DISCRIMINANT ADAPTIVE EDGE WEIGHTS FOR GRAPH EMBEDDING

Yuan Yuan [1] and Yanwei Pang [2]

[1]School of Engineering and Applied Science, Aston University, Birmingham B4 7ET, U.K.
[2]School of Electronic Information Engineering, Tianjin University, Tianjin 300072, P. R. China

ABSTRACT

Many *linear dimensionality reduction* (LDR) methods, such as PCA and LDA, can be reformulated in the framework of *graph embedding* (GE). In this framework, those LDR methods are differentiated by values of edge weights of a graph. This paper first proposes a linear dimensionality reduction method, which assigns edges with discriminant adaptive weights. Specifically, we compute a local decision hyper-plane by using *support vector machine* (SVM). Then edge weighs corresponding to the local region are expressed as a function of the angle between the direction of the edges and the normal vector of the hyper-plane. Experimental results demonstrate the advantages of this proposed method.

Index Terms— Graph embedding, edge weights

1. INTRODUCTION

Dimensionality reduction is effective to deal with the case of *curse of dimensionality* (i.e., the exponential growth of hyper-volume as a function of dimensionality). It can also result in faster classifier and less storage [9]. Widely used linear DR algorithms include *principal component analysis* (PCA) [3], *linear discriminant analysis* (LDA) [4] and Nflsapce [17]. Powerful nonlinear reduction methods have emerged in recent years. *Locally linear embedding* (LLE) [5], ISOMAP [6], and Laplacian eigenmap [7] stand as the representative ones, which are also called manifold learning algorithms. In spite of effectiveness for extract nonlinear features, they cannot give analytic transformation functions. Though Bengio *et al.* [16] propose to deal with this *out-of-sample* problem by formulating some manifold learning algorithms as a variant of *kernel PCA* (KPCA), and their computation cost is much larger than that of linear methods, e.g., PCA, LDA. In this paper, we focus on linear cases.

Different linear feature extraction methods are derived by different objective function. PCA seeks a subspace that best represents the data in the sense of mean-square error. Utilizing discrimination criterion, LDA defines a projection that makes the within-class scatter small and the between-class scatter large. *Locality preserving projections* (LPP) finds an embedding that preserves local information [8]. LPP is highly related to Laplacian eigenmaps: the former is the linear version of the latter. Both PCA and LLE try to represent data with lest mean square error. The differences are that PCA attempts to preserve the global geometry of the data while LLE attempts to preserve the local geometry.

The above methods derived with different motivations, but they can be reformulated in a unified framework: *graph embedding* (GE) [1][2]. GE consists of three steps: 1) build a weighted graph, 2) derive a matrix from the weighted graph, and 3) compute the eigenvectors of the matrix. The graphs of LLE, Laplacian eigenmpas, Isomap, and LDA are sparsely connected, but the graph of PCA is fully connected. The edge weighs of PCA are identical. For LDA, the training points belonging to the same class are connected while the pairs of points belonging to different classes are not connected at all.

The sparseness of LDA's graph makes LDA outperform PCA in most cases in the sense of classification accuracy. The intra-connection and extra-disconnection properties indicate that LDA utilise the information carried by the class labels. However, the edge weighs of one class equal to those of any other classes. It is assumed that edges should be weighted according to the decision boundary so that the generalisation ability can be greatly enhanced. In this paper, the proposed method is named DAWGE, which stands for *discriminant adaptive edge weights for graph embedding*.

It is not easy to find exact decision boundary in either original data space or in low dimensional feature space, while we can estimate the piecewise boundary in a small local region. The boundary may not always be correct. However, if the estimated local boundary is not too far away the true one, it can give a useful guideline for designing the edge weights.

The rest of this paper is organised as follows: Section 2 briefly describes *graph embedding* framework, and shows the value of edge weights of PCA, LDA, and Laplacian eigenmaps. The proposed method (DAWGE) is reported in Section 3, and Section 4 gives experimental results. Section 5 concludes the paper.

2. GRAPH EMBEDDING

In pattern recognition system, dimensionality reduction can not only reduce the computational cost but also improve the classification accuracy. Graph embedding is a general framework for dimensionality reduction [1][2]. Many subspace learning algorithms such as PCA and LDA can be

reformulated in this framework. In this section we briefly describe graph embedding which is the basis our method.

