Noname manuscript No.
(will be inserted by the editor)

Appropriate Points Choosing for Subspace Learning over Image Classification

Shangguang Wang · Chuntao Ding

Received: date / Accepted: date

Abstract Dimension reduction techniques are very important, as high dimensional data is ubiquitous in many real world applications, especially in this era of big data. In this paper, we propose a novel supervised dimensionality reduction method, called appropriate points choosing based DAG-DNE (Apps-DAG-DNE). In Apps-DAG-DNE, we choose appropriate points to construct adjacency graphs, for example, it chooses nearest neighbors to construct inter-class graph, which can build a margin between samples if they belong to the different classes, and chooses farthest points to construct intra-class graph, which can establish relationships between remote samples if and only if they belong to the same class. Thus, Apps-DAG-DNE could find a good representation for original data. To investigate the performance of Apps-DAG-DNE, we compare it with the state-of-the-art dimensionality reduction methods on Caltech-Leaves and Yale datasets. Extensive experimental demonstrates that the proposed Apps-DAG-DNE outperforms other dimensionality reduction methods and achieves state-of-the-art performance for image classification.

Keywords Dimension reduction · Appropriate points · Farthest points · Adjacency graphs

Shangguang Wang
The State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications
E-mail: sgwang@bupt.edu.cn

Chuntao Ding
The State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications
E-mail: ctding@bupt.edu.cn

1 Introduction

In the era of Big Data, an increasing amount of image data are being generated on the Internet through daily social communication. Most real data are high-dimensional, such as image data. High-dimensional data significantly increases the time and space requirements, and also brings the curse of dimensionality problem. So it is necessary to develop efficient dimensionality reduction methods that can scale to massive amounts of high-dimensional data. Dimensionality reduction techniques have attracted considerable interest in machine learning [1,2] which not only reduce the computational complexity and avoid the curse of dimensionality problem, but also improve the classification performance in the subspace. Here, we focus on subspace learning, which pursuits the low-dimensional representation of the original high-dimensional data.

Dimensionality reduction methods are usually divided into unsupervised methods and supervised methods according to the availability of class label information. Popular unsupervised dimensionality reduction methods include principle component analysis (PCA) [28], locally linear embedding (LLE) [4]. It is generally known that PCA obtains the projection matrix by maximizing the total scatter of training samples. LLE reconstructs a given point by its neighbors to represent the local geometry structure and then seeks a low-dimensional embedding. However, LLE cannot perform mapping for an unseen data, which is called the out-of-sample problem. To cover the drawback of LLE, many new methods have been proposed, such as locality preserving projection (LPP)[3], neighborhood preserving embedding (NPE) [5], Both LPP and NPE find an embedding to preserve local structure and can be simply extended to unseen samples. However, all above methods cannot work well in classification tasks since they do not utilize the class label information.

In order to solve the problem of unsupervised method, many supervised methods are proposed, such as linear discriminant analysis (LDA) [6,7], discriminant neighborhood embedding (DNE) [8], marginal Fisher analysis (MFA) [12,13], locality-based discriminant neighborhood embedding (LDNE) [9], discriminant neighborhood structure embedding (DNSE) [11], double adjacency graphs-based discriminant neighborhood embedding (DAG-DNE) [10] and others [15–17,20,18,21,19,14]. LDA can find the projection matrix by maximizing the ratio between the inter-class scatter and the intra-class scatter. However, LDA fails to explore the manifold structure of the given data when projecting them into a lower-dimensional subspace. MFA, as an extension of LDA, by constructing inter-class and intra-class adjacency graphs to preserve the local information, which is able to efficiently solve the drawbacks of LDA. DNE constructs an adjacency graph to distinguish between homogeneous points and heterogeneous points to keep the local structure. However, DNE fails to preserve the detailed position relationship between the samples and their neighbors. Thus, the performance of DNE in the low-dimensional subspace is not good enough for classification tasks. LDNE optimizes the difference between the inter-class distance and the intra-class distance under

constructing the adjacency graph being different from DNE by endowing different weights. DAG-DNE constructs two adjacency graphs, aiming to keep the neighbors compact if they belonging to the same class while neighbors belonging to different classes are separable in the subspace. In DAG-DNE, it chooses neighbors to construct inter-class and intra-class graphs in the same way, which finds the nearest neighbors. Thus, it can build a margin between different neighbors if and only if they belong to the different classes when building inter-class graph, and preserve the structure of nearest neighbors if they belong to the same class when constructing intra-class graph. However, during the classification task, we should focus on samples that are separable in the high-dimensional space, and the representations of these two points in the subspace are close to each other if they belong to the same class rather than preserving the local structure.

To achieve this goal, this paper presents a new supervised subspace learning method, called Appropriate points choosing DAG-DNE (Apps-DAG-DNE). In Apps-DAG-DNE, it chooses points to construct inter-class and intra-class graphs in different ways, that is, it chooses the nearest neighbors to construct inter-class graph, which can build a margin between different neighbors if and only if they belong to the different classes, and chooses the farthest points to construct intra-class graph, which can establish relationship between two farthest points if they belong to the same class. Thus, Apps-DAG-DNE could establish relationship among two remote samples and find a good projection matrix for them.

The reminder of this paper is organized as follows. Section 2 introduces the related works on DNE, LDNE and DAG-DNE. In Section 3, we propose the new method Apps-DAG-DNE. Section 4 reports simulation experimental results and Section 5 concludes this paper.

2 Related Works

In this section, we briefly review DNE [8] and its variant LDNE [9] and DAG-DNE [10]. Given a set of samples $\{(\mathbf{x}_i, y_i)\}_i^N$, where $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \{1, 2, \cdots, c\}$ is the class label of $\mathbf{x_i}$, d is the dimensionality of samples, N is the number of training samples, and c is the number of classes. We try to learn a linear transformation mapping which can project the original data from the high d-dimensional space into a low r-dimensional subspace ($r \ll d$) in which the samples are denoted as $\{(\mathbf{v}_i, y_i)\}_i^N$. Specifically, the linear transformation can be defined as

$$\mathbf{v}_i = \mathbf{P}^T \mathbf{x}_i \tag{1}$$

where $\mathbf{P} \in \mathbb{R}^{d \times r}$ is the projection matrix.

