Pattern Anal Applic (2017) 20:495–506
DOI 10.1007/s10044-015-0515-x

THEORETICAL ADVANCES

Spectral clustering based on similarity and dissimilarity criterion

Bangjun Wang[1,2] · Li Zhang[2] · Caili Wu[2] · Fan-zhang Li[2] · Zhao Zhang[2]

Received: 24 October 2014 / Accepted: 14 August 2015 / Published online: 8 September 2015

Abstract The clustering assumption is to maximize the within-cluster similarity and simultaneously to minimize the between-cluster similarity for a given unlabeled dataset. This paper deals with a new spectral clustering algorithm based on a similarity and dissimilarity criterion by incorporating a dissimilarity criterion into the normalized cut criterion. The within-cluster similarity and the between-cluster dissimilarity can be enhanced to result in good clustering performance. Experimental results on toy and real-world datasets show that the new spectral clustering algorithm has a promising performance.

Keywords Spectral clustering · Normalized cut · Similarity criterion · Dissimilarity criterion

1 Introduction

Being a powerful tool of unsupervised learning, cluster analysis, of course, is unlabeled-data-oriented [1, 2, 8] and widely used in the field of data mining, pattern recognition, and machine learning. Aiming to partition the given data into several clusters, clustering methods usually employ a general rule, called similarity criterion, which is to maximize the within-cluster similarity and to minimize the between-cluster similarity. As a simple and classical clustering method, the K-means clustering algorithm is expected to get a satisfied clustering result for spherical data [8]. However, if the data are non-spherical or seriously overlapping, the K-means clustering algorithm cannot perform well. Furthermore, being sensitive to the initial point, K-means often gets stuck in the local minima.

In recent years, spectral clustering has attracted a lot of attention in machine learning [4, 7, 9, 11, 13, 15, 17, 20–25]. Compared with K-means, spectral clustering is able to deal with the data with any manifold besides spherical one. In addition, spectral clustering has shown its advantage in applications to image segmentation and data mining. Presently, many improved versions on spectral clustering have been proposed. For example, Chen et al. proposed a parallel algorithm based on distributed system for spectral clustering to avoid the problems of limited memory and computational time when applying spectral clustering to process a large-scale dataset [4]. Since spectral clustering is completely unsupervised, researchers introduced the paired and constrained prior knowledge into spectral clustering, called semi-supervised spectral clustering [3, 13, 19, 20, 25].

The partition criterion plays an important role in the performance of spectral clustering. The common criteria include the min-cut [22], the average cut [17], the normalized cut [18], the min–max cut [7], the ratio cut [11, 21], etc. These criteria have their advantage and disadvantage. For example, the ratio cut criterion is able to avoid some weaknesses of min-cut so as to reduce the possibility of over-dividing; however, its running speed is slow. Although both the average cut and the normalized cut have a capacity to get a relatively precise partition, the normalized cut is more satisfying when they are applied to the same task. The min–max cut and the normalized cut have similar performance and can satisfy the minimization of between-cluster similarity and the maximization of within-cluster similarity, but the former with a more complex computation is better than the latter when data in different

✉ Bangjun Wang
wangbangjun@suda.edu.cn

[1] Beijing Jiaotong University, Beijing 100044, China

[2] Soochow University, Suzhou 215006, Jiangsu, China

clusters are partly overlapping. In a nutshell, all criteria are designed according to the similarity criterion.

From the discussion above, we know that the normalized cut criterion has some advantages. In this criterion, however, the between-cluster similarity has a negative impact on maximizing the within-cluster similarity. To reduce this negative effect, this paper presents a new criterion for spectral clustering, called similarity and dissimilarity (SAD) criterion. Based on the normalized cut criterion, SAD introduces the concept of dissimilarity so as to make the similarity and dissimilarity of samples more obvious and improves the clustering performance. Experimental results on artificial and real-word datasets show that the algorithm proposed here has a promising performance.

The contribution of this paper is to propose a similarity and dissimilarity criterion for spectral clustering. The rest of this paper is organized as follows. In Sect. 2, we introduce some related works for spectral clustering and the normalized cut. Section 3 presents the similarity and dissimilarity criterion for spectral clustering. We report experimental results in Sect. 4 and conclude this paper in Sect. 5.

