to outliers than FLDA.

### 5.2 Experiments on the Wine dataset from UCI

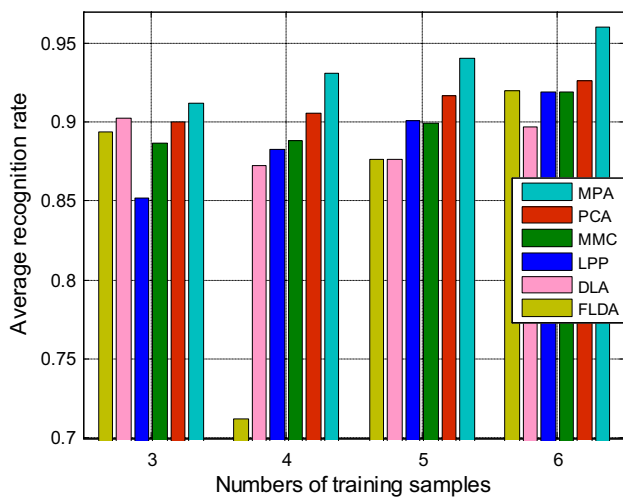
Wine dataset is a real-life dataset from the UCI machine learning repository (<http://archive.ics.uci.edu/ml>). Wine has 13 features, 3 classes and 178 instances. 48 instances of each class are selected and used in the experiments.



**Fig. 3** Samples of a person from the Yale database

**Table 1** The maximal average recognition rates (%) and t and the corresponding 95 % intervals of six methods on the Yale face database

	PCA	FLDA	LPP	MMC	DLA	MPA
3	90.00 [0.87, 0.93]	89.42 [0.80, 0.95]	85.17 [0.83, 0.93]	88.67 [0.87, 0.96]	90.25 [0.86, 0.95]	91.21 [0.85, 0.96]
4	90.57 [0.87, 0.93]	71.24 [0.53, 0.79]	88.24 [0.87, 0.95]	88.81 [0.86, 0.95]	87.24 [0.76, 0.93]	93.10 [0.86, 0.97]
5	91.67 [0.84, 1.00]	87.67 [0.73, 0.99]	90.11 [0.83, 1.00]	89.94 [0.83, 1.00]	87.67 [0.71, 0.98]	94.06 [0.83, 1.00]
6	92.60 [0.85, 0.99]	92.00 [0.84, 1.00]	91.93 [0.83, 1.00]	91.87 [0.86, 1.00]	89.67 [0.77, 0.99]	96.00 [0.88, 1.00]



**Fig. 4** The maximal average recognition rates of six methods versus the variation of the size of training set in the Yale face database

In our experiments,  $T$  ( $=10, 15, 20, 25$ , and  $30$ ) samples per class are randomly selected for training and the remainders for testing. The experiment is repeated for 30 times. PCA, FLDA, LPP, MMC, DLA, and the proposed MPA algorithm are, respectively, used for feature extraction based on the original 13-dimensional features. We set the range of neighbor parameter  $K$  of LPP from 2 to  $3T - 1$  with an interval of 1. Similarly, the range of between-class parameter  $K_2$  of MPA is set from 1 to  $T$  with an interval of 1, and the within-class parameter  $K_1$  of MPA is set from 1 to  $T - 1$  with an interval of 1. And then, we choose the neighbor parameter associated

with the top recognition rate as the optimal one. Table 2 lists the maximal average recognition rates and the 95 % interval of each method on the UCI Wine database across 30 runs. The maximal average recognition rates of six comparative methods versus the variation of the size of training set are illustrated in Fig. 5. Observing Table 2 and Fig. 5, we find MPA is the best one among six feature extraction methods, irrespective of the variation in the size of training samples in each class.

