Neurocomputing 73 (2010) 951–958

Contents lists available at ScienceDirect

Neurocomputing

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

Constrained Laplacian Eigenmap for dimensionality reduction

Chun Chen, Lijun Zhang, Jiajun Bu, Can Wang *, Wei Chen

Zhejiang Key Laboratory of Service Robot, College of Computer Science, Zhejiang University, Hangzhou 310027, China

ARTICLE INFO

Article history:
Received 22 March 2009
Received in revised form
29 August 2009
Accepted 30 August 2009
Communicated by D. Tao
Available online 17 November 2009

Keywords:
Dimensionality reduction
Graph embedding
Laplacian Eigenmap
Document clustering

ABSTRACT

Dimensionality reduction is a commonly used tool in machine learning, especially when dealing with high dimensional data. We consider semi-supervised graph based dimensionality reduction in this paper, and a novel dimensionality reduction algorithm called constrained Laplacian Eigenmap (CLE) is proposed. Suppose the data set contains r classes, and for each class we have some labeled points. CLE maps each data point into r different lines, and each map i tries to separate points belonging to class i from others by using label information. CLE constrains the solution space of Laplacian Eigenmap only to contain embedding results that are consistent with the labels. Then, each point is represented as a r-dimensional vector. Labeled points belonging to the same class are merged together, labeled points belonging to different classes are separated, and similar points are close to one another. We perform semi-supervised document clustering using CLE on two standard corpora. Experimental results show that CLE is very effective.

1. Introduction

In many real life applications, one is often confronted with high-dimensional data (e.g. documents, images). The infamous *curse of dimensionality* states that the performance of most machine learning algorithms degrades rapidly both in effectivity and efficiency as the dimensionality increases [15]. Dimensionality reduction allows one to represent the data in a lower dimensional space, and more importantly, reveal the intrinsic structure of the data [8,45,44,34]. The most popular techniques include principal component analysis (PCA), non-negative matrix factorization (NMF), and graph based dimensionality reduction.

Principal component analysis (PCA) [5,31,30] reduces the dimensionality of the data by finding a few orthogonal linear projections such that the variance of the projected data is maximized. In fact, it turns out that these projections are just the leading eigenvectors of the data's covariance matrix, which are called principal components. PCA is globally optimal in the sense that those projections best preserve the global Euclidean structure of the data. However, it can only discover linear manifold embedded in high dimensional space.

Non-negative matrix factorization (NMF) [22,40,23] is a matrix factorization algorithm that approximates the original non-negative data matrix by the product of two non-negative matrix. The first non-negative matrix can be regarded as containing a set of basis vectors, and the other non-negative matrix contains the new coordinate for each point. The non-negative constraints lead to a parts-based representation because they allow only additive, not subtractive, combinations. NMF is concerned with minimizing the data reconstruction error, whereas the local manifold structure is ignored.

Although the representation of many data is high dimensional, the process generating the data is usually characterized by relatively few degrees of freedom. One natural way to formalize this intuition is to model the data as lying on or near a low dimensional manifold embedded in the high dimensional space [3]. In the last decade, a series of graph based dimensionality algorithms that approximate data manifolds have been proposed, such as Isomap [37], Locally Linear Embedding (LLE) [32], Laplacian Eigenmap (LE) [2], and Locality Preserving Projection (LPP) [18]. A central construction in these algorithms is a neighbor graph which encodes the geometrical information of the data space. It has been shown that, all these algorithms can be interpreted in a general graph embedding framework, and their differences lie in the strategy to design the graph and the embedding type [41]. Besides, recently there have been some interests in tensor based algorithms for dimensionality reduction, see [17,24,35,36,33,39] for details.

In practice, there is usually some prior knowledge available. The most widely used prior knowledge includes class labels of some data points, and pairwise (must-link or cannot-link) constraints. We focus on the former case in this paper. Most previous approaches [21,43,16,26,11,7] use *soft* constraints in their object functions, so they cannot guarantee that data points belonging to the same class are actually mapped together. Bie et al. [4] proposed to use the subspace trick to constrain the solution space. Two-class dimensionality reduction problems can

* Corresponding author.
E-mail addresses: chenc@zju.edu.cn (C. Chen), zljzju@zju.edu.cn (L. Zhang), bjj@zju.edu.cn (J. Bu), wcan@zju.edu.cn (C. Wang), chenw@zju.edu.cn (W. Chen).

0925-2312/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2009.08.021

be handled perfectly. For multi-class problems, the algorithm does map points from the same class into the same point, but cannot guarantee that data points from different classes are mapped separately.

In this paper, we proposed a novel dimensionality reduction algorithm called constrained Laplacian Eigenmap (CLE), which supports multi-class problems. CLE aims to find the projection which respects the intrinsic geometrical structure inferred from all the data points, and also consists with labels. Suppose the data set contains r classes, and for each class we have some labeled points. CLE maps each data point into r different lines, and each map i tries to separate points belonging to class i from others by using label information. The mapping is performed by constraining the solution space of Laplacian Eigenmap to consist with labels. In the resulting r-dimensional space, labeled points belonging to the same class are merged together, labeled points belonging to different classes are separated, and similar points are close to one another.

We perform document clustering using CLE on two standard corpora: TDT2 and Reuters-21578. Experimental results show that this clustering model is very effective, and CLE becomes more competitive as the percentage of labeled documents or the number of clusters increases.

The paper is organized as follows: in Section 2, we give a brief review of related works. Our constrained Laplacian Eigenmap (CLE) algorithm is introduced in Section 3. The experimental results are presented in Section 4. Finally, we provide some concluding remarks and suggestions for future work in Section 5.

2. Related works

2.1. Principal component analysis (PCA)

PCA [5,31,30] can be defined in terms of the orthogonal projections which maximize the variance in the projected space. Given a set of n-dimensional data points $\mathbf{x}_1,\ldots,\mathbf{x}_m$, we first consider the projection onto a one-dimensional space using a n-dimensional vector $\mathbf{u}$. Then, we have the following optimization problem:

$$\mathbf{u}_{opt} = \arg\max_{\mathbf{u}} \mathbf{u}^T S\mathbf{u} \tag{1}$$

with the constraint

$$\mathbf{u}^T\mathbf{u} = 1$$

where S is the data covariance matrix defined by

$$S = \frac{1}{m}\sum_{i=1}^{m}(\mathbf{x}_i\text{-}\overline{\mathbf{x}})(\mathbf{x}_i\text{-}\overline{\mathbf{x}})^T \tag{2}$$

It can be proved that $\mathbf{u}_{opt}$ equals to the eigenvector of S that having the largest eigenvalue λ_1. This eigenvector is called as the first principal component. If we consider the general case of a r-dimensional projection space, the optimal linear projections for which the variance of the projected data is maximized are just the r leading eigenvectors $\mathbf{u}_1,\ldots,\mathbf{u}_r$ of S.