Let $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N]$ be an input matrix with N being the number of samples and D being the dimensionality the samples. Each sample vector $\mathbf{x}_i$ belongs to one of the C object classes $\{\mathbf{X}_1,\ldots,\mathbf{X}_C\}$, and $l(\mathbf{x}_i)$ is the class label for $\mathbf{x}_i$. The linearisation of graph embedding aims to find a D by d transformation matrix $\mathbf{U}=[\mathbf{u}_1\ \mathbf{u}_2\ \ldots\ \mathbf{u}_d]$ so that $\mathbf{x}_i$ can be mapped to new point $\mathbf{y}_i$ by $\mathbf{y}_i = \mathbf{U}^T\mathbf{x}_i$.

Let $G=\{\mathbf{X},\mathbf{W}\}$ be an undirected weighted graph with vertex set $\mathbf{X}$ and the weight matrix $\mathbf{W}$. In the framework of graph embedding, the weight matrix $\mathbf{W}$ is computed in different ways for different embedding algorithms. Given the matrix $\mathbf{W}$, all the algorithms compute the diagonal matrix $\mathbf{D}$ with its elements $d_{ii} = \sum_j w_{ij}$ where w_{ij} is the ij -th entry of $\mathbf{W}$, and then compute the Laplacian matrix $\mathbf{L}$ using $\mathbf{L}=\mathbf{D}-\mathbf{W}$. The goal of the linearisation form graph embedding is to find the optimal transformation matrix

$$\mathbf{U}^* = \arg\min_{\mathbf{U}} \| \mathbf{U}\mathbf{x}_i - \mathbf{U}\mathbf{x}_j \|^2 \ w_{ij} . \quad (1)$$

In (1), w_{ij} is usually chosen such that nearby points in $\mathbb{R}^D$ can be mapped to nearby in the subspace spanned by $\mathbf{U}$.

The objective function of the linearisation form of graph embedding can be reduced to the following minimisation problem [1]:

$$\mathbf{U}^* = \min_{\substack{\mathbf{U}^T\mathbf{X}\mathbf{B}\mathbf{X}^T\mathbf{U}=z \\ or\ \mathbf{U}^T\mathbf{U}=z}} tr(\mathbf{U}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{U}) , \quad (2)$$

where $\mathbf{U}^T\mathbf{X}\mathbf{B}\mathbf{X}^T\mathbf{U}=\mathbf{z}$ and $\mathbf{U}^T\mathbf{U}=\mathbf{z}$ are proper constraints.

For LDA, the Laplacian matrix $\mathbf{L}$ equals to

$$\mathbf{L} = \mathbf{D} - \mathbf{W} = \mathbf{I} - \sum_{c=1}^{C} \frac{1}{N_c} \mathbf{e}^c \mathbf{e}^{c^T} , \quad (3)$$

where the diagonal matrix $\mathbf{D}=\mathbf{I}$ ($\mathbf{I}$ stands for the identity matrix), the weight matrix $\mathbf{W} = \sum_{c=1}^{C} \frac{1}{N_c} \mathbf{e}^c \mathbf{e}^{c^T}$, and $\mathbf{e}^c$ is an N dimensional vector with $e^c(i)=1$, if $c=l(\mathbf{x}_i)$; 0, otherwise. Specifically, the element of the $\mathbf{W}$ is

$$w_{ij} = \begin{cases} 1/N_c & if\ l(\mathbf{x}_i) = l(\mathbf{x}_j) = c \\ 0 & otherwise \end{cases} . \quad (4).$$

From (4) one can see that the graph of LDA is sparsely connected: the training points belonging to the same class are connected while the pairs of points belonging to different classes are not connected at all (i.e. their edge weights equal to zero).

For PCA, the Laplacian matrix $\mathbf{L}$ equals to

$$\mathbf{L} = \mathbf{D} - \mathbf{W} = \mathbf{I} - \frac{1}{N}\mathbf{e}\mathbf{e}^T , \quad (5)$$

where $\mathbf{D}=\mathbf{I}$, $\mathbf{W} = \frac{1}{N}\mathbf{e}\mathbf{e}^T$, and $\mathbf{e}$ is a summing vector with all its N elements being 1. Therefore, the edge weights w_{ij} of PCA equal to $1/N$ for any i and j. This means that the graph of PCA is fully connected and the edge weighs are identical.

3. ALGORITHM

3.1. Motivation

In the above section, we have introduced how the edge weights of PCA and LDA are computed. One can conclude that how the graph is connected and how the value of edges weights is given play an important role in the dimensionality reduction researches and applications.

