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 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. Thus, 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.

© 2009 Elsevier B.V. All rights reserved.

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 effectiveness 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 matrices. 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
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:

\[
\mathbf{u}_{\text{opt}} = \arg\max_{\mathbf{u}} \mathbf{u}^T \mathbf{S} \mathbf{u}
\]

with the constraint

\[
\mathbf{u}^T \mathbf{u} = 1
\]

where \( \mathbf{S} \) is the data covariance matrix defined by

\[
\mathbf{S} = \frac{1}{m} \sum_{i=1}^{m} (\mathbf{x}_i - \overline{\mathbf{X}})(\mathbf{x}_i - \overline{\mathbf{X}})^T
\]

It can be proved that \( \mathbf{u}_{\text{opt}} \) equals to the eigenvector of \( \mathbf{S} \) that having the largest eigenvalue \( \lambda_1 \). 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 \( \mathbf{S} \).

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 \( \mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_m] \). In order to reduce the dimensionality, NMF finds two non-negative matrix \( \mathbf{W} \) and \( \mathbf{H} \) such that:

\[
\mathbf{X} \approx \mathbf{WH}
\]

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

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

\[
\mathbf{x}_i \approx \mathbf{W} \mathbf{h}_i, \quad i = 1, \ldots, m
\]

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 \( \mathbf{W} \) and \( \mathbf{H} \) [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 \( \mathbf{W} \), LE compute eigenvalues and eigenvectors for the generalized eigenvector problem:

\[
\mathbf{L} \mathbf{y} = \lambda \mathbf{D} \mathbf{y}
\]

where \( \mathbf{D} \) is the diagonal weight matrix with \( D_{ii} = \sum_{j} W_{ij} \), and \( \mathbf{L} = \mathbf{D} - \mathbf{W} \) 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 \( \mathbf{Y} = [\mathbf{y}_1; \ldots; \mathbf{y}_r] \). LE tries to map similar points as closely as possible. The objective function of LE is

\[
Y_{\text{opt}} = \arg\min_{\mathbf{Y}} \sum_{ij} \| \mathbf{Y}_i - \mathbf{Y}_j \|^2 W_{ij} = \text{tr}(\mathbf{Y}^T \mathbf{LY})
\]

with the constraint

\[
\mathbf{Y}^T \mathbf{D} \mathbf{Y} = I
\]

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 \( \overline{\mathbf{S}} \), a new diagonal weight matrix \( \mathbf{D} \), and a new graph Laplacian \( \overline{\mathbf{L}} \). Secondly, we calculate eigenvectors of \( \overline{\mathbf{L}} \) 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 \( r \) 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 \( Y \in \mathbb{R}^{m \times r} \), 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 \( \bar{c} \). 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 consistent with this assumption. Assume the maximum similarity is 1. \( W \) is modified as follows:

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

After modification, within class \( (c_k \text{ or } \bar{c}) \) links become tighter, and between class \( (c_k \text{ and } \bar{c}) \) 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 \( (c_k \text{ or } \bar{c}) \) 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 \( y = [y_1, \ldots, y_m]^T \) be the result of \( k \)-th map. We have the following objective function:

\[
y_{opt} = \arg\min_y \sum_{i,j} (y_i - y_j)^2 W_{ij} = \arg\min_y y^T L_k y
\]

with constraints

\[
\begin{align*}
y_i &= y_j & \text{ if } \mathbf{x}_i, \mathbf{x}_j \in l_k \\
y_i &= y_j & \text{ if } \mathbf{x}_i \in l_k, \mathbf{x}_j \in l_b, a \neq k, \text{ and } b \neq k \\
y_i &\neq y_j & \text{ if } \mathbf{x}_i \in l_k, \mathbf{x}_j \in l_b, \text{ and } a \neq k \\
y^T D_k y &= 1
\end{align*}
\]

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 \( P_k \in \{0, \pm 1\}^{m \times (2+m-p)} \), 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:

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

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

\[
y = P_k z
\]

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

\[
z_{opt} = \arg\min_z z^T P_k^T L_k P_k z
\]
with the constraint
\[ z^T P_k^T D_k P_k z = 1 \]

Note that three constraints of Eq. (7) are dropped. They are automatically guaranteed by equation \( y = P_k z \), 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:
\[ P_k^T L_k P_k z = \lambda P_k^T D_k P_k z \]

(9)

Let \( z_0 \) be the smallest eigenvalue of Eq. (9). We have the following proposition:

**Proposition 1.** The smallest eigenvalue \( z_0 \) is 0 with eigenvector \([1, 0, \ldots, 0]^T\), and the corresponding \( y \) is the constant vector 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 \( z \), \( P_k z \) 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
\[ P_k^T L_k P_k [1, 0, \ldots, 0]^T = P_k^T L_k = P_k^T D_k P_k = 0 \]