PCA is closely related to latent semantic indexing (LSI) [14] which projects documents onto a lower dimensional space through singular value decomposition (SVD).

2.2. Non-negative matrix factorization (NMF)

NMF [22,40,23] aims to find the non-negative factorization of the original data matrix. Given a set of non-negative n-dimensional data points $\mathbf{x}_1,\ldots,\mathbf{x}_m$, they can be represented as a $n\times m$ data matrix $X=[\mathbf{x}_1,\ldots,\mathbf{x}_m]$. In order to reduce the dimensionality, NMF finds two non-negative matrix W and H such that:

$$X \approx WH \tag{3}$$

where W is a $n\times r$ non-negative matrix, and H is a $r\times m$ non-negative matrix. Usually r is chosen to be smaller than n or m, so that W and H are smaller than the X.

Let $H=[\mathbf{h}_1,\ldots,\mathbf{h}_m]$. NMF can be rewritten column by column as

$$\mathbf{x}_i \approx W\mathbf{h}_i, \quad i=1,\ldots,m \tag{4}$$

So, W can be regarded as containing a set of basis vectors, and each column of H is called an encoding and is in one-to-one correspondence with a data point in X. $\mathbf{h}_i$ can be used as the new coordinate of point $\mathbf{x}_i$. There are many iterative algorithms to calculate W and H [25,23]. Recently, some work has been developed to incorporate manifold structure into NMF [10,42].

2.3. Laplacian Eigenmap (LE)

LE [2] is a typical graph based dimensionality reduction technique. It constructs a nearest neighbor graph to capture local structure in the data. Vertices in the graph correspond to points in the data, and the edges denote neighborhood relationships between them. The non-negatives weights of edges represent the similarity between neighbor points. Given the similarity matrix W, LE compute eigenvalues and eigenvectors for the generalized eigenvector problem:

$$L\mathbf{y} = \lambda D\mathbf{y} \tag{5}$$

where D is the diagonal weight matrix with $D_{ii}=\sum_j W_{ji}$, and $L=D\text{-}W$ is the *graph Laplacian* [12]. Let $\mathbf{y}_1,\ldots,\mathbf{y}_r$ be first r smallest eigenvectors of Eq. (5). The new coordinate of point i is given by the i-th row of $Y=[\mathbf{y}_1,\ldots,\mathbf{y}_r]$. LE tries to map similar points as closely as possible. The objective function of LE is

$$Y_{opt} = \arg\min_{Y}\sum_{i,j}\|Y_i\text{-}Y_j\|^2 W_{ij} = \mathrm{tr}(Y^TLY) \tag{6}$$

with the constraint

$$Y^TDY = I$$

2.4. Semi-supervised (supervised) graph embedding

If we have some data points labeled, the most natural way to make use of label information is to modify the edge weights of the graph [21,43]. If two points belong to the same class, then the edge weight is increased. If two points belong to different class, then the edge weight is decreased. We can incorporate label information into Laplacian Eigenmap as follows. Firstly, we modify the edge weights according to the label information. Then we obtain a new similarity matrix S_{new}, a new diagonal weight matrix D_{new}, and a new graph Laplacian L_{new}. Secondly, we calculate eigenvectors of $L_{new}\mathbf{y}=\lambda D_{new}\mathbf{y}$ to reduce the dimension. We call this algorithm semi-supervised Laplacian Eigenmap (Semi-LE).

Other typical semi-supervised (supervised) graph based algorithms include maximum margin projection (MMP) [16], augmented relation embedding (ARE) [26], and local discriminant embedding (LDE) [11]. The major disadvantage of these approaches is that they cannot guarantee that data points belonging to the same class are actually mapped together. Besides, the subspace trick proposed by Bie et al. can only handle two-class problems.

2.5. Constrained spectral clustering (CSC)

CSC is one closely related topic, since most clustering algorithms perform dimensionality reduction before clustering. We discuss two typical algorithms as follows.

Coleman et al. [13] have proposed one constrained spectral clustering algorithm, which supports must-link and cannot-link advices. Different from pervious methods, the proposed algorithm can handle inconsistent advices. However, their algorithm can only be applied to two-class clustering problems. Ji et al. [20] have proposed a document clustering model that enables the user to provide must-link constraints. Specifically, they introduce a constraint matrix U to encode user's prior knowledge. A penalty term that depends on U^TU is added to the objective function of spectral clustering. It is easy to check that U^TU is just the *graph Laplacian* of one graph constructed by adding an edge between any two must-link points. So, this algorithm is essentially equivalent to the semi-supervised graph embedding algorithm that increases the weight of the edge between any two must-link points.

3. Constrained Laplacian Eigenmap

Definition 1. The semi-supervised dimensionality reduction problem considered in this paper is defined as:

1. *Input*: a set of n-dimensional data points $\mathbf{x}_1,\ldots,\mathbf{x}_m$ that belong to r classes, and for each class i, there are m_i points labeled.
2. *Output*: the projection that respects the intrinsic geometrical structure inferred from all the data points and also consists with labels.

Denote class i as c_i, and the set of labeled points belonging to c_i as l_i.

3.1. Motivation

We can translate one multi-class dimensionality reduction problem into a set of two-class problems. This idea can be justified by considering the indicator matrix. $M\in\{0,1\}^{m\times r}$ is an indicator matrix if $M_{ij}=1$ iff $\mathbf{x}_i\in c_j$. Checking columns of M individually, we can see that the i-th column of M can only separate points belonging to c_i from others, so it can be seen as the embedding result of a two-class dimensionality reduction problem. However, by combining r such vectors to form M, it is sufficient to solve r-class problems, no matter r is 2 or greater.

Based on the above discussion, we propose a novel semi-supervised dimensionality reduction algorithm called constrained Laplacian Eigenmap (CLE). If the data set contains r classes, CLE maps each data point into r different lines, and each map i tries to separate points belonging to c_i from others by using the label information. The mapping is performed based on Laplacian Eigenmap, whose solution space is constrained by *modifying the similarity matrix* and *using the subspace trick*. Putting the result of each map as a column, we can form a matrix $Y\in\mathbf{R}^{m\times r}$, where the i-th row gives the embedding coordinate of the i-th point.