2.1 Discriminant neighborhood embedding

DNE is a supervised subspace learning method, which aims to project the samples from high-dimensional space into a low-dimensional space, and makes

the gaps between different samples as wide as possible if they belong to the different classes and as close as possible if they belong to the same class. For a sample $\mathbf{x}_i$, $N_K^w(\mathbf{x}_i)$ and $N_K^b(\mathbf{x}_i)$ denote its K homogeneous and heterogeneous neighbors, respectively. The DNE method has the following steps:

1) Construct the adjacency graph by using K-nearest neighbors. The adjacency weight matrix $\mathbf{F}$ is defined as:

$$F_{ij} = \begin{cases} +1, & \mathbf{x}_i \in S_K^w(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in S_K^w(\mathbf{x}_i) \\ -1, & \mathbf{x}_i \in S_K^b(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in S_K^b(\mathbf{x}_i) \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

2) Feature mapping: Optimize the following objective function:

$$\begin{cases} \min \ \sum\limits_{ij} ||\mathbf{v}_i - \mathbf{v}_j||^2 \mathbf{F}_{ij} \\ \text{s.t.} \quad \mathbf{P}^T\mathbf{P} = \mathbf{I}. \end{cases} \tag{3}$$

with simple algebra, the minimization problem can be rewritten as

$$\begin{cases} \min\limits_{\mathbf{P}} \ \text{tr}\{\mathbf{P}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{P}\} \\ \text{s.t.} \quad \mathbf{P}^T\mathbf{P}{=}\mathbf{I} \end{cases} \tag{4}$$

Where $\mathbf{L} = \mathbf{D} - \mathbf{W}$ and $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$. The projection matrix $\mathbf{P}$ can be optimized by computing the minimum eigenvalue solution to the generalized eigenvalue problem as follows:

$$\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{P} = \lambda\mathbf{P} \tag{5}$$

where $\mathbf{P}$ is composed of the optimal r projection vectors corresponding to the r smallest eigenvalues.

2.2 Locality-based discriminant neighborhood embedding

Different from DNE, LDNE uses a heat kernel function instead of 1 or 0 to adopt the adjacency graph and aims to find an optimal projection matrix by maximizing the difference between the inter-class scatter and the intra-class scatter. Similar to DNE, the LDNE method has the following steps:

1) Construct the adjacency graph by using K-nearest neighbors. The adjacency weight matrix $\mathbf{S}$ is defined as:

$$S_{ij} = \begin{cases} -\exp(\frac{-||\mathbf{x}_i - \mathbf{x}_j||^2}{\beta}), & \mathbf{x}_i \in S_K^w(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in S_K^w(\mathbf{x}_i) \\ +\exp(\frac{-||\mathbf{x}_i - \mathbf{x}_j||^2}{\beta}), & \mathbf{x}_i \in S_K^b(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in S_K^b(\mathbf{x}_i) \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

2) Feature mapping: Optimize the following objective function:

$$\begin{cases} \max\limits_{\mathbf{P}} \ \text{tr}\{\mathbf{P}^T\mathbf{X}\mathbf{H}\mathbf{X}^T\mathbf{P}\} \\ \text{s.t.} \quad \mathbf{P}^T\mathbf{P}{=}\mathbf{I} \end{cases} \tag{7}$$

Where $\mathbf{H} = \mathbf{D} - \mathbf{S}$, and $\mathbf{D}$ is a diagonal matrix with $\mathbf{D}_{ii} = \sum_j \mathbf{S}_{ij}$. The optimization problem (7) can also be cast into a generalized eigen-decomposition problem:

$$\mathbf{X}\mathbf{H}\mathbf{X}^T\mathbf{P} = \lambda\mathbf{P} \tag{8}$$

The optimal projection $\mathbf{P}$ consists of r eigenvectors corresponding to the r largest eigenvalues.

2.3 Double adjacency graphs-based discriminant neighborhood embedding

DAG-DNE is a useful linear manifold learning method for dimensionality reduction and preserves the local geometric reconstruction relationship of data, which tries to make the gaps among submanifolds for different classes as wide as possible and for same class as compact as possible in the subspace, simultaneously. First, DAG-DNE is to construct two adjacency graphs, let $\mathbf{F}^b$ and $\mathbf{F}^w$ be the inter-class and intra-class adjacency matrices, respectively.

The inter-class adjacency matrix $\mathbf{F}^b$ is defined as

$$F_{ij}^b = \begin{cases} 1, & \mathbf{x}_i \in S_K^b(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in S_K^b(\mathbf{x}_i) \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

The intra-class adjacency matrix $\mathbf{F}^w$ is defined as

$$F_{ij}^w = \begin{cases} 1, & \mathbf{x}_i \in S_K^w(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in S_K^w(\mathbf{x}_i) \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

DAG-DNE seeks to find a projection $\mathbf{P}$ by solving the following objective function

$$\begin{cases} \max\limits_{\mathbf{P}} \ \mathrm{tr}\{\mathbf{P}^T\mathbf{X}\mathbf{G}\mathbf{X}^T\mathbf{P}\} \\ \text{s.t.} \quad \mathbf{P}^T\mathbf{P}{=}\mathbf{I} \end{cases} \tag{11}$$

Where $\mathbf{G} = \mathbf{D}^b - \mathbf{F}^b - \mathbf{D}^w + \mathbf{F}^w$, and $\mathbf{D}^b$ and $\mathbf{D}^w$ are diagonal matrices with $\mathbf{D}_{ii}^b = \sum_j \mathbf{F}_{ij}^b$ and $\mathbf{D}_{ii}^w = \sum_j \mathbf{F}_{ij}^w$. The projection matrix $\mathbf{P}$ can be found by solving the generalized eigenvalue problem as follows:

$$\mathbf{X}\mathbf{G}\mathbf{X}^T\mathbf{P} = \lambda\mathbf{P} \tag{12}$$

The optimal projection $\mathbf{P}$ consists of r eigenvectors corresponding to the r largest eigenvalues.

3 Proposed Method

In this section, we develop a novel supervised subspace learning method called Appropriate points choosing DAG-DNE (Apps-DAG-DNE). As described above, those methods all choose the nearest points to construct inter-class graph which can formulate the marginal between samples if and only if they belong to the different classes, and choose the nearest points to construct intra-class graph which can preserve the local geometric structure between samples if and only if they belong to the same class. However, during the classification task, we should focus on samples that are separable in the high-dimensional space, but the representations of these two points in the subspace should be close to each other if they belong to the same class rather than preserving the local structure.

