Pattern Recognition 43 (2010) 4165–4176

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

LPP solution schemes for use with face recognition

Yong Xu [a,*], Aini Zhong [a], Jian Yang [b], David Zhang [c]

[a] Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, China
[b] School of Computer Science & Technology, Nanjing University of Science & Technology, Nanjing, Jiangsu 210094, China
[c] The Biometrics Research Center, The Hong Kong Polytechnic University, Kowloon, Hong Kong

ARTICLE INFO

Article history:
Received 18 February 2010
Received in revised form
19 May 2010
Accepted 18 June 2010

Keywords:
Face recognition
Feature extraction
Locality preserving projection
Small sample size problems

ABSTRACT

Locality preserving projection (LPP) is a manifold learning method widely used in pattern recognition and computer vision. The face recognition application of LPP is known to suffer from a number of problems including the small sample size (SSS) problem, the fact that it might produce statistically identical transform results for neighboring samples, and that its classification performance seems to be heavily influenced by its parameters. In this paper, we propose three novel solution schemes for LPP. Experimental results also show that the proposed LPP solution scheme is able to classify much more accurately than conventional LPP and to obtain a classification performance that is only little influenced by the definition of neighbor samples.

1. Introduction

Locality preserving projection (LPP) is a manifold learning method [1–7] widely used in pattern recognition and computer vision. LPP is also well-known as a linear graph embedding method [8,9]. When LPP transforms different samples into new representations using the same linear transform, it tries to preserve the local structure of the samples, i.e., the neighbor relationship between samples [10–15] so that samples that were originally in close proximity in the original space remain so in the new space. We note that the original LPP method was unsupervised and was proposed for only vector samples, not being able to be directly applied to image samples. Here 'unsupervised' means that when producing the transforming axis the original LPP method does not exploit the class-label information. Hereafter this method is referred to as conventional LPP.

There have been several types of improvements to conventional LPP. The first type of improvement is supervised LPP [16–19], which seeks to improve the performance of LPP in recognition problems by exploiting the class-label information of samples in the training phase. The main difference between unsupervised LPP and supervised LPP is that unsupervised LPP uses only the distance metric between samples to determines 'neighbor samples' whereas supervised LPP uses both the distance metric and the class label of samples to determine 'neighbor samples'. Supervised LPP does not regard two samples from two different classes are 'neighbors' even if they are in close proximity to each other. Since the weight matrix is determined on the basis of neighbor relationship between samples, having different weight matrices is also one of the main differences between supervised LPP and unsupervised LPP. It is usually thought that supervised LPP can outperform unsupervised LPP in classification applications owing to the use of the class-label information. Local discriminant embedding (LDE) [20] and marginal Fisher analysis (MFA) [21] can also be viewed as supervised LPP methods. This is because their training phases both exploit the class-label information of samples. They are derived by using a motivation partially similar to LPP and each of them is based on an eigen-equation formally similar to the eigen-equation of LPP. On the other hand, since LDE and MFA partially borrow the idea of discriminant analysis and try to produce satisfactory linear separability, their ideas are also somewhat different from the idea of preserving the local structure of LPP. LDE and MFA can be viewed as two combinations of the locality preserving technique and the linear discriminant analysis. The two methods probably perform worse than the conventional supervised LPP in preserving the local structure.

The second type of improvement changes conventional LPP to a nonlinear transform method by using the kernel trick [19–24]. This type of improvement transforms a sample into a linear combination of a number of kernel functions each being determined by this sample and one training sample. The method uses the same linear combination coefficients to transform each sample into the new representation. Because the kernel function is nonlinearly related to the sample, the transform mapping is nonlinear. The third type of improvement to conventional LPP mainly focuses on directly implementing LPP for two-dimensional rather than one-dimensional vectors. This allows LPP to have a

* Corresponding author. Tel.: +86 752 26032458; fax: +86 752 26032461.
E-mail address: laterfall286@yahoo.com (Y. Xu).

0031-3203/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2010.06.016

higher computational efficiency. This type of improvement has been referred to as two-dimensional locality preserving projection (2DLPP) [25,26]. The fourth type of improvement to conventional LPP seeks to obtain LPP solutions with different solution properties. Examples of this type of improvement include orthogonal locality preserving method [27], uncorrelated LPP feature extraction method [28], the fast implementation algorithm for unsupervised orthogonal LPP [29], and the LPP algorithm for the SSS problem [30].

Different improvements to conventional LPP can also be regarded as implementations of the idea of locality preserving projection under different constraint conditions or in different cases. For example, unsupervised LPP requires all the samples to preserve their local neighbor relationship, whereas supervised LPP requires only samples from the same class to preserve their neighbor relationship. As conventional LPP was devised for vector data, its implementation on an image-based application requires that the image be converted into a vector in advance. However, 2DLPP is devised for image matrices, which means that 2DLPP directly applies the idea of locality preserving projection to image matrices rather than the vector corresponding to the image. Recently it has been demonstrated that LPP is theoretically related and formally similar to other linear dimensionality reduction methods and the main difference between LPP and them is in the weight matrix. Indeed, many popular linear dimensionality reduction methods including unsupervised LPP, supervised LPP, linear discriminant analysis (LDA), MFA, LDE and neighborhood preserving embedding (NPE) can be described as the implementations of the linear graph embedding framework with different weight matrices [31]. Conventional LPP and its improvements have been used in face recognition, image retrieval, document analysis, data clustering, etc. [11,16–18,32,33].