3.2. Constraining the solution space of LE

Suppose we are going to map each point into the k-th line. The data set is treated as containing two classes: c_k and $\overline{c_k}$. Then, this map is required to separate points belonging to c_k from others by using the label information. We use two steps to constrain the solution space of Laplacian Eigenmap. Firstly, we incorporate label information into the graph structure by modifying the similarity matrix. Since we are facing a two-class dimensionality reduction problem, the subspace trick is used later.

3.2.1. Modifying the similarity matrix

Since the data set is treated as containing two classes, we modify the similarity matrix W to make it more consist with this assumption. Assume the maximum similarity is 1. W is modified as follows:

1. $\forall i,j$, if $\mathbf{x}_i,\mathbf{x}_j\in l_k$, set $W_{ij}=1$.
2. $\forall i,j$, if $\mathbf{x}_i\in l_a,\mathbf{x}_j\in l_b, a\neq k$, and $b\neq k$, set $W_{ij}=1$.
3. $\forall i,j$, if $\mathbf{x}_i\in l_k,\mathbf{x}_j\in l_a$, and $a\neq k$, set $W_{ij}=W_{ji}=0$.

After modification, within class (c_k or $\overline{c_k}$) links become tighter, and between class (c_k and $\overline{c_k}$) links become looser. Unlike previous algorithm [21], the modification here is different for different map. Let W_k be the similarity matrix after modification, D_k be the corresponding diagonal matrix, and L_k be the corresponding Laplacian matrix.

3.2.2. Using the subspace trick

Ideally, we want to map all the points belonging to the same class (c_k or $\overline{c_k}$) into a single point. Since we only have a few labeled points for each class, the best we can do is to represent labeled points belonging to c_k by one 1-dimensional vector, represent other labeled points by another vector, and map similar points as closely as possible. Let $\mathbf{y}=[y_1,\ldots,y_m]^T$ be the result of k-th map. We have the following objective function:

$$\mathbf{y}_{opt}=\arg\min_{\mathbf{y}}\sum_{i,j}(y_i\text{-}y_j)^2(W_k)_{ij}=\arg\min_{\mathbf{y}}\mathbf{y}^TL_k\mathbf{y} \tag{7}$$

with constraints

$$\begin{cases} y_i=y_j & \text{if } \mathbf{x}_i,\mathbf{x}_j\in l_k\\ y_i=y_j & \text{if } \mathbf{x}_i\in l_a,\mathbf{x}_j\in l_b, a\neq k, \text{ and } b\neq k\\ y_i\neq y_j & \text{if } \mathbf{x}_i\in l_k,\mathbf{x}_j\in l_a \text{ and } a\neq k\\ \mathbf{y}^TD_k\mathbf{y}=1 \end{cases}$$

Without loss of generality, assume the first m_1 points $d_1,\ldots d_{m_1}$ are labeled points belonging to c_1, the next m_2 points $d_{m_1+1},\ldots,d_{m_1+m_2}$ are labeled points belonging to c_2, and so on. All the rest points are unlabeled. In order to meet the above constraint, we introduce the label constraint matrix $P_k\in\{0,\pm1\}^{m\times(2+m-p)}$, where $p=m_1+\cdots+m_r$. Each row i of P_k corresponds to point $\mathbf{x}_i$, and the matrix is represented as follows:

$$P_k=\begin{pmatrix} \mathbf{1}_{m_1} & \text{-}\mathbf{1}_{m_1} & 0_{m_1\times(m\text{-}p)}\\ \vdots & \vdots & \vdots\\ \mathbf{1}_{m_{k\text{-}1}} & \text{-}\mathbf{1}_{m_{k\text{-}1}} & 0_{m_{k\text{-}1}\times(m\text{-}p)}\\ \mathbf{1}_{m_k} & \mathbf{1}_{m_k} & 0_{m_k\times(m\text{-}p)}\\ \mathbf{1}_{m_{k+1}} & \text{-}\mathbf{1}_{m_{k+1}} & 0_{m_{k+1}\times(m\text{-}p)}\\ \vdots & \vdots & \vdots\\ \mathbf{1}_{m_r} & \text{-}\mathbf{1}_{m_r} & 0_{m_r\times(m\text{-}p)}\\ \mathbf{1}_{(m\text{-}p)} & \mathbf{0}_{(m\text{-}p)} & I_{(m\text{-}p)\times(m\text{-}p)} \end{pmatrix}$$

Using the label constraint matrix P_k, we can map labeled points into two different points by introducing an auxiliary vector $\mathbf{z}$ and equating:

$$\mathbf{y}=P_k\mathbf{z}$$

Substituting it into Eq. (7), we obtain the following optimization problem:

$$\mathbf{z}_{opt}=\arg\min_{\mathbf{z}}\mathbf{z}^TP_k^TL_kP_k\mathbf{z} \tag{8}$$

with the constraint

$$\mathbf{z}^T P_k^T D_k P_k \mathbf{z} = 1$$

Note that three constraints of Eq. (7) are dropped. They are automatically guaranteed by equation $\mathbf{y} = P_k \mathbf{z}$, which we will explain later. By the Rayleigh–Ritz theorem [28], we know the solution of this problem is the smallest eigenvector of the following generalized eigenvector problem:

$$P_k^T L_k P_k \mathbf{z} = \lambda P_k^T D_k P_k \mathbf{z} \tag{9}$$

Let $\mathbf{z}_0$ be the smallest eigenvalue of Eq. (9). We have the following proposition:

Proposition 1. *The smallest eigenvalue* $\mathbf{z}_0$ *is* 0 *with eigenvector* $[1,0,\ldots,0]^T$, *and the corresponding* $\mathbf{y}$ *is the constant vector* $\mathbf{1}$.