3.1 Motivation

To clearly illustrate the problem, Fig. 1 gives two ways of choosing points to construct adjacency graphs. In the figure, there are two ways to find nearest neighbors to construct adjacency graphs: (1) a1, which is the traditional way to choose neighbors, and (2) b1, which is the ideal way to choose neighbors. Then, we also have two results: (1) a2, which is the result of constructing the adjacency graph using the traditional way to find points, and (2) b2, which is the result of constructing the adjacency graph using the ideal way to find points. Our purpose is to make the representations of these two points as close as possible in the new space if they belong to the same class, whether or not they are close in the high-dimensional space. However, traditional way has two shortcomings, on one hand, the traditional way chooses the nearest points and preserves the local geometric structure by establishing the relationships of them. Thus, if the two same class samples are remote, they cannot establish any relationships, and they will not be close in the subspace either. During the classification task, we should give priority to the samples in the same class that are remote, as the classification ability of the method will be severely deteriorated if they are ignored. On the other hand, the inter-class scatter is larger than intra-class, traditionally. Thus, the effect of intra-class scatter is very small in the objective function of DAG-DNE, and this will deteriorate the classification ability of DAG-DNE. The fundamental challenge is determining how to choose appropriate points to establish relationships in order to achieve the same result as the ideal way.

3.2 Basic idea of Apps-DAG-DNE

Let $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ be a set of N samples in the multi-class classification task, where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{1, 2, ..., c\}$. Our task is to find a subspace, which let these samples, belonging to the same class, as close as possible in the subspace

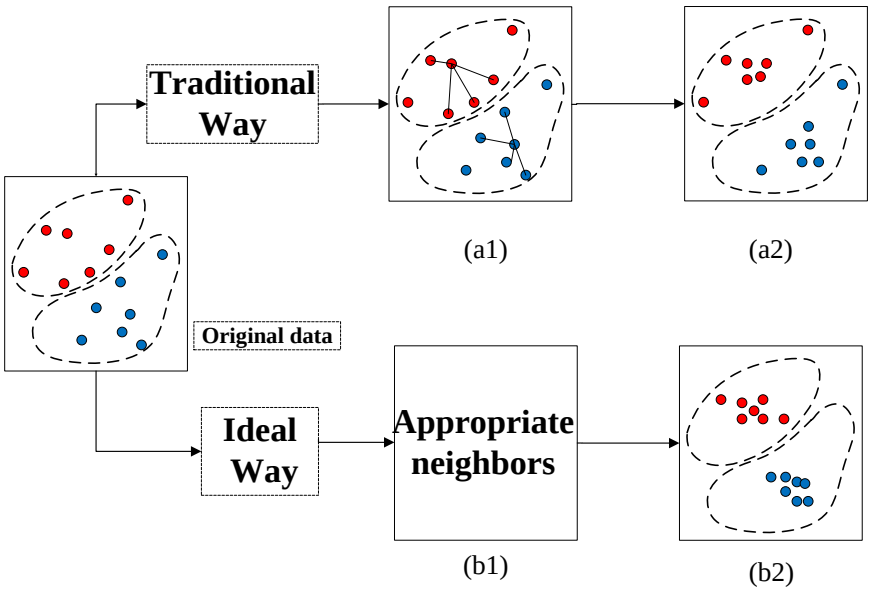


Fig. 1 Procedure of graph-based subspace learning. (a1) traditional way, (a2) the result of choosing the nearest neighbors to construct adjacency graph. (b1) ideal way, (b2) the result of choosing appropriate points to construct adjacency graph. Traditional way makes the samples close if and only if they are close in the high-dimensional space, and the ideal result is that the samples should be close in the subspace if they belong to the same class whether or not they are separable in the high-dimensional space.

even though they are separable in the high-dimensional space. Therefore, we can not only reduce the complexity, but also improve the classification performance.

For a sample $\mathbf{x}_i$, $AN_K^w(i)$ indicates the index set of its K homogeneous points. It is worth noting that, K homogeneous points are its farthest points rather than its nearest neighbors. $AN_K^b(i)$ indicates the index set of its K heterogeneous neighbors. In order to describe the relationships between points, similar to DAG-DNE, we build two adjacency graphs: the inter-class separability graph $\mathbf{F}^b$ and the intra-class compactness graph $\mathbf{F}^w$.

The inter-class separability matrix $\mathbf{F}^b$ is defined as:

$$F_{ij}^b = \begin{cases} 1, & i \in AN_K^b(j) \text{ or } j \in AN_K^b(i) \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

The intra-class compactness matrix $\mathbf{F}^w$ is defined as:

$$F_{ij}^w = \begin{cases} 1, & i \in AN_K^w(j) \text{ or } j \in AN_K^w(i) \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

In the procedure of finding a projection matrix, we utilize the idea that maximizing the inter-class scatter and minimizing the intra-class scatter simul-

taneously. First, we define inter-class scatter and intra-class scatter as follows: the inter-class scatter is

$$\begin{aligned} B(\mathbf{P}) &= \sum_i \sum_{i\in AN_K^b(j) \text{ or } j\in AN_K^b(i)} ||\mathbf{P}^T\mathbf{x}_i - \mathbf{P}^T\mathbf{x}_j|| \\ &= 2\mathbf{P}^T\mathbf{X}(\mathbf{D}^b - \mathbf{F}^b)\mathbf{X}^T\mathbf{P} \end{aligned} \tag{15}$$

where $\mathbf{D}^b$ is a diagonal matrix and its entries are column sum of $\mathbf{F}^b$, i.e. $\mathbf{D}^b_{ii} = \sum_j \mathbf{F}^b_{ij}$.

The intra-class scatter is

$$\begin{aligned} W(\mathbf{P}) &= \sum_i \sum_{i\in AN_K^w(j) \text{ or } j\in AN_K^w(i)} ||\mathbf{P}^T\mathbf{x}_i - \mathbf{P}^T\mathbf{x}_j|| \\ &= 2\mathbf{P}^T\mathbf{X}(\mathbf{D}^w - \mathbf{F}^w)\mathbf{X}^T\mathbf{P} \end{aligned} \tag{16}$$

where $\mathbf{D}^w$ is a diagonal matrix and its entries are column sum of $\mathbf{F}^w$, i.e. $\mathbf{D}^w_{ii} = \sum_j \mathbf{F}^w_{ij}$.