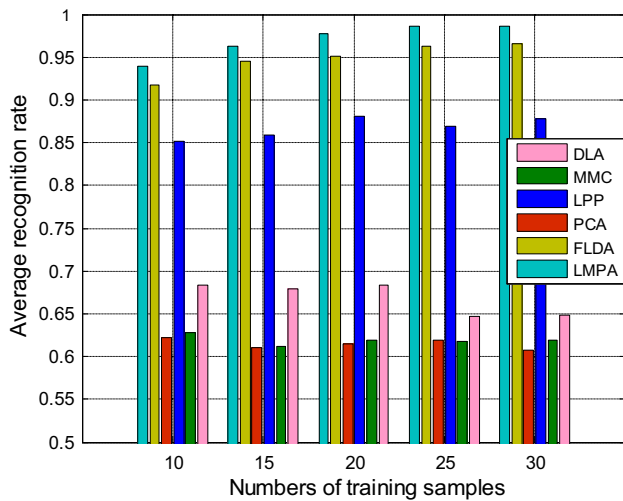
### 5.3 Experiments on the extended Yale-B face database

The extended Yale B face database (Lee et al. 2005) contains 38 human subjects under 9 poses and 64 illumination conditions. The 64 images of a subject in a particular pose are acquired at camera frame rate of 30 frames/ second, so there is only small change in head pose and facial expression for those 64 images. All frontal-face images marked with P00 are used, and each image is resized to  $42 \times 48$  pixels in our experiment. Some sample images of one person are shown in Fig. 6.

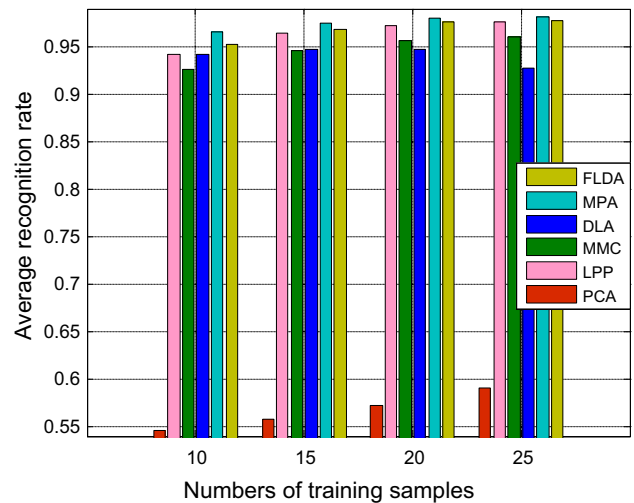
In our experiments,  $T$  ( $=10, 15, 20$ , and  $25$ ) images are randomly selected from the image gallery of each individual to form the training sample set. The remaining images are used for testing. PCA, FLDA, LPP, MMC, DLA, and MPA are, respectively, used for feature extraction. Note that, FLDA, LPP, MMC, DLA, and MPA are performed on the 150-dimensional PCA subspace. Similarly, the neighbor parameters of MPA are set as that used in the UCI Wine dataset. In addition, considering the case of large training

**Table 2** The maximal average recognition rates (%) and the corresponding 95 % intervals of six methods on the UCI Wine database

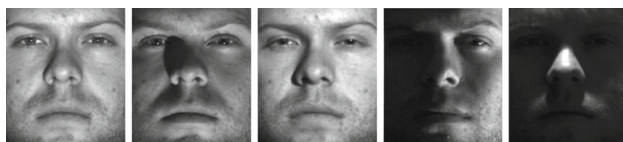
	PCA	FLDA	LPP	MMC	DLA	MPA
10	62.16 [0.59, 0.68]	91.73 [0.84, 0.96]	85.20 [0.77, 0.94]	62.78 [0.59, 0.71]	68.33 [0.54, 0.78]	93.95 [0.88, 0.97]
15	61.08 [0.59, 0.64]	94.58 [0.90, 0.98]	85.89 [0.78, 0.93]	61.14 [0.58, 0.64]	67.91 [0.56, 0.77]	96.33 [0.94, 0.98]
20	61.51 [0.56, 0.64]	95.12 [0.92, 0.98]	88.06 [0.81, 0.93]	61.98 [0.57, 0.66]	68.41 [0.61, 0.77]	97.78 [0.95, 0.99]
25	61.87 [0.55, 0.69]	96.23 [0.93, 0.99]	86.86 [0.78, 0.93]	61.84 [0.55, 0.69]	64.64 [0.54, 0.74]	98.60 [0.97, 1.00]
30	60.80 [0.57, 0.65]	96.54 [0.92, 1.00]	87.72 [0.77, 0.94]	61.85 [0.57, 0.67]	64.81 [0.51, 0.80]	98.58 [0.96, 1.00]