Proof. It is easy to check that L_k is positive semi-definite and D_k is positive definite. Since the column vectors of P_k are independent, for any non-zero $\mathbf{z}$, $P_k\mathbf{z}$ is not a zero vector. Therefore, $P_k^T L_k P_k$ is still positive semi-definite and $P_k^T D_k P_k$ is positive definite. This implies that $\lambda \geq 0$. Actually, the smallest eigenvalue is 0 with eigenvector $[1,0,\ldots,0]^T$, since

$$P_k^T L_k P_k [1,0,\ldots,0]^T = P_k^T L_k \mathbf{1} = P_k^T \mathbf{0} = 0$$

The corresponding $\mathbf{y} = P_k[1,0,\ldots,0]^T = \mathbf{1}$. □

Vector $\mathbf{1}$ is useless, since all the data points have the same representations. So we choose the second smallest eigenvector $\mathbf{z}_1$. Let $\mathbf{z}_1 = [z_1^1, z_2^1, \ldots, z_{2+m-p}^1]^T$, it is easy to check that

$$\mathbf{y} = \Big[\underbrace{(z_1^1 - z_2^1), \ldots, (z_1^1 - z_2^1)}_{n_1+\cdots+n_{k-1}}, \underbrace{(z_1^1 + z_2^1), \ldots, (z_1^1 + z_2^1)}_{n_k},$$
$$\underbrace{(z_1^1 - z_2^1), \ldots, (z_1^1 - z_2^1)}_{n_{k+1}+\cdots+n_r}, (z_1^1 + z_3^1), \ldots, (z_1^1 + z_{2+m-p}^1)\Big]^T$$

So labeled points belongs to c_k are represented by $z_1^1 + z_2^1$, and other labeled points are represented by $z_1^1 - z_2^1$. Since points belonging to same class are usually similar to one another, then in theory unlabeled points belonging to c_k will be mapped around $z_1^1 + z_2^1$, and other unlabeled points will be mapped around $z_1^1 - z_2^1$.

3.3. The algorithm

Given a set of data points $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathbf{R^n}$ belonging to r classes and some labeled points for each class, CLE projects each points onto a r-dimensional discriminative space. It is performed as follows:

1. *Constructing the adjacency graph*. Let G denote a graph with m nodes, and the i-th node corresponds to the point $\mathbf{x}_i$. There are two variations [2]:
   (1) ε- neighborhoods [parameter $\varepsilon \in \mathbf{R}$]: Nodes i and j are connected by an edge if $\|\mathbf{x}_i - \mathbf{x}_j\|^2 < \varepsilon$ where the norm is the usual Euclidean norm in $\mathbf{R^n}$.
   (2) k nearest neighbors [parameter $k \in \mathbf{N}$]: Nodes i and j are connected by an edge if $\mathbf{x}_i$ is among k nearest neighbors of $\mathbf{x}_j$ or $\mathbf{x}_j$ is among k nearest neighbors of $\mathbf{x}_i$.
   - *Choosing the weights*. There are many kinds of weighting methods [6]:
     (1) 0-1 weighting,
     (2) Gaussian kernel weighting,
     (3) dot-product weighting, and
     (4) polynomial kernel weighting.
2. *Mapping each data point into r different lines*. Each map i tries to separate points belonging to c_i from others by using the label information.
   Suppose we are going to map each point into the k-th line. We use the following two steps to constrain the solution space of Laplacian Eigenmap (Section 3.2).
   (1) Modifying the similarity matrix:
   If both of two labeled points belong to c_k, the edge weight is increased. If neither of two labeled points belong to c_k, the edge weight is also increased. If one labeled point belongs to c_k, and the other labeled point does not, the edge weight is decreased.
   (2) Using the subspace trick:
   Firstly, we construct the label constraint matrix P_k for the k-th map. Secondly, we calculate the second smallest eigenvector $\mathbf{z}_1$ of the generalized eigenvector problem (9). Then, $\mathbf{y} = P_k \mathbf{z}_1$ gives the result of the k-th map.
4. *Combining the result of each mapping*. Denote the results of all the r maps as $\mathbf{y}_1, \ldots, \mathbf{y}_r$. The embedding coordinate of point $\mathbf{x}_i$ is given by the i-th row of $Y = [\mathbf{y}_1, \ldots, \mathbf{y}_r]$.

After CLE, we get r-dimensional representations of the original data points. It is easy to check that labeled points belonging to the same class are merged together, labeled points belonging to different classes are separated, and similar points are close to one another. So, in this space, better classification or clustering performance can be obtained.

4. Experimental result

In this section, we perform document clustering using CLE to show the effectiveness of our algorithm. Document clustering has received a lot of attention as a fundamental tool for organization, summarization and retrieval of large volumes of text documents. Two standard document collections were used in the experiments: TDT2 and Reuters-21578.

4.1. Data corpora

We used the TDT2 and Reuters-21578 corpora as [6]. The TDT2 corpus[1] consists of documents collected from six sources (APW, NYT, VOA, PRI, CNN, and ABC) during 1998. It consists of 11,201 documents and 96 categories. In our experiments, we removed those documents belonging to two or more categories and used the largest 30 categories. This led to a data set with 9394 documents in 30 categories as described in Table 1. In this table, CluID means cluster's ID and DocNum means number of documents contained in the cluster.

Reuters-21578 corpus[2] contains 21,578 documents in 135 categories. In our experiments, we excluded those documents with multiple labels, and chose the largest 30 categories. It left us with 8067 documents in 30 categories as described in Table 2.

We removed the stop words, and represented each document as a term-frequency vector. Each document vector is normalized to unit length.

4.2. Evaluation metric

The clustering performance is evaluated by comparing the label obtained from our clustering algorithm with that provided by the document corpus. Two metrics, the accuracy (AC) and the normalized mutual information metric ($\overline{MI}$), are used to measure the clustering performance [40,6]. Given a document d_i, let p_i and

[1] Nist Topic Detection and Tracking corpus is at http://www.nist.gov/speech/tests/tdt/1998/

[2] Reuters-21578 corpus is at http://www.daviddlewis.com/resources/testcollections/reuters21578/

Table 1
TDT2 used in our experiments.

CluID	DocNum	CluID	DocNum	CluID	DocNum
c_1	1844	c_{11}	160	c_{21}	76
c_2	1828	c_{12}	145	c_{22}	74
c_3	1222	c_{13}	141	c_{23}	72
c_4	811	c_{14}	140	c_{24}	71
c_5	441	c_{15}	131	c_{25}	66
c_6	407	c_{16}	123	c_{26}	65
c_7	272	c_{17}	123	c_{27}	63
c_8	238	c_{18}	120	c_{28}	58
c_9	226	c_{19}	104	c_{29}	56
c_{10}	167	c_{20}	98	c_{30}	52

Table 2
Reuters-21578 used in our experiments.