Then, we need to maximize the margin between inter-class scatter and intra-class compactness, i.e.

$$T(\mathbf{P}) = B(\mathbf{P}) - W(\mathbf{P}) \tag{17}$$

Indeed, the margin $T(\mathbf{P})$ measures the total differences among the distances from $\mathbf{x}_i$ to the inter-class neighbors and intra-class points in the projected space.

To gain more insight, we rewrite $T(\mathbf{P})$ in the form of trace by following some simple algebraic steps:

$$\begin{aligned} T(\mathbf{P}) &= B(\mathbf{P}) - W(\mathbf{P}) \\ &= 2\text{tr}\{\mathbf{P}^T\mathbf{X}(\mathbf{D}^b - \mathbf{F}^b)\mathbf{X}^T\mathbf{P} - 2\mathbf{P}^T\mathbf{X}(\mathbf{D}^w - \mathbf{F}^w)\mathbf{X}^T\mathbf{P}\} \\ &= 2\text{tr}\{\mathbf{P}^T\mathbf{X}(\mathbf{D}^b - \mathbf{F}^b - \mathbf{D}^w + \mathbf{F}^w)\mathbf{X}^T\mathbf{P}\} \\ &= 2\text{tr}\{\mathbf{P}^T\mathbf{X}\mathbf{Q}\mathbf{X}^T\mathbf{P}\} \\ &= 2\sum_{i=1}^{d} \mathbf{P}_i^T\mathbf{X}\mathbf{Q}\mathbf{X}^T\mathbf{P}_i \end{aligned} \tag{18}$$

Where $\mathbf{Q} = \mathbf{D}^b - \mathbf{F}^b - \mathbf{D}^w + \mathbf{F}^w$, with equation (18), it can be modified to

$$\begin{cases} \max\limits_{\mathbf{P}} \ \text{tr}\{\mathbf{P}^T\mathbf{X}\mathbf{Q}\mathbf{X}^T\mathbf{P}\} \\ \text{s.t.} \quad \mathbf{P}^T\mathbf{P}=\mathbf{I} \end{cases} \tag{19}$$

Given the constraint $\mathbf{P}^T\mathbf{P} = \mathbf{I}$, the columns of $\mathbf{P}$ are orthogonal. The orthogonal projection matrix is better able to enhance the discriminant ability [10, 22, 23].

In order to solve the formulation (19), we give the following theorems.

Theorem 1. If $\mathbf{D}^b, \mathbf{D}^w, \mathbf{F}^b, \mathbf{F}^w$ are all real symmetric matrices, then $\mathbf{X}\mathbf{Q}\mathbf{X}^T$ is a real symmetric matrix.

Proof. Since $\mathbf{D}^b, \mathbf{D}^w, \mathbf{F}^b, \mathbf{F}^w$ are real symmetric matrices, so $(\mathbf{D}^b)^T = \mathbf{D}^b$, $(\mathbf{D}^w)^T = \mathbf{D}^w$, $(\mathbf{F}^b)^T = \mathbf{F}^b$, $(\mathbf{F}^w)^T = \mathbf{F}^w$, hence,

$$\begin{aligned}(\mathbf{X}\mathbf{Q}\mathbf{X}^T)^T &= \{\mathbf{X}(\mathbf{D}^b - \mathbf{F}^b - \mathbf{D}^w + \mathbf{F}^w)\mathbf{X}^T\}^T \\ &= (\mathbf{X}^T)^T(\mathbf{D}^b - \mathbf{F}^b - \mathbf{D}^w + \mathbf{F}^w)^T\mathbf{X}^T \\ &= \mathbf{X}(\mathbf{D}^b - \mathbf{F}^b - \mathbf{D}^w + \mathbf{F}^w)\mathbf{X}^T \\ &= \mathbf{X}\mathbf{Q}\mathbf{X}^T\end{aligned} \tag{20}$$

Thus, the matrix $\mathbf{X}\mathbf{Q}\mathbf{X}^T$ is a symmetric matrix.

This completes the proof. □

Theorem 2. Since $\mathbf{X}\mathbf{Q}\mathbf{X}^T$ is a real symmetric matrix, the optimization problem (19) is equivalent to the eigen-decomposition problem of matrix $\mathbf{X}\mathbf{Q}\mathbf{X}^T$. Assume that $\lambda_1 \geq \cdots \geq \lambda_i \geq \cdots \lambda_r \geq 0 \geq \cdots \geq \lambda_d$ are the eigenvalues of $\mathbf{X}\mathbf{Q}\mathbf{X}^T$, and $\mathbf{P}$ is the corresponding eigenvector of the eigenvalue λ_r. The optimal projection matrix $\mathbf{P}$ is only composed of eigenvectors corresponding to the top r largest positive eigenvalues, or

$$\mathbf{P} = [\mathbf{P}_1, \cdots, \mathbf{P}_r] \tag{21}$$

Proof. Since $\mathbf{X}\mathbf{Q}\mathbf{X}^T$ is a real symmetric matrix, assume that $\lambda_i (1 \leq i \leq N)$. The eigenvalues of $\mathbf{X}\mathbf{Q}\mathbf{X}^T$ and $\mathbf{P}_i$ are the corresponding eigenvectors of the eigenvalues λ_i. According to the properties of the matrix, we yield

$$\mathbf{X}\mathbf{Q}\mathbf{X}^T\mathbf{P}_i = \lambda_i\mathbf{P}_i \tag{22}$$

Thus,

$$\mathbf{P}_i^T\mathbf{X}\mathbf{Q}\mathbf{X}^T\mathbf{P}_i = \mathbf{P}_i^T\lambda_i\mathbf{P}_i = {\mathbf{P}_i}^T\mathbf{P}_i\lambda_i = \lambda_i \tag{23}$$

Thus, the objective function (19) can be rewritten as

$$\begin{aligned}tr\{\mathbf{P}^T\mathbf{X}\mathbf{Q}\mathbf{X}^T\mathbf{P}\} &= \sum_{i=1}^{d}\mathbf{P}_i^T\mathbf{X}\mathbf{Q}\mathbf{X}^T\mathbf{P}_i \\ &= \sum_{i=1}^{d}\mathbf{P}_i^T\lambda_i\mathbf{P}_i = \sum_{i=1}^{d}\lambda_i\end{aligned} \tag{24}$$