2 Spectral clustering and normalized cut

2.1 Spectral clustering

Spectral clustering was proposed based on the spectrogram theory. The main idea behind spectral clustering is to cast clustering problems into the graph optimization ones. In the spectrogram theory, the optimal partition methods and criteria about graphs have been researched thoroughly, of which the basic rule is to make the similarity in each sub-graph be maximal and the similarity between sub-graphs be minimal. This rule is totally consistent with the basic assumption of clustering.

Suppose that we have an unlabeled data set $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ where $\mathbf{x}_i \in R^d$, R^d denotes d-dimensional real-valued sample space, and n is the number of samples. If each sample is treated as a vertex of graph and the edges are weighted according to the similarity between samples, then we can get a weighted and undirected graph G (V, E), where V represents the vertex set of graph G, $V = X$, and E denotes the edge set. As a result, we can formulate the clustering problems as the graph partition problems. Specifically, it requires dividing the graph G (V, E) into k subsets $X_1, X_2, \ldots, X_k$ which are mutual exclusion. The important thing is that the similarity in the subset X_i is maximized and the dissimilarity between the different subsets X_i and X_j is also maximized.

Generally specking, three steps are included when using spectral clustering:

1. Calculate the similarity matrix **W** among samples;
2. Obtain the eigenvectors of matrix **W** or other related matrix;
3. Implement the clustering of eigenvectors using classical clustering methods.

The first step is very important and now there are many methods to implement the construction of similarity matrix, such as the nearest neighbor and the full connection method Luxburg [15]. Here, the full connection method is considered. Gaussian kernel is typically selected as the measure of similarity between the sample $\mathbf{x}_i$ and $\mathbf{x}_j$ [13, 15, 18, 24]. The reason is that Gaussian kernel would give a large similarity for the two close samples and a small similarity for the two far samples, which is just the definition of similarity. In general, the elements of similarity matrix can be denoted as

$$W_{ij} = \exp\left(-\frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{\sigma^2}\right), \tag{1}$$

where σ is a pre-determined parameter. If let $\sigma^2 = \sigma_i\sigma_j$, then the similarity function described in Zelnik-Manor and Perona [24] can be written as

$$W_{ij} = \exp\left(-\frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{\sigma_i\sigma_j}\right), \tag{2}$$

where σ_i and σ_j are related with the two samples $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively. The similarity function (2) is more powerful than the similarity function (1) on the capacity of similarity representation since the similarity function (1) only takes into account one global parameter σ, but the similarity function (2) contains two local parameters σ_i and σ_j. Note that different eigenvectors generated in the second step when different partition criteria are employed would result in different partitions.

2.2 Normalized cut

Assume that a graph G is partitioned into two sub-graphs X_1 and X_2 where $X_1 \cup X_2 = V$ and $X_1 \cap X_2 = \varnothing$. In 2000, Shi and Malik proposed a normalized cut criterion for bipartition according to the spectrogram theory Shi and Malik [18]:

$$\min N\text{cut}(X_1, X_2) = \frac{\text{cut}(X_1, X_2)}{\text{assoc}(X_1, V)} + \frac{\text{cut}(X_1, X_2)}{\text{assoc}(X_2, V)}, \tag{3}$$

where cut(X_1, X_2) is the total connection from nodes in X_1 to nodes in X_2, or

$$\text{cut}(X_1, X_2) = \sum_{\mathbf{x}_u \in X_1} \sum_{\mathbf{x}_v \in X_2} W_{uv}$$

and assoc(X_i, V) is the total connection from nodes in X_i to all nodes in the graph, or

$$\operatorname{assoc}(X_i, V) = \sum_{x_u \in X_i} \sum_{x_p \in V} W_{up}$$

In fact, cut(X_1, X_2) can be taken as the between-cluster similarity, and assoc(X_i, V) is the summation of the between-cluster similarity and within-cluster similarity.