Then a question should be answer: which the best edges weights for classification task. We attempt to develop a novel scheme to assign weights to the edges such that better classification accuracy can be obtained. It is assumed that the edges should be weighted according to the decision boundary (separating hyper-plane).

Specifically, the edge weight stage is monotonically decreasing with the angle between the normal vector of the separating hyper-plane and the edge under consideration. The motivation of this strategy is is inspired by the work by Hastile *et al.* in [10] and by Peng *et al.* in [11][12] which we will describe as follows. Their methods are variants of classical nearest neighbor classifier and do not focus on feature extraction and dimensionality reduction.

In [10], Hastile *et al.* have proposed a novel classifier: d*iscriminant adaptive nearest neighbor* (DANN). DANN determines the local decision boundaries from centroid information, and then shrink neighborhoods in directions orthogonal to these local decision boundaries, and elongate them parallel to the boundaries. Domeniconi *et al.* [12] and Peng *et al.* [11] proposed another local feature weighting scheme. They utilise local *support vector machine* (SVM) to estimate an effective metric for producing neighborhoods - elongated along less discriminant feature dimensions and constricted along most discriminant ones. Domeniconi *et al.* [12] have proved that this method increases the margin in the weighted space where nearest neighbor classification takes place. As a result, generalisation ability is enhanced. The algorithm is named as LAMANNA.

We adopt the advantages of the above ideas to graph embedding based feature extraction.

By feature weighting in a local region, both DANN and LAMANN shrink neighborhoods in directions perpendicular to local decision boundaries, and elongate them parallel to the boundaries. We mimic the process by assigning large weight value to the edge of the graph whose direction is orthogonal to the local decision boundary and by assigning less weight value to the edge whose direction is parallel to the local decision boundary. Specifically, the edge weight is monotonically decreasing with the angle between the normal vector of the decision boundary and the edge under consideration. From (1), we can see that the less the w_{ij}, the smaller the distance between $\mathbf{U}\mathbf{x}_i$ and $\mathbf{U}\mathbf{x}_j$. Take intra-connection and extra-disconnection properties into account, the net effect of our weighting is that the intra-distances are shrunk along the local boundaries and the

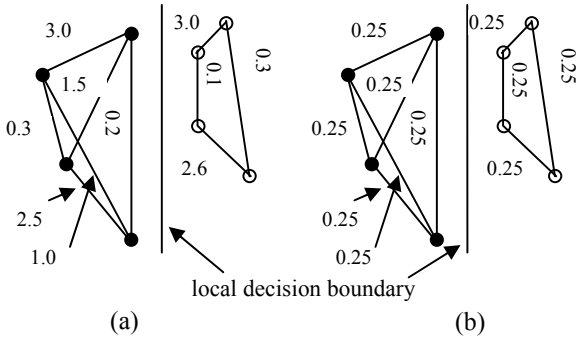

Fig. 1. Comparison of the proposed method with LDA. (a) The edge weights of DAWGE. (b) The edge weights of LDA

extra-distances are elongated consequently. It is expected that this will lead to better generalisation ability of the resulting low-dimensional features.

It is difficult to find exact decision boundary in either original data space or feature space (lower dimensionality than the original data space) which is one of the most important goal for classification. But we can estimate the piecewise boundary in a small local region (neighbors of one sample). The boundary may not always be correct. However, if the estimated local boundary is not too far away the true one, it can give a useful guide for designing the edge weights.

Fig. 1(a) illustrates a possible weight-assigning sample that embodies our idea. Note how the weight varies with the angle between its edge and the normal vector of the local decision boundary. For comparison, we show the edge weights of LDA in Fig. 1 (b), where the edge weights of the same class are equal everywhere.

3.2. The proposed method

Formally, the proposed method can be stated as follows. Suppose class c is concerned which has N_c training points $\mathbf{X}_c=[\mathbf{x}_1 \ \ldots \ \mathbf{x}_{N_c}]$ with $l(\mathbf{x}_1)=l(\mathbf{x}_2)=\ldots l(\mathbf{x}_{N_c})=c$, where $l(\mathbf{x}_i)$ represents the class label of $\mathbf{x}_i$. We find K nearest neighbors for class c. Denote $\mathbf{X}_{\bar{c}}$ the set of these neighbors. For $\mathbf{X}_{\bar{c}}$, its elements $\mathbf{x}_i$ and $\mathbf{x}_j$ satisfy $l(\mathbf{x}_i)\neq c$ and $l(\mathbf{x}_j)\neq c$. We can regard this problem as two-class problem with class c and non-c, i.e. $\bar{c}$.