Since matrix $\mathbf{X}\mathbf{Q}\mathbf{X}^T$ is non-positive definite, the eigenvalues of $\mathbf{X}\mathbf{Q}\mathbf{X}^T$ can be positive, negative, or zero. In order to maximize $\mathrm{tr}\{\mathbf{P}\mathbf{X}\mathbf{Q}\mathbf{X}^T\mathbf{P}\}$, we should choose all positive eigenvalues, or $\sum_{i=1}^{r}\lambda_i$. Thus, when $\mathrm{tr}\{\mathbf{X}\mathbf{Q}\mathbf{X}^T\}$ achieves its maximal value $\sum_{i=1}^{r}\lambda_i$, the optimal solution to (19) must be

$$\mathbf{P} = [\mathbf{P}_1, \mathbf{P}_2, \cdots, \mathbf{P}_r] \tag{25}$$

This completes the proof. □

From Theorem 1, we know that $\mathbf{X}\mathbf{Q}\mathbf{X}^T$ is a symmetric matrix, and switches to a simple eigenvalue and eigenvector problem with respect to symmetric matrix $\mathbf{X}\mathbf{Q}\mathbf{X}^T$. From Theorem 2, we know that the optimal project matrix is only composed of eigenvectors corresponding to the top r largest positive eigenvalues. Hence, the transformation matrix $\mathbf{P}$ is constituted by the r eigenvectors

of $\mathbf{XQX}^T$ corresponding to its first r positive eigenvalues. So the embedding of new test sample $\mathbf{x}_{new} \in \mathbb{R}^d$ is accomplished by $\mathbf{y}_{new} = \mathbf{P}^T\mathbf{x}_{new}$, where $\mathbf{y}_{new} \in \mathbb{R}^r (r \ll d)$.

The details of the Apps-DAG-DNE are given in Algorithm 1.

Algorithm 1 : Apps-DAG-DNE

Input: A training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, and the dimensionality of subspace r;
Output: Projection matrix $\mathbf{P}$.
Step 1: Compute the inter-class matrix $\mathbf{F}^b$ by

$$F_{ij}^b = \begin{cases} 1, & i \in AN_K^b(j) \text{ or } j \in AN_K^b(i) \\ 0, & \text{otherwise} \end{cases}$$

and the intra-class $\mathbf{F}^w$ by

$$F_{ij}^w = \begin{cases} 1, & i \in AN_K^w(j) \text{ or } j \in AN_K^w(i) \\ 0, & \text{otherwise} \end{cases}$$

Step 2: Reformulate the objective function, where $\mathbf{XQX}^T = \mathbf{X}(\mathbf{D}^b - \mathbf{F}^b - \mathbf{D}^w + \mathbf{F}^w)\mathbf{X}^T$;
Step 3: Eigendecompose the matrix $\mathbf{XQX}^T$;
Step 4: Choose the first r (assume it has r positive eigenvalues) largest positive eigenvalues corresponding to eigenvectors $\mathbf{P} = [\mathbf{P}_1, \cdots, \mathbf{P}_r]$.

4 Performance Evaluation

4.1 Experiment setup

To evaluate the effectiveness of the proposed method, we have extensively validated our Apps-DAG-DNE method on two widely used datasets, i.e., Yale and Caltech-Leaves datasets, and the results are compared with NPE, DNE, LDNE and DAG-DNE. All of these methods are adopted to find the low-dimensional representations, which require the nearest neighbor parameter K for constructing adjacency graphs. For simplicity, the nearest neighbor (NN) classifier is used for classifying test images in the projected spaces.

All experiments are performed on the personal computer with a 2.30 GHz Intel(R) Core (TM) i5-6200 CPU and 8 G bytes of memory. This computer runs on windows 7, with Matlab 2013a compiler installed.

4.2 Datasets

In order to study the performance and generality of different methods, we perform experiments on two image datasets:

1) The Caltech-Leaves dataset [24] consists of 186 images of 20 species of Leaves against cluttered different backgrounds. Each image was resized to 32×32 pixels in the experiment. Fig. 2 shows some image samples in Caltech-Leaves dataset.

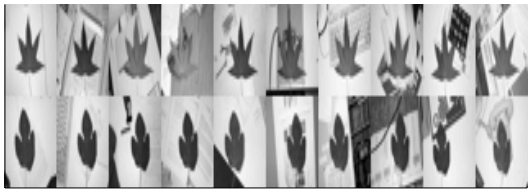


Fig. 2 Samples from the Caltech-Leaves dataset.

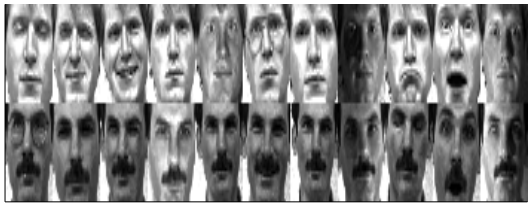


Fig. 3 Samples from the Yale face dataset.

2) The Yale face dataset [25] contains 165 images of 15 people. Each person has 11 images. Each image was manually cropped and resized to 32×32 pixels. Fig. 3 shows some image samples in Yale dataset.

4.3 Comparison methods

Here we further compare the proposed Apps-DAG-DNE method with the following methods.

1) NPE [5]. NPE is an unsupervised method, which finds an embedding to preserve local information and can be extended to unseen samples.

2)DNE [8]. DNE is a supervised method, which keeps local structure by constructing an adjacency graph, and could find optimal projection directions by using spectrum analysis.

3)LDNE [9]. LDNE is also a supervised method, which optimizes the difference between the inter-class distance and the intra-class distance under constructing the adjacency graph by endowing different weights.

4)DAG-DNE [10]. DAG-DNE is also a supervised method, which finds the best projection matrix by constructing double adjacency graphs.