**Fig. 5** The maximal average recognition rates of PCA, FLDA, LPP, MMC, DLA, and MPA versus the variation of the size of training set in UCI Wine dataset



**Fig. 7** The maximal average recognition rates of PCA, FLDA, LPP, MMC, DLA, and MPA versus the variation of the size of training set in the Yale-B face database



**Fig. 6** Samples of a person under pose 00 and different illuminations, which are cropped images in the extended Yale-B face database

samples, we take the strategy of searching from global to local. Let neighbor parameter of LPP vary from 2 to  $38T - 1$  with an interval of 5, and select the parameter associated with the best recognition rate. After that, we reset the range to a smaller one which surrounds the parameter associated with the best recognition rate. The optimal neighbor parameter is

the one associated the top recognition rate. For each  $T$ , we run the system 20 times and record the top average recognition rate and the corresponding 95 % interval of each method in Table 3. Figure 7 shows us the histogram of experimental results. By comparing the experimental results in Table 3 and Fig. 7, we find that the top recognition rate of MPA is also higher than other methods. It is well known that the illumination variation problem is one of the well-known problems in face recognition in uncontrolled environment. The experimental results in Yale-B face database verify the validity of the proposed method in dealing the changing illumination on images.

**Table 3** The maximal average recognition rates (%) and the corresponding 95 % intervals of six methods on the Yale-B face database

	PCA	FLDA	LPP	MMC	DLA	MPA
10	54.58 [0.44, 0.68]	95.23 [0.90, 0.98]	94.17 [0.87, 0.98]	92.59 [0.86, 0.97]	94.25 [0.87, 0.97]	96.57 [0.94, 0.99]
15	55.83 [0.46, 0.63]	96.84 [0.92, 0.99]	96.41 [0.90, 0.98]	94.65 [0.87, 0.98]	94.78 [0.86, 0.98]	97.55 [0.95, 0.99]
20	57.22 [0.48, 0.65]	97.59 [0.96, 0.99]	97.21 [0.95, 0.99]	95.65 [0.92, 0.98]	94.67 [0.86, 0.98]	98.05 [0.97, 0.99]
25	59.12 [0.51, 0.65]	97.78 [0.97, 0.99]	97.67 [0.97, 0.99]	96.05 [0.92, 0.98]	92.77 [0.88, 0.98]	98.12 [0.97, 1.00]





**Fig. 8** Sample images of one person in the AR face database

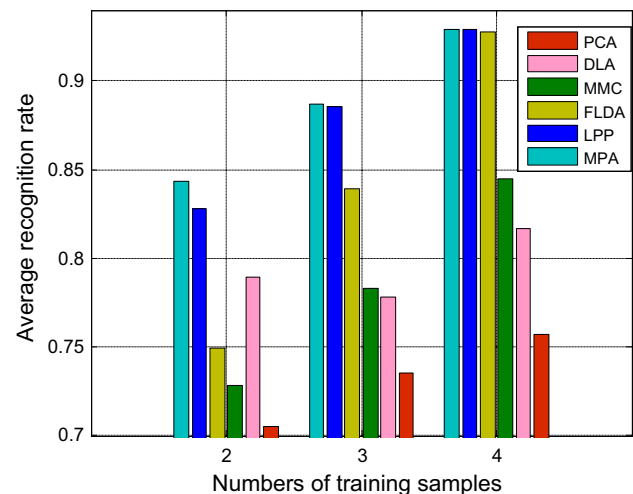
**Table 4** The maximal average recognition rates (%) and the corresponding 95 % intervals of six methods on the AR database across 10 runs