By using linear *support vector machine* (SVM) [13], we can obtain the separating hyper-plane that separates class c and $\bar{c}$. Let $\mathbf{v}$ be the normal vector of the hyper-plane. Then we define the edge weight w_{ij} (where $l(\mathbf{x}_i)= l(\mathbf{x}_j)=c$) as

$$w_{ij} = f\left(\theta(\mathbf{v}, \overrightarrow{\mathbf{x}_i\mathbf{x}_j})\right) = \cos\left(\theta(\mathbf{v}, \overrightarrow{\mathbf{x}_i\mathbf{x}_j})\right) \quad i,j=1,\ldots,N_c \text{ and } i\neq j, \tag{6}$$

where $\overrightarrow{\mathbf{x}_i\mathbf{x}_j}$ stands for the vector formed by point $\mathbf{x}_i$ and $\mathbf{x}_j$, $0 \le \theta(\mathbf{v}, \overrightarrow{\mathbf{x}_i\mathbf{x}_j}) \le \pi$ is the angle between $\mathbf{v}$ and $\overrightarrow{\mathbf{x}_i\mathbf{x}_j}$, and f is a monotonically decreasing function with respect to the angle value. The cosine function in (6) is a possible instance of f.

Fig.2 Example images from AR database (before normalisation)

From the above formula one can find that the larger the angle between $\mathbf{v}$ and $\overrightarrow{\mathbf{x}_i\mathbf{x}_j}$ is, the larger the w_{ij} is. Thus (6) is consistent with the idea mentioned in subsection 3.2. Note that the function f in (6) is only one possible form and it can be taken in other forms. However, we do not focus on the optimal form of the function f in this paper.

Then, we let the diagonal elements of $\mathbf{W}$ are

$$w_{ii} = 1 - \sum_{\substack{l(\mathbf{x}_i)=l(\mathbf{x}_j) \\ i\neq j}} w_{ij}, \tag{7}$$

which guarantees that $d_{ii} = \sum_j w_{ij} = 1$ (d_{ii} is the diagonal elements of the matrix $\mathbf{D}$). Refer to Section 2 for graph embedding.

According to the theory of graph embedding (see Section 2), we can get the Laplacian matrix $\mathbf{L}=\mathbf{D}-\mathbf{W}$. By using the simplest constraint $\mathbf{u}^T\mathbf{u}=1$, (2) can be reformulated as

$$\mathbf{U}^* = \min_{\mathbf{U}^T\mathbf{U}=\mathbf{I}} tr(\mathbf{U}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{U}) \tag{8}$$

where $\mathbf{u}$ and $\mathbf{U}$ are defined in Section 2.
The solution to (8) can be found by solving the following eigenvalue problem:

$$\mathbf{L}\mathbf{u}_i = \lambda_i\mathbf{u}_i \tag{9}$$

with $\lambda_1<\lambda_2<\ldots<\lambda_d$.

Having obtained the transformation matrix $\mathbf{U}=[\mathbf{u}_1 \ \mathbf{u}_2 \ \ldots \ \mathbf{u}_d]$, we can extract the features of $\mathbf{x}_i$ by $\mathbf{y}_i = \mathbf{U}^T\mathbf{x}_i$. Finally, the nearest neighbor classifier is employed for pattern recognition.

The proposed *discriminant adaptive edge weights for graph embedding* method is abbreviated as DAWGE.

4. EXPERIMENTAL RESULTS

The AR face database [14] and FERET [15] face database were used to evaluate the proposed method (*DAWGE*), PCA and LDA. In our experiments, all the images were cropped based on the centres of eyes and resized to 42×42 pixels. They were normalized to have zero mean and unit variance. The parameters of each method are chosen so that it can approach its best performance. The nearest neighborhood classifier is adopted in the experiments.

Fig.3 Example images from FERET database (before normalization)

In the first experiment, 117 subjects were selected from a total of 126 subjects in the AR database. Only 14 non-occluded images (see Fig. 2 for example) per subject were used. Seven images of each subject were randomly chosen for training and the remaining seven images were used for testing. The system is run 15 times to obtaine 15 different training and testing sets. The average recognition rates of PCA, LDA, and DAWGE, are 79.82%, 91.55%, and 93.65% respectively. The results show that the proposed methods are significantly better than PCA and LDA.