CluID	DocNum	CluID	DocNum	CluID	DocNum
c_1	3713	c_{11}	87	c_{21}	37
c_2	2055	c_{12}	63	c_{22}	36
c_3	321	c_{13}	60	c_{23}	33
c_4	298	c_{14}	53	c_{24}	30
c_5	245	c_{15}	45	c_{25}	27
c_6	197	c_{16}	45	c_{26}	24
c_7	142	c_{17}	44	c_{27}	23
c_8	114	c_{18}	42	c_{28}	20
c_9	110	c_{19}	38	c_{29}	19
c_{10}	90	c_{20}	38	c_{30}	18

q_i be the obtained cluster label and the label provided by the corpus, respectively. The *AC* is defined as follows:

$$AC = \frac{\sum_{i=1}^{m} \delta(q_i, \mathrm{map}(p_i))}{m} \tag{10}$$

where m is the total number of documents, $\delta(x,y)$ is the delta function that equals one if $x=y$ and equals zero otherwise, and $\mathrm{map}(p_i)$ is the permutation mapping function that map each cluster label p_i to the equivalent label from the data corpus. The best mapping can be found by using the Kuhn–Munkres algorithm [27].

Let C denote the set of clusters provided by the document corpus and C' obtained from our algorithm. Their mutual information metric $MI(C,C')$ is defined as following:

$$MI(C,C') = \sum_{c_i \in C, c_j' \in C'} p(c_i, c_j') \cdot \log_2 \frac{p(c_i, c_j')}{p(c_i) \cdot p(c_j')}, \tag{11}$$

where $p(c_i)$ and $p(c_j')$ are the probabilities that a document arbitrarily selected from the corpus belongs to the clusters c_i and c_j', respectively, and $p(c_i, c_j')$ is the joint probability that the arbitrarily selected document belongs to the clusters c_i as well as c_j' at the same time. In our experiments, we use the normalized mutual information $\overline{MI}$ as follows:

$$\overline{MI} = \frac{MI(C,C')}{\max(H(C), H(C'))} \tag{12}$$

where $H(C)$ and $H(C')$ are the entropies of C and C', respectively. It is easy to check that $\overline{MI}$ takes values between 0 and 1.

4.3. Clustering results

To demonstrate how our method improves the performance of document clustering, we compared it with the following five methods:

1. k-means on original term-document matrix (k-means),
2. clustering based on Non-negative Matrix Factorization (NMF-NCW, [40]),
3. k-means after LSI (LSI),
4. k-means after Laplacian Eigenmaps (LE),
5. k-means after semi-Laplacian Eigenmap (Semi-LE).

Since k-means algorithm can only find local minimum, and is sensitive to initial points. When we need to perform k-means, we apply it 10 times with different start points and the best result in terms of the objective function of k-means was recorded.

The weighted non-negative matrix factorization based document clustering algorithm (NMF-NCW, [40]) is a recently proposed algorithm. NMF-NCW weights each documents before performing NMF, which has shown to be very effective in document clustering. We use the projected gradient algorithm for NMF proposed by Lin [25]. For the same reason as k-means, when performing NMF, we apply it 10 times with different initial value and the best result in terms of the objective function of NMF was recorded.

Latent semantic indexing (LSI) [14] is one of the most popular linear document indexing methods. LSI is similar with PCA. The covariance matrix of data in PCA corresponds now to the document-term matrix multiplied by its transpose.

Note that LE, Semi-LE, and CLE need to use the similarity matrix. In the following experiments, we used the 15 nearest neighbors approach to construct the graph, and choose the dot-product weighting method. If nodes i and j are connected, $W_{ij} = W_{ji} = d_i^T d_j$. Otherwise, $W_{ij} = W_{ji} = 0$. Since each document vector d_i is normalized to have unit length, so $0 \le W_{ij} \le 1$.

For the Semi-LE, we change the similarity matrix as follows. If two labeled documents belong to the same cluster, then the edge weight is set to 1. If two labeled documents belong to different cluster, then the edge weight is set to 0.

In spectral clustering, the dimensions of the subspace are set to the number of clusters [29]. So, given a cluster number r, we use LE and Semi-LE to map documents into r-dimensional subspace by choosing the first r smallest eigenvector of Eq. (5). For comparison, LSI embeds the documents into a r-dimensional subspace by using the first r largest left singular vector. CLE also maps documents into r-dimensional discriminative space.

The evaluations were conducted with different number of clusters, ranging from 2 to 6. For each given cluster number r, 20 tests were generated by choosing different clusters randomly. For each test, different percentage documents were labeled for each cluster. For each given percentage e, 10 different cases were generated randomly. The results for k-means, LSI, NMF-NCW, and LE were averaged over these 20 tests. Given a percentage e of labeled documents, the results for Semi-LE and CLE were firstly averaged over 10 cases, then averaged over 20 tests. As we can see from Tables 1 and 2, there are 32 document clusters containing less than 100 documents. So in experiments we set percentage e to range from 3% to 10%.

Table 3 shows the experimental results on the TDT2 corpus. As can be seen, LE outperforms k-means, LSI, and NMF-NCW on every cluster number r. The performance of LSI is even worse than k-means on the original document space. The optimal dimension of LSI with r clusters is much higher than r, which has been discussed in [6]. When the percentage of labeled documents is 3% or more, CLE outperforms both LE and Semi-LE on average. Because LE has achieved very good performance—average *AC* is above 0.99 and average $\overline{MI}$ is above 0.95, label information is not very valuable on this corpus. The improvements of both Semi-LE and CLE compared with LE are not very obvious.

Table 4 shows the experimental results on the Reuters corpus. On this corpus, LE outperforms k-means and LSI on every cluster number r, and outperforms NMF-NCW on average. The performance of LSI is worse than k-means on the original

Table 3
Performance comparison on TDT2 corpus.