	PCA	FLDA	LPP	MMC	DLA	MPA
2	70.51 [0.66, 0.77]	74.94 [0.63, 0.84]	82.84 [0.75, 0.90]	72.87 [0.66, 0.81]	78.92 [0.74, 0.88]	84.34 [0.77, 0.89]
4	73.52 [0.66, 0.84]	83.93 [0.75, 0.93]	88.54 [0.84, 0.95]	78.30 [0.67, 0.92]	77.85 [0.63, 0.88]	88.73 [0.83, 0.95]
6	75.71 [0.65, 0.87]	92.75 [0.77, 0.99]	92.92 [0.84, 0.98]	84.18 [0.69, 0.94]	81.69 [0.55, 0.93]	92.93 [0.87, 0.98]

#### 5.4 Experiments on the AR face database

The AR face database (Martinez and Benavente 1998, 2006) contains over 4000 color face images of 126 people, including frontal views of faces with different facial expressions, lighting conditions and occlusions. The images of 120 persons including 65 males and 55 females are selected and used in our experiments. The pictures of each person are taken in two sessions (separated by two weeks) and each section contains seven color images without occlusions. The face portion of each image is manually cropped and then normalized to  $5014 \times T40$  pixels. The sample images of one person are shown in Fig. 8.

We randomly choose  $T$  ( $= 2, 4,$  and  $6$ ) images from the images gallery of each individual to form the training sample set. The remaining  $14 - T$  images are used for testing. The proposed algorithm is compared with PCA, FLDA, LPP, DLA, and MMC. For fair comparisons, 220 principal components are kept for FLDA, LPP, MMC, DLA, and MPA methods in the PCA step. For each given  $T$ , we average the results over 10 random splits and report the means in Table 4. The maximal average recognition rates of six methods versus the variation of the size of training set are illustrated in Fig. 9. Observing Table 4 and Fig. 9, we find that MPA can achieve higher recognition rates than PCA, FLDA, MMC, and DLA no matter with the variation of the size of training samples in each class. But MPA and LPP achieve the comparative results.



**Fig. 9** The maximal average recognition rates of PCA, FLDA, LPP, MMC, DLA, and MPA versus the variation of the size of training set in the AR face database

## 6 Conclusions

Under the PA framework, we developed a dimensionality reduction technique, termed MPA. In MPA, the marginal points associated with each training sample are selected to build the local patches. By performing patch optimization, we distinguish the contributions of each sample for optimal subspace selection by exploiting the marginal structure

information of each sample to extract the useful discriminative features. As a result, the marginal distances between the two different categories are accordingly enlarged in the low-dimensional transformed subspace. The following whole alignment unifies all of local patches into a globally linear system and makes the algorithm obtain the whole optimization. The proposed MPA method is evaluated using the Yale face database, the UCI Wine dataset, the Yale-B face database, and the AR face database. The experimental results demonstrate that MPA outperforms other comparative methods.

**Acknowledgements** This work was partially supported by the National Nature Science Foundation of China (Grant nos. 61305036, 61322306, 61333013, and 61273192), the China Postdoctoral Science Foundation funded project (Grant 2014M560657 and 2015T80898), Scientific Funds approved in 2013 for Higher Level Talents by Guangdong Provincial universities and Project supported by GDHVPS 2014.

#### Compliance with ethical standards

**Conflict of interest** Jie Xu, Shengli Xie and Wenkang Zhu, their immediate family, and any research foundation with which they are affiliated did not receive any financial payments or other benefits from any commercial entity related to the subject of this article.