4.4 Performance metric

In the classification of high-dimensional data, many researchers (e.g., Xiaofei He [26], Zhao Zhang [27]) have used the classification accuracy to verify the performance of the methods. The performance of all methods is measured by classification accuracy which is calculated by

$$Accuracy = \frac{CTest}{Test} \tag{26}$$

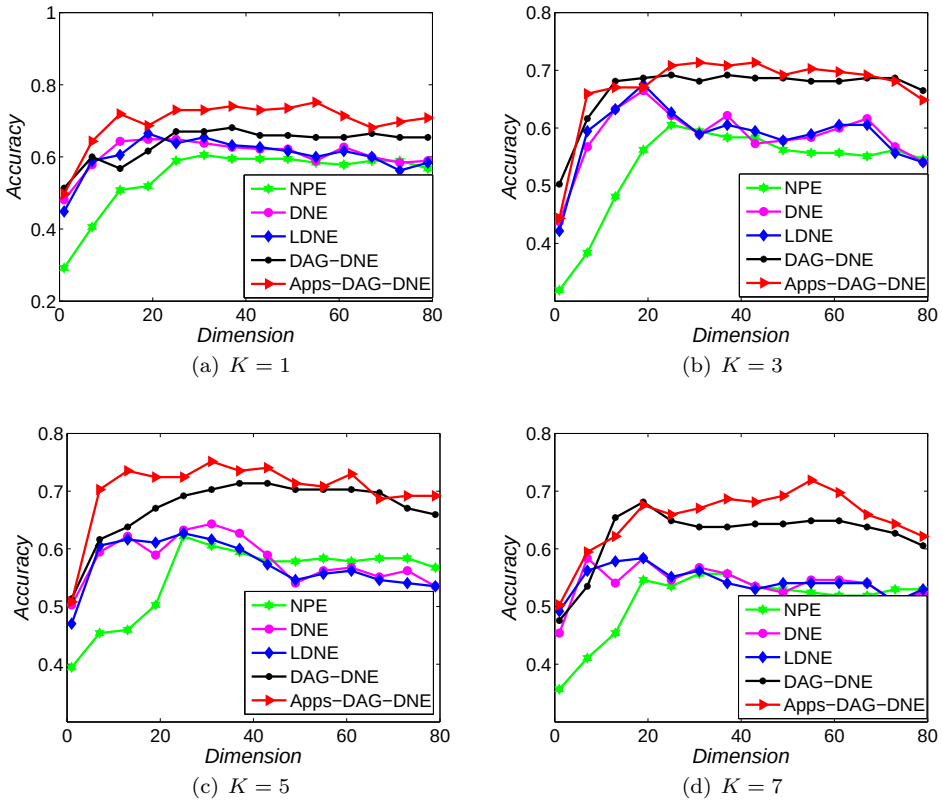
where $CTest$ denotes the number of correctly classified test samples, and $Test$ denotes the number of all test samples.

4.5 Experimental Results

4.5.1 Results of Caltech-Leaves Recognition

In the experiment, we investigated the effect of the number of neighbors K and the ratio between the number of training and testing data to classification performance. First, we set $K = 3$ and evaluated the effect of ratio between the number of training and testing data that 60% and 90% of the training samples were randomly selected, and the remaining samples were used to test. Second, we chose 80% of samples as training samples and evaluated the effect of the number of neighbors. Without prior knowledge, K was set to be 1, 3, 5 and 7. PCA was utilized to reduce dimensionality from 1024 to 100, which could reduce the computational complexity and diminish the majority of noises. we regulated the number of projection vectors from 1 to 80 at an interval of 6, and the results were averaged over the 15 trials. Fig. 4 shows the accuracy of five methods vs. dimensionality of subspace with different K, and Fig. 5 shows the accuracy of five methods vs. dimensionality of subspace with different training samples. From Figs. 4-5, we found that: 1) The performance of each method improved rapidly, and then almost became stable. 2) The DNE, LDNE, DAG-DNE and Apps-DAG-DNE performed better than NPE because NPE is an unsupervised method. More importantly, Apps-DAG-DNE performed better than DNE, LDNE and DAG-DNE across a wide range of dimensionality on the Caltech-Leaves dataset. 3) When the training sample size is large enough to sufficiently characterize the data distribution, such as the case for the 90% training samples on Caltech-Leaves dataset, all methods we discussed in this work can achieve good performance. Based on choosing appropriate points, our Apps-DAG-DNE delivered higher accuracy than other techniques, primarily due to advantages of choosing appropriate points to some extent.

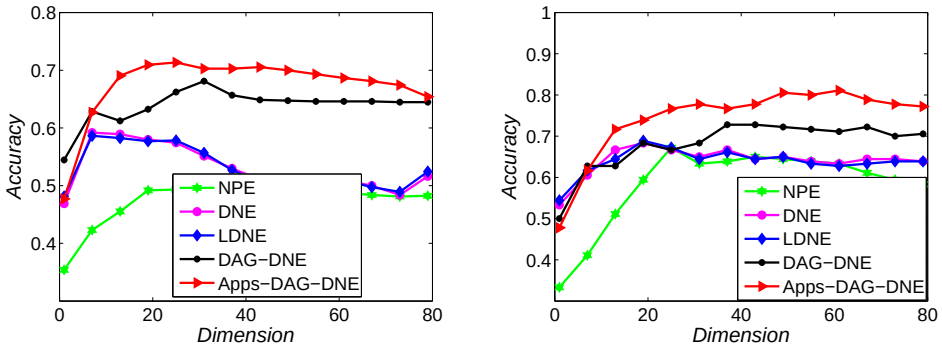
Furthermore, Table 1 reports the best average classification accuracy on test sets of all methods under different K, and in Table 2, we summarize the statistics according to Fig. 5, including the mean accuracy, the best record, and the optimal image subspace dimensions (i.e., Dim), where the optimal subspace corresponded to the highest recognition accuracy for each method in each setting. We made the following similar observations: 1) In spite of the variation of K, Apps-DAG-DNE had the highest classification accuracy among these methods. 2) Apps-DAG-DNE did not have the lowest number of dimensions of the subspace when achieving the best performance, e.g., 90% training samples, NPE: 28, DNE: 24, LDNE: 24, DAG-DNE: 24, Apps-DAG-DNE: 50. It is



(a) $K = 1$ (b) $K = 3$

(c) $K = 5$ (d) $K = 7$

Fig. 4 Accuracy vs. dimension on the Caltech-Leaves dataset under different K.



(a) 60% training samples (b) 90% training samples

Fig. 5 Accuracy vs. dimension on the Caltech-Leaves dataset under different training samples.