In the second set of experiments, a subset of the FERET database is used. The subset includes 1394 images of 197 subjects with each of the subjects has 7 images (Fig. 3 shows example images). It is composed of the images whose names are marked with two-character strings: "ba", "bj","bk","be","bf","bd", and "bg". Three images of each subject were randomly chosen for training and the remaining four images were used for testing. We obtained 15 different training sets and testing sets. The average recognition rates of PCA, LDA and DAWGE are 49.22%, 67.30%, and 79.67% respectively. The proposed method is greatly superior to PCA and LDA.

These experiments demonstrate that proper design of edge weights can improve the classification performance of the graph embedding.

5. CONCLUSIONS

This paper proposed *discriminant adaptive edge weights for graph embedding* (DAWGE) as an improved method of graph embedding. The novelty of the DAWGE method lays in the discriminant adaptive edge weights. As we know, the *edge weights* of PCA and LDA are merely associated with the number of training samples of *either* the overall classes *or* per class. While, the *edge weights* of the proposed method are computed based on the normal vector of the separating plane and thus they are adaptive to discriminant direction. The edges of a graph are weighted so that the relevant edges for classification are emphasized. Therefore, it is expected and then has been demonstrated in two sets of experiments that the proposed DAWGE method has better classification performance. In the future, the advantages of tensor techniques will also be studied [18][19].

ACKNOWLEDGEMENTS

This research is partially supported by National Natural Science Foundation of China (grant number 60605005).

REFERENCES

[1] S. Yan *et al.*, "Graph Embedding: A General Framework for Dimensionality Reduction," *Proc. of IEEE Int'l Conf. CVPR*, vol. 2, pp. 830-837, 2005.

[2] M. Brand, "Minimax Embedding," *Advances in Neural Information Processing Systems*, vol.16, 2004.

[3] M. Turk and A. Pentland, "Eigenfaces for Recognition," *J. Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, 1991.

[4] P. N. Belhumeur *et al.*, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projections," *IEEE Trans. PAMI*, vol. 19, no. 7, pp. 771-720, 1997.

[5] S. Roweis and K. Saul, "Nonlinear Dimension Reduction by Locally Linear Embedding," *Science*, vol. 290, no. 5500, pp. 2323-2326, 2000.

[6] J. Tenenbaum *et al.*, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science*, vol. 290, pp. 2319-2322, 2000.

[7] M. Belkin and P. Niyogi, "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation," *Neural Computation,* vol. 14, pp. 1373-1396, 2003.

[8] X. He *et al.*, "Learning a Locality Preserving Subspace for Visual Recognition", *Proc. of IEEE Int'l Conf. ICCV*, 2003.

[9] A. K. Jain *et al.*, "Statistical Pattern Recognition – A Review," *IEEE Trans. PAMI*, vol. 22, no. 1, 2000.

[10] T. Hastie and R. Tibshirani, "Discriminant Adaptive Nearest Neighbor Classification," *IEEE Trans. PAMI*, vol. 18, no. 6, pp. 607-615, 1996.

[11] J. Peng *et al.*, "LDA/SVM Driven Neighbor Classification," *IEEE Trans. Neural Networks*, vol. 14, no. 4, pp. 940-942, 2003.

[12] C. Domeniconi *et al.*, "Large Margin Nearest Neighbor Classifiers," *IEEE Trans. Neural Networks*, vol. 16, no. 4, pp. 899-909, 2005.

[13] N. Cristianini and J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-based Learning Methods, Cambridge Press, 2000.

[14] A. Martinez and R. Benavente, The AR face database, CVC Technical Report #24, 1998.

[15] P. J. Phillips *et al.*, "The FERET Evaluation Methodology for Face-Recognition Algorithms," *IEEE Trans. PAMI*, vol. 22, no. 10, pp.1090-1104, 2000.

[16] Y. Bengio *et al.*, "Learning Eigenfunctions Links Spectral Embedding and Kernel PCA," *Neural Computation*, vol.16, no.10, pp.2197-2219, 2004.

[17] Y. Pang *et al.*, "Generalised Nearest Feature Line for Subspace Learning," *Electronics Letters*, vol. 43, no.20, pp. 1079-1080, 2007.

[18] X. Li *et al.*, "Discriminant Locally Linear Embedding with High Order Tensor Data," *IEEE Trans. SMC, Part B*, vol. 38, no. 2, 2008.

[19] D. Tao *et al.*, "General Tensor Discriminant Analysis and Gabor Features for Gait Recognition," *IEEE Trans. PAMI*, vol. 29, no. 10, pp. 1700-1715, 2007.