## References

- Belhumeur PN, Hespanha JP, Kriegman DJ (1997) Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Trans Pattern Anal Mach Intell* 19(7):711–720
- Belkin M, Niyogi P (2001) Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in neural information processing systems*. MIT Press, Cambridge, pp 585–591
- Belkin M, Niyogi P (2003) Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput* 15(6):1373–1396
- Bengio Y, Paiement J, Vincent P (2003) Out-of-sample extensions for LLE, isomap, MDS, eigenmaps, and spectral clustering. in *Proc. Adv. Neural Inf. Process. Syst.* 177–184
- Fukunaga K (1990) *Statistical pattern recognition*. Academic Press, New York
- Fukunaga K (1991) *Introduction to Statistical Pattern Recognition*, 2nd edn. Academic Press, New York
- Gonzalez RC, Woods RE (1997) *Digital Image Processing*. Addison Wesley
- He X, Niyogi P (2003) Locality Preserving Projections. In: *Proceedings of the 16th conference on neural information processing systems*
- He X, Yan S, Hu Y, Niyogi P, Zhang H (2005) Face recognition using Laplacianfaces. *IEEE Trans Pattern Anal Mach Intell* 27(3):328–340
- He X, Cai D, Yan S, Zhang HJ (2005) Neighborhood preserving embedding. In *Proc. Int. Conf. computer Vision (ICCV'05)*
- Lee KC, Ho J, Kriegman DJ (2005) Acquiring linear subspaces for face recognition under variable lighting. *IEEE Trans Pattern Anal Mach Intell* 27(5):684–698
- Li H, Jiang T, Zhang K (2006) Efficient and robust feature extraction by maximum margin criterion. *IEEE Trans Neural Netw* 17(1):157–165
- Liu Q, Lu H, Ma S (2004) Improving kernel Fisher discriminant analysis for face recognition. *IEEE Trans Circuits Syst Video Technol* 14(1):42–49
- Martinez AM, Benavente R (1998) *The AR Face Database*. CVC Technical Report #24
- Martinez AM, Benavente R (2006) *The AR face database*. [http://rv11.ecn.purdue.edu/aleix/~aleix\\_face\\_DB.html](http://rv11.ecn.purdue.edu/aleix/~aleix_face_DB.html)
- Müller K, Mika S, Rietsch G, Tsuda K, Schölkopf B (2001) An introduction to kernel-based learning algorithms. *IEEE Trans Neural Netw* 12(2):181–201
- Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. *Science* 290:2323–2326
- Tenenbaum JB, deSilva V, Langford JC (2000) A global geometric framework for nonlinear dimensionality reduction. *Science* 290:2319–2323
- Xu J, Yang J, Gu Z, Zhang N (2014) Median-mean line based discriminant analysis. *Neurocomputing* 123:233–246
- Yan S, Xu D, Zhang B (2007) Graph embedding and extensions: a general framework for dimensionality reduction. *IEEE Trans Pattern Anal Mach Intell* 29(1):40–51
- Yang W, Wang J, Ren M, Yang J (2009) Feature extraction based on laplacian bidirectional maximum margin criterion. *Pattern Recogn* 42(11):2327–2334
- Yang W, Sun C, Zhang L (2011) A multi-manifold discriminant analysis method for image feature extraction. *Pattern Recogn* 44(8):1649–1657
- Zhang T, Tao DC, Yang J (2008) Discriminative locality alignment. In: *Proceedings of the 10th European Conference on Computer Vision (ECCV)*. Springer, Berlin, Heidelberg, pp 725–738
- Zhang T, Tao DH, Li XL, Yang J (2009) Patch alignment for dimensionality reduction. *IEEE Trans Knowl Data Eng* 21(9):1299–1313
- Zhang Z, Zha H (2004) Principle manifolds and nonlinear dimensionality reduction via local tangent space alignment. *SIAM J Sci Comput* 26(1):313–338