Table 1 Performance comparison of five methods on Caltech-Leaves dataset with different numbers of points

Method	$K = 1$	$K = 3$	$K = 5$	$K = 7$
NPE	$63.78\% \pm 2.42$	$63.78\% \pm 3.69$	$65.41\% \pm 2.92$	$58.38\% \pm 2.39$
DNE	$66.49\% \pm 2.81$	$68.11\% \pm 2.58$	$65.41\% \pm 1.62$	$62.16\% \pm 1.29$
LDNE	$66.49\% \pm 1.75$	$67.57\% \pm 3.52$	$64.76\% \pm 2.62$	$62.16\% \pm 2.31$
DAG-DNE	$70.27\% \pm 2.47$	$69.73\% \pm 1.40$	$71.35\% \pm 1.22$	$68.65\% \pm 1.37$
Apps-DAG-DNE	$\mathbf{76.22\% \pm 1.42}$	$\mathbf{74.05\% \pm 1.41}$	$\mathbf{76.76\% \pm 1.37}$	$\mathbf{72.97\% \pm 0.61}$

Table 2 Performance comparison of five methods on Caltech-Leaves dataset with different numbers of training samples

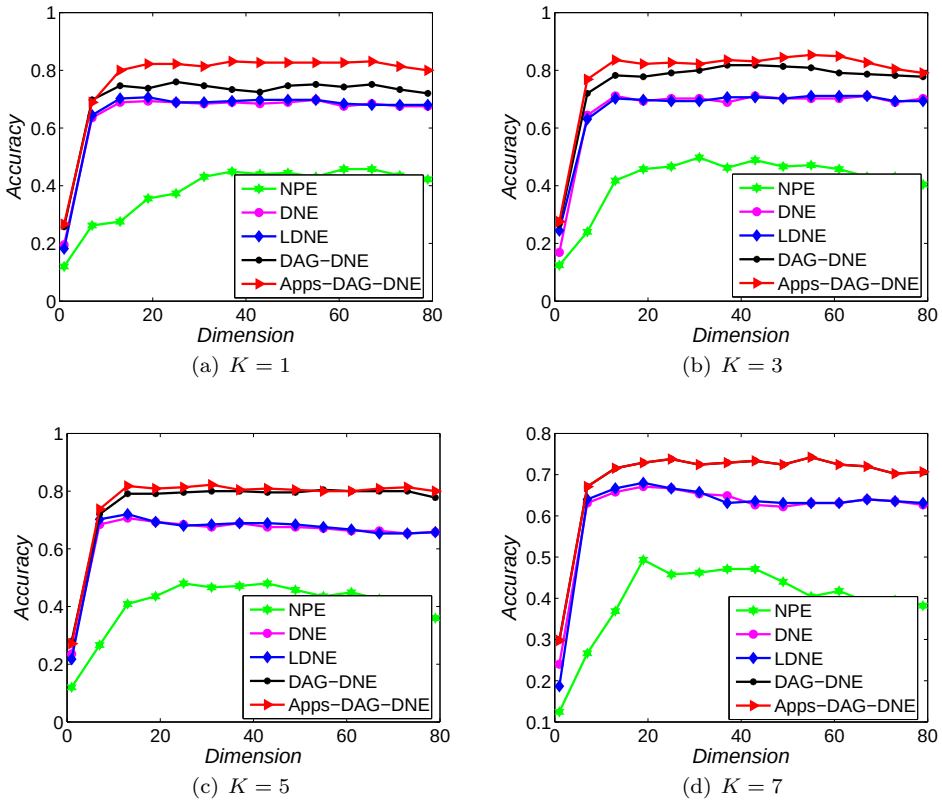
Method/Result	Leaves (60% training samples)			Leaves (90% training samples)		
	Mean	Best	Dim	Mean	Best	Dim
NPE	0.5054	0.6261	23	0.6944	0.8889	24
DNE	0.5946	0.7432	6	0.6944	0.9444	24
LDNE	0.5986	0.7297	6	0.6944	0.9444	24
DAG-DNE	0.6838	0.8243	30	0.7278	0.9444	35
Apps-DAG-DNE	**0.7162**	**0.8378**	**27**	**0.8278**	**1**	**50**

worth noting that when the number of dimensions of the subspace of Apps-DAG-DNE is 24, it yielded better performance than NPE, DNE, LDNE and DAG-DNE.

4.5.2 Results of Yale recognition

In this simulation, we focused on the effect of the dimensionality of subspace under different choices of nearest neighbor parameters K. Similar to the experiment on Caltech-Leaves, we set the numbers of nearest neighbors to be 1, 3, 5 and 7. We randomly selected 70% training samples from each class, where the remaining images were used for testing. For simplicity, PCA was utilized to reduce the number of dimensions from 1,024 to 100. We repeated 15 trials and reported the average results. For each setting, the number of projection vectors were regulated from 1 to 80 at an interval of 6 for each fixed K value. Fig. 6 shows the average accuracy values of the five methods versus the dimensionality of the subspace with different values of K on Yale dataset. From Fig. 6, we found that: 1) Apps-DAG-DNE performed better than NPE, DNE, LDNE, DAG-DNE across a wide range of dimensionality on the Yale dataset. 2) In spite of the variation of K, Apps-DAG-DNE had the highest classification accuracy among these methods. The major reason for this might have been that Apps-DAG-DNE chooses the appropriate points.

Furthermore, Table 3 reports the best average classification accuracy on test sets of all five methods under different K. We can see that Apps-DAG-DNE has the highest accuracy among these methods. By using Apps-DAG-DNE to learn subspace, we can not only reduce the computational complexity but also improve the classification performance.



(a) $K = 1$ (b) $K = 3$

(c) $K = 5$ (d) $K = 7$

Fig. 6 Accuracy vs. dimension on the Yale dataset under different K.