l	Accuracy											
	k-Means	LSI	NMF-NCW	LE	3%		5%		7%		9%	
					Semi-LE	CLE	Semi-LE	CLE	Semi-LE	CLE	Semi-LE	CLE
2	0.9328	0.9092	0.9827	0.9987	0.9987	0.9987	0.9988	0.9988	0.9988	0.9989	0.9989	0.9989
3	0.8953	0.8546	0.9379	0.9956	0.9957	0.9964	0.9958	0.9963	0.9962	0.9966	0.9959	0.9961
4	0.8547	0.7580	0.9268	0.9910	0.9951	0.9948	0.9951	0.9945	0.9955	0.9935	0.9959	0.9948
5	0.8392	0.7686	0.9229	0.9907	0.9841	0.9885	0.9874	0.9911	0.9887	0.9913	0.9890	0.9911
6	0.8303	0.7538	0.9407	0.9764	0.9786	0.9782	0.9768	0.9807	0.9804	0.9824	0.9749	0.9827
Ave.	0.8704	0.8088	0.9422	0.9905	0.9904	0.9913	0.9908	0.9923	0.9919	0.9925	0.9909	0.9927
	Mutual information											
2	0.8487	0.7645	0.9451	0.9765	0.9771	0.9777	0.9780	0.9788	0.9785	0.9792	0.9796	0.9809
3	0.7980	0.7388	0.8695	0.9626	0.9638	0.9683	0.9649	0.9684	0.9677	0.9702	0.9660	0.9675
4	0.7809	0.6853	0.8819	0.9542	0.9695	0.9682	0.9704	0.9684	0.9713	0.9673	0.9731	0.9696
5	0.7803	0.7040	0.8777	0.9563	0.9496	0.9552	0.9569	0.9614	0.9584	0.9621	0.9599	0.9628
6	0.8008	0.7137	0.8937	0.9379	0.9402	0.9427	0.9423	0.9493	0.9451	0.9513	0.9403	0.9528
Ave.	0.8018	0.7213	0.8936	0.9575	0.9600	0.9624	0.9625	0.9652	0.9642	0.9660	0.9638	0.9667

Table 4
Performance comparison on Reuters corpus.

l	Accuracy											
	k-Means	LSI	NMF-NCW	LE	3%		5%		7%		9%	
					Semi-LE	CLE	Semi-LE	CLE	Semi-LE	CLE	Semi-LE	CLE
2	0.7486	0.7348	0.8689	0.8644	0.8872	0.9016	0.9007	0.9056	0.9051	0.9141	0.9127	0.9302
3	0.6434	0.6297	0.7247	0.7996	0.7996	0.8024	0.8173	0.8272	0.8293	0.8447	0.8456	0.8595
4	0.6896	0.6609	0.7635	0.7394	0.7492	0.7552	0.7706	0.7648	0.7849	0.7838	0.8013	0.8143
5	0.5328	0.5151	0.6539	0.7203	0.7495	0.7493	0.7697	0.7763	0.7912	0.7982	0.7985	0.8297
6	0.4921	0.4603	0.6029	0.6340	0.6493	0.6568	0.6665	0.7024	0.6941	0.7142	0.7232	0.7477
Ave.	0.6213	0.6002	0.7228	0.7515	0.7669	0.7731	0.7850	0.7952	0.8009	0.8110	0.8163	0.8363
	Mutual information											
2	0.3095	0.2728	0.5242	0.5240	0.5581	0.5865	0.5920	0.6023	0.5962	0.6143	0.6192	0.6613
3	0.3867	0.3528	0.4266	0.5381	0.5397	0.5413	0.5643	0.5748	0.5859	0.6127	0.6087	0.6258
4	0.4691	0.4485	0.5119	0.5102	0.5094	0.5045	0.5229	0.5257	0.5343	0.5482	0.5529	0.5841
5	0.3726	0.3485	0.4225	0.5042	0.5171	0.5116	0.5395	0.5523	0.5573	0.5767	0.5627	0.6254
6	0.3679	0.3365	0.4107	0.4498	0.4636	0.4618	0.4785	0.5087	0.4993	0.5267	0.5152	0.5576
Ave.	0.3811	0.3518	0.4592	0.5053	0.5176	0.5211	0.5395	0.5528	0.5546	0.5757	0.5718	0.6108

document space again. Similarly, when the percentage of labeled documents is 3% or more, CLE outperforms both LE and Semi-LE on average. But this time the improvement is much more obvious. With 9% labeled documents, the average AC of CLE is about 2% point higher than that of Semi-LE and 8.5% point higher than LE; the average $\overline{MI}$ is about 4% point higher than of Semi-LE and 10.5% point higher than of LE.

Fig. 1 shows the accuracy of the six algorithms with different percentages of labeled documents and different cluster number on Reuters corpus. Apparently, the performance of Semi-LE and CLE becomes better as the percentage of labeled documents increases. More important, the improvement of CLE compared with Semi-LE becomes more obvious as the percentage of labeled documents e or the number of clusters r increases.

5. Conclusion and future work

In this paper, we propose a novel semi-supervised dimensionality reduction algorithm called constrained Laplacian Eigenmap (CLE). CLE aims to find the projection which respects the intrinsic geometrical structure inferred from all the data points, and also consists with labels. Document clustering experiments on TDT2 and Reuters-21578 corpora have shown that: when the label information is not scarce, CLE performs much better than the Semi-LE which incorporates label information through modifying the similarity matrix, and other unsupervised methods (k-means, LSI, NMF, LE). The advantage of CLE becomes more obvious as the percentage of labeled documents or the number of clusters increases.

Dimensionality reduction is widely used in face recognition. Eigenface [38], Fisherface [1], and Laplacianface [19,9] are three state-of-the-art face recognition techniques based on principal component analysis (PCA), linear discriminant analysis (LDA) and locality preserving projection (LPP), respectively. Compared with PCA and LDA, CLE can discover nonlinear manifolds. LPP seeks to preserve the intrinsic geometry of the data, but it is unsupervised. Previous studies have shown that face images are possibly reside on a nonlinear submanifold. Thus, CLE is very suitable for semi-supervised face recognition, which will be investigated in our future work.