The corresponding \( y = P_k [1, 0, \ldots, 0]^T \) = 1. \( \square \)

Vector 1 is useless, since all the data points have the same representations. So we choose the second smallest eigenvector \( z_1 \). Let \( z = [z_1, z_2, \ldots, z_{m+p}]^T \), it is easy to check that
\[
\begin{aligned}
y &= \left[ (z_1^2 - z_2^2), \ldots, (z_1^2 - z_{m+p}^2), (z_2^2 - z_2^2), \ldots, (z_2^2 - z_{m+p}^2), \ldots, (z_{m+p}^2 - z_{m+p}^2) \right]^T \\
&= \left[ (z_1^2 - z_2^2), \ldots, (z_1^2 - z_{m+p}^2), (z_2^2 - z_2^2), \ldots, (z_2^2 - z_{m+p}^2), \ldots, (z_{m+p}^2 - z_{m+p}^2) \right]^T 
\end{aligned}
\]

So labeled points belongs to \( c_k \) are represented by \( z_1^2 + z_2^2 \), and other labeled points are represented by \( z_1^2 + z_2^2 \). 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 \( z_1^2 + z_2^2 \), and other unlabeled points will be mapped around \( z_1^2 - z_2^2 \).

3.3. The algorithm

Given a set of data points \( x_1, \ldots, x_m \in \mathbb{R}^n \) 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 \( x_i \). There are two variations [2]:
   (1) \( \epsilon \)-neighborhoods [parameter \( \epsilon \in \mathbb{R} \): Nodes \( i \) and \( j \) are connected by an edge if \( ||x_i - x_j|| < \epsilon \) where the norm is the usual Euclidean norm in \( \mathbb{R}^n \).
   (2) \( k \) nearest neighbors [parameter \( k \in \mathbb{N} \): Nodes \( i \) and \( j \) are connected by an edge if \( x_i \) is among \( k \) nearest neighbors of \( x_j \) or \( x_j \) is among \( k \) nearest neighbors of \( x_i \).

2. **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.

3. **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 \( z_1 \) of the generalized eigenvector problem (9). Then, \( y = P_k z_1 \) gives the result of the \( k \)-th map.

**Combining the result of each mapping.** Denote the results of all the \( r \) maps as \( y_1, \ldots, y_r \). The embedding coordinate of point \( x_i \) is given by the \( i \)-th row of \( Y = [y_1, \ldots, y_r] \).

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.

Reuter-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 (NMI), 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.

<table>
<thead>
<tr>
<th>CluID</th>
<th>DocNum</th>
<th>CluID</th>
<th>DocNum</th>
<th>CluID</th>
<th>DocNum</th>
</tr>
</thead>
<tbody>
<tr>
<td>c0</td>
<td>1844</td>
<td>c1</td>
<td>160</td>
<td>c2</td>
<td>76</td>
</tr>
<tr>
<td>c1</td>
<td>1828</td>
<td>c2</td>
<td>145</td>
<td>c2</td>
<td>74</td>
</tr>
<tr>
<td>c2</td>
<td>2222</td>
<td>c3</td>
<td>141</td>
<td>c3</td>
<td>72</td>
</tr>
<tr>
<td>c3</td>
<td>811</td>
<td>c4</td>
<td>140</td>
<td>c4</td>
<td>71</td>
</tr>
<tr>
<td>c4</td>
<td>441</td>
<td>c5</td>
<td>131</td>
<td>c5</td>
<td>66</td>
</tr>
<tr>
<td>c5</td>
<td>407</td>
<td>c6</td>
<td>123</td>
<td>c6</td>
<td>65</td>
</tr>
<tr>
<td>c6</td>
<td>272</td>
<td>c7</td>
<td>123</td>
<td>c7</td>
<td>63</td>
</tr>
<tr>
<td>c7</td>
<td>238</td>
<td>c8</td>
<td>120</td>
<td>c8</td>
<td>58</td>
</tr>
<tr>
<td>c8</td>
<td>226</td>
<td>c9</td>
<td>104</td>
<td>c9</td>
<td>56</td>
</tr>
<tr>
<td>c9</td>
<td>167</td>
<td>c10</td>
<td>98</td>
<td>c10</td>
<td>52</td>
</tr>
</tbody>
</table>

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

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

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 $\text{map}(p_i)$ 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 $MI(C, C')$ is defined as follows:

$$MI(C, C') = \sum_{c_i \in C, c'_j \in C'} p(c_i, c'_j) \cdot \log \frac{p(c_i, c'_j)}{p(c_i) \cdot p(c'_j)}$$ 

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:

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

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, $W_{ij} = W_{ji} = d_i d_j$. Otherwise, $W_{ij} = W_{ji} = 0$. Since each document vector $d_i$ is normalized to have unit length, so $0 \leq W_{ij} \leq 1$.

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.
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 \( MI \) 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 or the number of clusters 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.
Acknowledgment

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

References

[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.