As in image-based applications, conventional LPP should first convert the image into the vector and as conventional LPP obtains the transforming axes by solving the minimum or maximum eigenvalue solution of a generalized eigen-equation, conventional LPP usually suffers from several problems. The first problem is that the dimensionality of the sample is usually larger than the number of the samples and the generalized eigen-equation cannot be directly solved due to the matrix singularity problem. This problem is also referred to as the small sample size (SSS) problem. An image-based recognition problem such as face recognition is usually a SSS problem. On the other hand, image-based recognition covers a wide range of pattern recognition problems. Thus, the study of how to properly apply LPP to the SSS problem is crucial. To the best of our knowledge, no satisfactory approach to this study has been proposed. Most of previous LPP-based image recognition applications avoid the SSS problem. For example, a number of face recognition applications of conventional LPP first reduce the size of the face image and then implement the conventional LPP algorithm for the resized images. In order to make the conventional LPP algorithm workable, the dimensionality of the vector of the resized image should be smaller than the number of the training samples. Consequently, in order to avoid the SSS problem, the original image usually should be resized into a very small size. This will cause a large quantity of image information loss. Another example of avoiding the SSS problem is to first reduce the dimensionality of the sample by performing principal component analysis and then to carry out the conventional LPP algorithm [23]. But there are no guidelines for how to use principal component analysis to transform the sample into a proper dimensionality. If the extent of reduction is too great, there will be considerable information loss. On the other hand, if the dimension reduction extent is small, the corresponding eigen-equation is still singular and cannot be solved directly.

A further drawback of conventional LPP is that if it is implemented by solving the minimum eigenvalue problem, the minimum eigenvalue solution is not always optimal for preserving the local structure. There are two reasons for this. The first is that if there are zero eigenvalues, conventional LPP will take as transforming axes the eigenvectors corresponding to the zero-eigenvalues of the generalized eigen-equation. As a result, after conventional LPP transforms samples into a new space using these transforming axes, a sample statistically will have the same representation as its neighbors, which will be formally demonstrated in Section 2. This is not how locality preserving projection works. The goal of LPP is not to make samples have the same representation but is to preserve the neighbor relationships between samples. The second reason is that the classifier cannot correctly classify samples when conventional LPP is implemented in the unsupervised case, since two neighbor samples from two different classes might have the same representation in the new space.

We also note that when implementing a LPP solution scheme, we should define a specific number of neighbor samples for each particular sample. In practice, it is not known how different values of this number influence the classification performance.

In this paper, we propose three new solution schemes for LPP. These new schemes have three advantages. The first is that they can be directly implemented no matter whether there exists the SSS problem or not. The second is that they are consistent with the goal of LPP and have a clear justification. The third advantage is that experimental results show that these schemes are more accurate than conventional LPP. This paper also conducts experiments to show the effect on classification performance of the number of neighbor samples and the value of the parameter k of k-nearest-neighbor classifiers (KNNC). The experimental results show that the improved LPP solution scheme 3 is not only computationally efficient, but also classifies much more accurately than conventional LPP. It is also a well-behaved LPP solution scheme whose classification accuracy is little influenced by the definition of neighbor samples.

The remainder of the paper is organized as follows. In Section 2 we introduce the algorithm of conventional LPP and analyze its characteristics. In Section 3 we present our LPP solution schemes and show their characteristics. In Section 4 we describe the experimental results. Section 5 offers our Conclusion.

2. Description of LPP

LPP was proposed as a way to transform samples into a new space and to ensure that samples that were in close proximity in the original space remain so in the new space. The goal of LPP is to minimize the following function:

$$\frac{1}{2}\sum_{ij}(y_i-y_j)^2 w_{ij}, \tag{1}$$

where y_i, y_j are transform results of vector samples $\mathbf{x}_i$, $\mathbf{x}_j$, and w_{ij} is the weight coefficient. y_i is obtained by using a transforming axis $\mathbf{z}$. That is, we have $y_i=\mathbf{x}_i^T\mathbf{z}$ and $y_j=\mathbf{x}_j^T\mathbf{z}$. The function defined in Eq. (1) can be rewritten as

$$\frac{1}{2}\sum_{ij}(\mathbf{z}^T\mathbf{x}_i-\mathbf{z}^T\mathbf{x}_j)^2 w_{ij}=\sum_{ij}\mathbf{z}^T\mathbf{x}_i w_{ij}\mathbf{x}_i^T\mathbf{z}-\sum_{ij}\mathbf{z}^T\mathbf{x}_i w_{ij}\mathbf{x}_j^T\mathbf{z}. \tag{2}$$

By defining a matrix $\mathbf{W}$and a dialog matrix $\mathbf{D}$ as $(\mathbf{W})_{ij}=w_{ij}$, $(\mathbf{D})_{ii}=\sum_j w_{ij}$ we can transform (2) into

$$\mathbf{z}^T\mathbf{X}(\mathbf{D}-\mathbf{W})\mathbf{X}^T\mathbf{z}=\mathbf{z}^T\mathbf{XLX}^T\mathbf{z}, \tag{3}$$