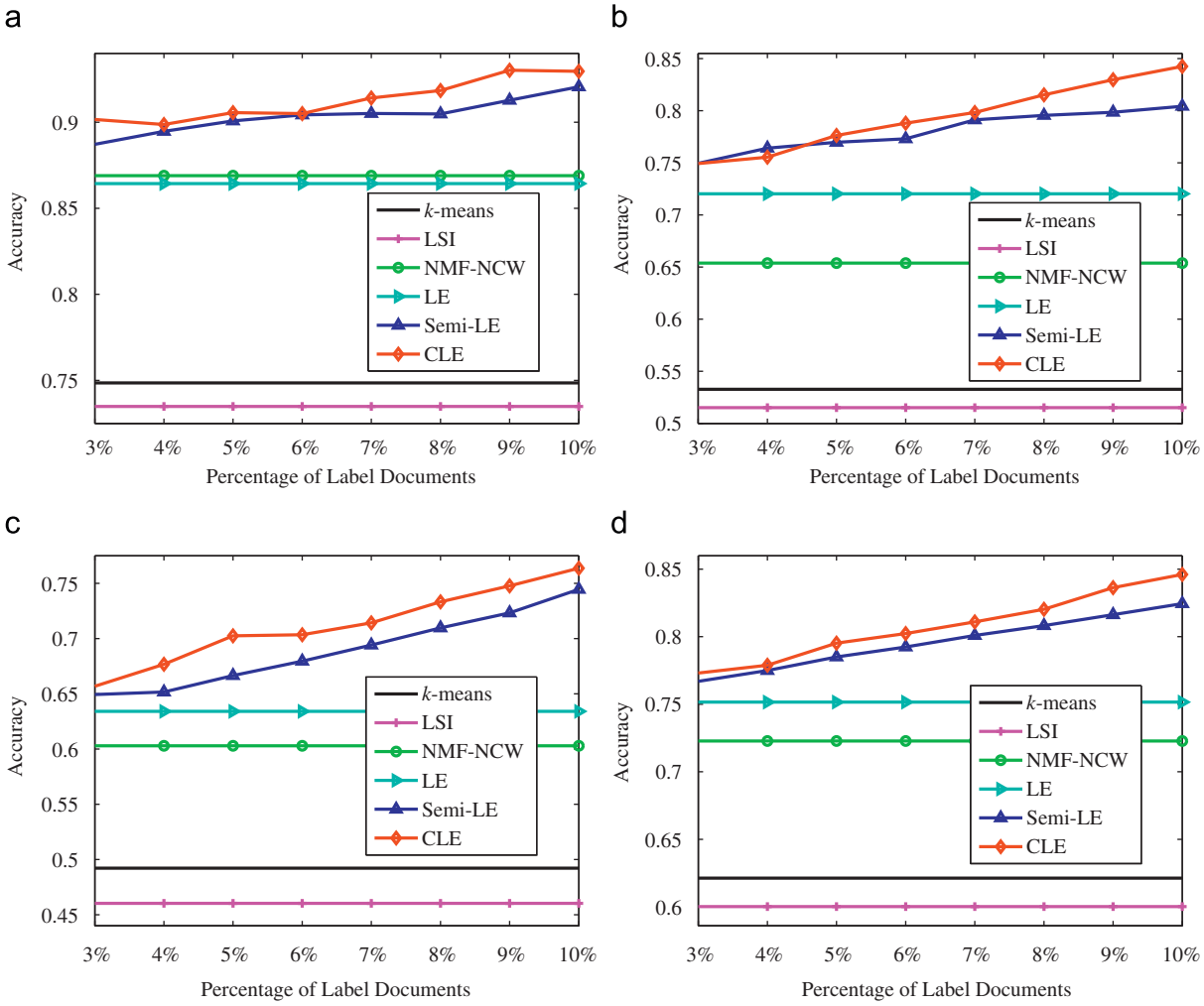


Fig. 1. Accuracy of the six algorithms on Reuters corpus. (a) The accuracy with cluster number 2; (b) the accuracy with cluster number 5; (c) the accuracy with cluster number 6; (d) the average accuracy.

Acknowledgment

This work was supported by National Key Technology R&D Program (2008BAH26B02).

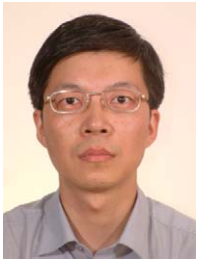
References

[1] P. Belhumeur, J. Hespanha, D. Kriegman, Eigenfaces vs. Fisherfaces: recognition using class specific linear projection, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (1997) 711–720.
[2] M. Belkin, P. Niyogi, Laplacian Eigenmaps and spectral techniques for embedding and clustering, in: Advances in Neural Information Processing Systems, vol. 14, MIT Press, Cambridge, MA, 2002, pp. 585–591.
[3] M. Belkin, P. Niyogi, Towards a theoretical foundation for Laplacian-based manifold methods, in: COLT '05: Proceedings of the 18th Annual Conference on Learning Theory, , 2005, pp. 486–500.
[4] T.D. Bie, J.A.K. Suykens, B.D. Moor, Learning from general label constraints, in: SPR '04: Proceedings of IAPR International Workshop on Statistical Pattern Recognition, 2004, pp. 671–679.
[5] C.M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer, Berlin, 2007.
[6] D. Cai, X. He, J. Han, Document clustering using locality preserving indexing, IEEE Transactions on Knowledge and Data Engineering 17 (12) (2005) 1624–1637.
[7] D. Cai, X. He, J. Han, Semi-supervised discriminant analysis, in: ICCV '07: IEEE 11th International Conference on Computer Vision, , October 2007, pp. 1–7.
[8] D. Cai, X. He, J. Han, Srda: an efficient algorithm for large-scale discriminant analysis, IEEE Transactions on Knowledge and Data Engineering 20 (1) (2008) 1–12.
[9] D. Cai, X. He, J. Han, H.-J. Zhang, Orthogonal Laplacianfaces for face recognition, IEEE Transactions on Image Processing 15 (11) (2006) 3608–3614.
[10] D. Cai, X. He, X. Wu, J. Han, Non-negative matrix factorization on manifold, in: ICDM '08: Proceedings of the 8th IEEE International Conference on Data Mining, IEEE Computer Society, Washington, DC, USA, 2008, pp. 63–72.
[11] H.-T. Chen, H.-W. Chang, T.-L. Liu, Local discriminant embedding and its variants, in: CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Washington, DC, USA, 2005.
[12] F.R.K. Chung, Spectral graph theory, in: Regional Conference Series in Mathematics, vol. 92, American Mathematical Society, 1997.
[13] T. Coleman, J. Saunderson, A. Wirth, Spectral clustering with inconsistent advice, in: ICML '08: Proceedings of the 25th International Conference on Machine Learning, ACM, New York, NY, USA, 2008, pp. 152–159.
[14] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, R. Harshman, Indexing by latent semantic analysis, Journal of the American Society for Information Science 41 (1990) 391–407.
[15] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, Wiley-Interscience, New York, 2000.
[16] X. He, D. Cai, J. Han, Learning a maximum margin subspace for image retrieval, IEEE Transactions on Knowledge and Data Engineering 20 (2) (2008) 189–201.
[17] X. He, D. Cai, P. Niyogi, Tensor subspace analysis, in: Y. Weiss, B. Schölkopf, J. Platt (Eds.), Advances in Neural Information Processing Systems, vol. 18, MIT Press, Cambridge, MA, 2006, pp. 499–506.
[18] X. He, P. Niyogi, Locality preserving projections, in: Advances in Neural Information Processing Systems, vol. 16, MIT Press, Cambridge, MA, 2004, pp. 153–160.
[19] X. He, S. Yan, Y. Hu, P. Niyogi, H.-J. Zhang, Face recognition using Laplacianfaces, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (3) (2005) 328–340.