The normalized cut criterion can not only maximize the within-cluster similarity but also maximize the between-cluster dissimilarity. What is more important is that the normalized cut can also efficiently avoid the preference of small region partition which often occurs in the min-cut criterion. Another form of (3) can be represented as [18]

$$\min_{y} \frac{\mathbf{y}^T(\mathbf{D} - \mathbf{W})\mathbf{y}}{\mathbf{y}^T\mathbf{D}\mathbf{y}} \quad (4)$$
$$\text{s.t.} \quad \mathbf{y}^T\mathbf{D}\mathbf{y} = 1$$
$$\mathbf{y}^T\mathbf{D}\mathbf{1} = 0,$$

where **1** is a vector with all elements of 1 and the matrix $\mathbf{D} \in R^{n\times n}$ is a diagonal one with the elements of $D_{ii} = \sum_{j=1}^{n} W_{ij}$, and the indicator vector $y = [y_1, y_2,\ldots, y_n]^T$. Obviously, the objective of (4) is a Rayleigh equation so that the continuous and loose form of indicator vector y can be taken into account. Then the solution to (4) is equivalent to the solution to $(\mathbf{D} - \mathbf{W})\mathbf{y} = \lambda\mathbf{D}\mathbf{y}$.

The goal of the normalized cut is to maximize the within-cluster similarity and minimize the between-cluster similarity, but the denominator of the objective in (4), $\mathbf{y}^T\mathbf{D}\mathbf{y}$, can be roughly viewed as the sum of the within-cluster similarity and the between-cluster similarity. It is a contradiction that we maximize $\mathbf{y}^{\mathrm{T}}\mathbf{D}\mathbf{y}$ while making the between-cluster similarity be minimized. But how to explain the good performance obtained by normalized cut? The main reason is that the numerator term of the objective in (4) emphasizes the minimization of the between-cluster similarity. Meanwhile, if the within-cluster similarity is large enough, the effect of between-cluster similarity on the denominator can be ignored. But the maximization of between-cluster similarity in the denominator still has some negative effects on the clustering performance.

3 Spectral clustering method based on the similarity and dissimilarity

In this section, we introduce a dissimilarity criterion, propose the criterion of similarity and dissimilarity (SAD), and describe the spectral clustering method based on SAD in detail.

3.1 Dissimilarity criterion

The clustering assumption is to maximize the within-cluster similarity and simultaneously to minimize the between-cluster similarity, which can also be described as minimizing the within-cluster dissimilarity and maximizing the between-cluster dissimilarity, called dissimilarity criterion. Let **Q** be the dissimilarity matrix, where Q_{ij} is in direct proportion to the distance, the smaller the similarity is and the greater the dissimilarity do. If (2) is selected to measure the similarity, then the measure of dissimilarity is given by

$$Q_{ij} = 1 - \exp\left(-\frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{\sigma_i\sigma_j}\right) \quad (5)$$

According to (5), the within-cluster dissimilarity can be defined as

$$\sum_{x_i \in X_1} \sum_{x_j \in X_1} Q_{ij} + \sum_{x_i \in X_2} \sum_{x_j \in X_2} Q_{ij}, \quad (6)$$

and the between-cluster dissimilarity can be described as

$$\sum_{x_i \in X_1} \sum_{x_j \in X_2} Q_{ij} + \sum_{x_i \in X_2} \sum_{x_j \in X_1} Q_{ij} \quad (7)$$

Define a new indicator vector $z = [z_1, z_2,\ldots, z_n]^T$. If $\mathbf{x}_i \in X_1$, then $z_i = 1$; otherwise $z_i = -1$. Then **z** can reflect the clustering results. $(1 + \mathbf{z})/2$ and $(1 - \mathbf{z})/2$, respectively, represent the indicator vectors of $\mathbf{x}_i \in X_1$ and $\mathbf{x}_i \in X_2$. Thus, the dissimilarity between samples in the same cluster and between clusters can be, respectively, denoted as

$$\begin{aligned}&\sum_{x_i \in X_1} \sum_{x_j \in X_2} Q_{ij} + \sum_{x_i \in X_2} \sum_{x_j \in X_1} Q_{ij} \\ &= \frac{1}{4}(1 - \mathbf{z})^T\mathbf{Q}(1 - \mathbf{z}) + \frac{1}{4}(1 + \mathbf{z})^T\mathbf{Q}(1 + \mathbf{z})\end{aligned} \quad (8)$$

and

$$\sum_{x_i \in X_1} \sum_{x_j \in X_1} Q_{ij} + \sum_{x_i \in X_2} \sum_{x_j \in X_2} Q_{ij} = \frac{2}{4}(1 + \mathbf{z})^T\mathbf{Q}(1 - \mathbf{z}) \quad (9)$$