Table 3 Performance comparison of five methods on Yale dataset with different numbers of points

Method	$K = 1$	$K = 3$	$K = 5$	$K = 7$
NPE	$46.22\% \pm 2.49$	$52.00\% \pm 3.45$	$48.89\% \pm 3.34$	$51.56\% \pm 2.87$
DNE	$70.22\% \pm 2.30$	$71.11\% \pm 2.49$	$71.56\% \pm 3.13$	$67.56\% \pm 3.19$
LDNE	$70.67\% \pm 2.33$	$72.44\% \pm 1.45$	$73.33\% \pm 2.15$	$68.44\% \pm 1.24$
DAG-DNE	$76.00\% \pm 1.42$	$81.73\% \pm 2.17$	$81.33\% \pm 2.24$	$74.22\% \pm 2.18$
Apps-DAG-DNE	$\mathbf{83.56\% \pm 1.27}$	$\mathbf{85.78\% \pm 1.26}$	$\mathbf{82.22\% \pm 1.39}$	$\mathbf{74.22\% \pm 2.18}$

5 Conclusion

In this paper, we proposed an appropriate points choosing method based on double adjacency graphs-based discriminant neighborhood embedding, called Apps-DAG-DNE, which chooses different points with traditional way to construct adjacency graphs. It chooses nearest neighbors to construct inter-class adjacency graph, which can build a margin between samples if they belong to the different classes, and chooses farthest points to construct intra-class

graph, which can establish relationships between remote samples if and only if they belong to the same class. Therefore, the low-dimensional representations produced by the proposed method are close, even if they are separable in the original high-dimensional space. The experimental results show that Apps-DAG-DNE can be very effective for data classification. As for future research, we plan to introduce the knowledge of deep learning for discovering more discriminative subspace.

Acknowledgements This work was supported by the National Science Foundation of China (61571066 and 61472047).

References

1. M. A. Turk, A. P. Pentland (1991) Face recognition using eigenfaces. In: Proceedings of the Conference on Computer Vision and Pattern Recognition, IEEE, pp. 586-591
2. X. F. He, S. C. Yan, Y. X. Hu, P. Niyogi (2001) Face recognition Using laplacian faces. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(3):328-340
3. X. F. He, P. Niyogi (2003) Locality preserving projections. Advances in Neural Information Processing Systems 16:153-160
4. S. T. Roweis, L. K. Saul (2000) Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500):2323-2326
5. X. F. He, D. Cai, S. C. Yan, H. J. Zhang (2005) Neighborhood preserving embedding. In: Proceedings of the Tenth International Conference on Computer Vision, IEEE, 2:1208-1213
6. A. M. Martinez, A. C. Kak (2001) PCA versus LDA. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(2):228-233
7. H. Yu, J. yang (2001) A direct LDA algorithm for high-dimensional data with application to face recognition. Pattern Recognition 34(10):2067-2070
8. W. Zhang, X. Y. Xue, Y. F. Guo (2006) Discriminant neighborhood embedding for classification. Pattern Recognition 39(11):2240-2243
9. J. P. Gou, Z. Y (2012) Locality-based discriminant neighborhood embedding. Computer Joural 56(9):1063-1082
10. C. T. Ding, L. Zhang (2015) Double adjacency graphs-based discriminant neighborhood embedding. Pattern Recognition 48(5):1734-1742
11. S. Miao, J. Wang, Q. Cao, F. Chen, Y. Wang (2016) Discriminant structure embedding for image recognition. Neurocomputing 174:850-857
12. S. C. Yan, D. Xu, B. Y. Zhang, H. J. Zhang (2005) Graph Embedding: a general framework for dimensionality reduction. In: Proceedings of the Conference on Computer Vision and pattern Recognition, IEEE, 2:830-837
13. S. C. Yan, D. Xu, B. Y. Zhang, H. J. Zhang, Q. Yang, S. Lin (2007) Graph Embedding and Extensions: a general framework for dimensionality reduction. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(1):20-51
14. Y. Q. Lu, C. Lu, M. Qi, S. Y. Wang (2010) A supervised locality preserving projections based local matching algorithm for face recognition. Advances in Computer Science and Information Technology, pp. 28-37
15. N. Srivastava, S. Rao (2016) Learning-based text classifiers using the Mahalanobis distance for correlated datasets. International Journal of Big Data Intelligence 3(1):18-27
16. W. W. Lin, X. W. Pang, B. S. Wan, H. F. Li(2016) MR-LDA: An Efficient Topic Model for Classification of Short Text in Big Social Data. International Journal of Grid and High Performance Computing 8(4):100-113
17. M. Chen, Y. H. Li, Z. F. Zhang, C. H. Hsu, S. G. Wang (2016) Real-time and large scale duplicate image detection method based on multi-feature fusion. Journal of Real-Time Image Processing, pp. 1-14

18. W. W. Yu, X. L. Teng, C. Q. Liu (2006) Face recognition using discriminant locality preserving projections. Image and Vision Computing 24(3):239-248
19. L. P. Yang, W. G. Gong, X. H. Gu, W. H. Li, Y. X. Liang (2008) Null space discriminant locality preserving projections for face recognition. Neurocomputing 71(16):3644-3649
20. X. Bao, L. Zhang, B. J. Wang, J. W. Yang (2014) A supervised neighborhood preserving embedding for face recognition. In: Proceedings of the International Joint Conference on Neural Networks, IEEE, pp. 278-284
21. G. B. You, N. N. Zheng, S. Y. Du, Y. Wu (2007) Neighborhood discriminant projection for face recognition. Pattern Recognition Letters 28(10):1156-1163
22. E. Kokiopoulou, Y. Sadd (2007) Orthogonal neighborhood preserving projections: A projection-based dimensionality reduction technique. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(12):2143-2156
23. J. H. Wang, Y. Xu, D. Zhang, J. You (2010) An efficient method for computing orthogonal discriminant vectors. Neurcomputing 73(10):2168-2176
24. Caltech-Leaves dataset [Online]. http://www.vision.caltech.edu/html-files/archive.html. Accessed 30 December 2016
25. Yale dataset [Online]. http://vision.ucsd.edu/content/yale-face-database. Accessed 30 December 2016
26. D. Cai, X. F. He, J. W. Han (2007) Spectral Regression: A unified approach for sparse subspace learning. In: Proceedings of the Seventh International Conference on Data Mining, IEEE, pp. 73-82
27. Z. Zhang, F. Z. Li, M. B. Zhao, L. Zhang, S. C. Yan (2016) Joint low-rank and sparse principal feature coding for enhanced robust representation and visual classification. IEEE Transactions on Image Processing 25(6):2429-2443
28. I. Jolliffe (2002) Principal Component Analysis, 2nd, Springer-Verlag