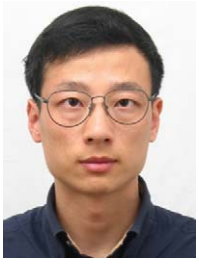
[20] X. Ji, W. Xu, Document clustering with prior knowledge, in: SIGIR '06: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, New York, NY, USA, 2006, pp. 405–412.
[21] S.D. Kamvar, D. Klein, C.D. Manning, Spectral learning, in: IJCAI '03: Proceedings of the 18th International Joint Conference on Artificial Intelligence, , 2003, pp. 561–566.
[22] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (6755) (1999) 788–791.
[23] D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, in: Advances in Neural Information Processing Systems, vol. 13, MIT Press, Cambridge, MA, 2001, pp. 556–562.
[24] X. Li, S. Lin, S. Yan, D. Xu, Discriminant locally linear embedding with high-order tensor data, IEEE Transactions on Systems, Man, and Cybernetics, Part B 38 (2) (2008) 342–352.
[25] C.-J. Lin, Projected gradient methods for nonnegative matrix factorization, Neural Computation 19 (10) (2007) 2756–2779.
[26] Y.-Y. Lin, T.-L. Liu, H.-T. Chen, Semantic manifold learning for image retrieval, in: MULTIMEDIA '05: Proceedings of the 13th Annual ACM International Conference on Multimedia, ACM, New York, NY, USA, 2005, pp. 249–258.
[27] L. Lováz, M.D. Plummer, Matching Theory, North-Holland, Amsterdam, 1986.
[28] H. Lütkepohl, Handbook of Matrices, Wiley, Chichester, 1997.
[29] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm, in: Advances in Neural Information Processing Systems, vol. 14, MIT Press, Cambridge, MA, 2002, pp. 849–856.
[30] Y. Pang, D. Tao, Y. Yuan, X. Li, Binary two-dimensional pca, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 38 (4) (2008) 1176–1180.
[31] Y. Pang, Y. Yuan, X. Li, Iterative subspace analysis based on feature line distance, IEEE Transactions on Image Processing 18 (4) (2009) 903–907.
[32] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.
[33] D. Tao, X. Li, W. Hu, S. Maybank, X. Wu, Supervised tensor learning, in: ICDM '05: Proceedings of the 5th IEEE International Conference on Data Mining, IEEE Computer Society, Washington, DC, USA, 2005, pp. 450–457.
[34] D. Tao, X. Li, X. Wu, S. Maybank, Geometric mean for subspace selection, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2) (2009) 260–274.
[35] D. Tao, X. Li, X. Wu, S.J. Maybank, General tensor discriminant analysis and gabor features for gait recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (10) (2007) 1700–1715.
[36] D. Tao, X. Li, X. Wu, S.J. Maybank, Tensor rank one discriminant analysis—a convergent method for discriminative multilinear subspace selection, Neurocomputing 71 (10–12) (2008) 1866–1882.
[37] J.B. Tenenbaum, V. de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.
[38] M.A. Turk, A.P. Pentland, Face recognition using eigenfaces, in: CVPR '91: Proceedings of 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, , 1991, pp. 586–591.
[39] D. Xu, S. Yan, D. Tao, L. Zhang, X. Li, H.-J. Zhang, Human gait recognition with matrix representation, IEEE Transactions on Circuits and Systems for Video Technology 16 (7) (2006) 896–903.
[40] W. Xu, X. Liu, Y. Gong, Document clustering based on non-negative matrix factorization, in: SIGIR '03: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, New York, NY, USA, 2003, pp. 267–273.
[41] S. Yan, D. Xu, B. Zhang, H. jiang Zhang, Graph embedding: a general framework for dimensionality reduction, in: CVPR 05': Proceedings of the Internal Conference on Computer Vision and Pattern Recognition, 2005, pp. 830–837.
[42] J. Yang, S. Yang, Y. Fu, X. Li, T. Huang, Non-negative graph embedding, in: CVPR '08: Proceedings of the 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Washington, DC, USA, June 2008.
[43] Y. Yuan, Y. Pang, Discriminant adaptive edge weights for graph embedding, in: ICASSP '08: IEEE International Conference on Acoustics, Speech and Signal Processing, 2008, pp. 1993–1996.
[44] T. Zhang, D. Tao, X. Li, J. Yang, Patch alignment for dimensionality reduction, IEEE Transactions on Knowledge and Data Engineering 21 (9) (2009) 1299–1313.
[45] T. Zhang, D. Tao, J. Yang, Discriminative locality alignment, in: ECCV '08: Proceedings of the 10th European Conference on Computer Vision, Springer, Berlin, Heidelberg, 2008, pp. 725–738.

Chun Chen received the BS degree in Mathematics from Xiamen University, China, in 1981, and his MS and Ph.D. degrees in Computer Science from Zhejiang University, China, in 1984 and 1990, respectively. He is a professor in College of Computer Science, Zhejiang University. His research interests include information retrieval, data mining, computer vision, computer graphics and embedded technology.

Lijun Zhang received the BS degree in Computer Science from Zhejiang University, China, in 2007. He is currently a candidate for a Ph.D. degree in Computer Science from Zhejiang University. His research interests include machine learning, information retrieval, and data mining.



Jiajun Bu received the BS and Ph.D. degrees in Computer Science from Zhejiang University, China, in 1995 and 2000, respectively. He is a professor in College of Computer Science, Zhejiang University. His research interests include embedded system, data mining, information retrieval and mobile database.



Can Wang received the BS degree in Economics, MS and Ph.D. degrees in Computer Science from Zhejiang University, China, in 1995, 2003 and 2009, respectively. He is currently a faculty member in College of Computer Science at Zhejiang University. His research interests include information retrieval, data mining and machine learning.

Wei Chen received the BS degree in Computer Science from Zhejiang University, China, in 2005. He is currently a candidate for a Ph.D. degree in Computer Science at Zhejiang University. His research interests include mobile and ubiquitous information systems, web information filtering and retrieval, and accessible computing.