We combine (8) and (9), and have the following objective:

$$\begin{aligned}\max \quad &\sum_{x_i \in X_1} \sum_{x_j \in X_2} Q_{ij} + \sum_{x_i \in X_2} \sum_{x_j \in X_1} Q_{ij} - \sum_{x_i \in X_1} \sum_{x_j \in X_1} Q_{ij} \\ &- \sum_{x_i \in X_2} \sum_{x_j \in X_2} Q_{ij} = -\mathbf{z}^T\mathbf{Q}\mathbf{z},\end{aligned} \quad (10)$$

which can implement minimizing the within-cluster dissimilarity and maximizing the between-cluster dissimilarity. In the normalized cut, we optimize the vector **y** instead of vector $\mathbf{z}$. Thus, we perform the similar replacement. Then (10) can be rewritten as

$$\max\ -\mathbf{y}^T\mathbf{Q}\mathbf{y}, \tag{11}$$

where $y_i \in \{1, -b\}$, and $b = \frac{\sum_{z_i>0} D_{ii}}{\sum_{z_i<0} D_{ii}}$

3.2 Similarity and dissimilarity criterion

According to the analysis in Sect. 2, we want to further reduce the effect of the between-cluster similarity in the denominator of the objective in (4) in the normalized cut. Thus, we introduce the dissimilarity criterion (11) into the denominator of the objective in (4), and get

$$\min_{y} \frac{\mathbf{y}^T(\mathbf{D}-\mathbf{W})\mathbf{y}}{(1-m)\mathbf{y}^T\mathbf{D}\mathbf{y} - m\mathbf{y}^T\mathbf{Q}\mathbf{y}} \tag{12}$$
$$s.t.\quad \mathbf{y}^T\mathbf{D}\mathbf{y} = 1,$$
$$\mathbf{y}^T\mathbf{D}\mathbf{1} = 0,$$

where $0 \le m \le 1$ is a trade-off factor that determines the influence of dissimilarity criterion. The larger the factor m, the greater the influence. If $m = 0$, then the method proposed here is totally the same to the spectral clustering method based on normalized cut.

Theorem 1. *Given $0 \le m \le 1$ in (12), the between-cluster similarity in the denominator of (12) has a smaller weight than that of (4).*

The proof of Theorem 1 is described in Appendix A. Theorem 1 states that the effect of the between-cluster similarity on the maximization of within-cluster similarity can be reduced provided that the dissimilarity criterion is introduced into the normalized cut criterion. For a better analysis on the similarity and dissimilarity criterion, the objective in (12) can be rewritten as

$$\min_{y} \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}(y_i - y_j)^2 W_{ij}}{(1-m)\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j D_{ij} - m\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j Q_{ij}} \tag{13}$$

By analyzing (13), we have the following remarks:

1. When Q_{ij} is very small, Q_{ij} makes less contribution to the second term of denominator, and the two samples $\mathbf{x}_i$ and $\mathbf{x}_j$ tend to be partitioned into the same cluster. At the same time, the similarity W_{ij} in the numerator is large. If we want to minimize (13), it requires the value of $(y_i - y_j)^2$ to be very small so as to show the similarity between the two samples $\mathbf{x}_i$ and $\mathbf{x}_j$.
2. When Q_{ij} is very large, the two samples $\mathbf{x}_i$ and $\mathbf{x}_j$ tend to be partitioned into different clusters. Due to the effect of Q_{ij}, the value of mQ_{ij} in the denominator of (13) is also very large. In this case, even though we increase the value of $(y_i - y_j)^2$ to make the data points being far away from each other, it would not have a huge effect on the optimization of the objective. By doing so, we can increase the dissimilarity of between-cluster samples.

3.3 Spectral clustering based on the similarity and dissimilarity criterion

By introducing the dissimilarity criterion, we design a new clustering criterion, which further increases the within-cluster similarity and the between-cluster dissimilarity. As a consequence, a better clustering result is guaranteed. The detailed description of the new method is shown in Algorithm 1. Similar to the standard spectral clustering method, the dimensionality of samples is identical to the number of clusters.

In the following, we analyze the space complexity and computational complexity of the spectral clustering method based on SAD. For spectral clustering based on the normalized cut, the space complexity coming from the storage of similarity matrix is $O(n^2)$, and the computational complexity includes the computation of constructing the similarity matrix, of the generalized eigen-decomposition, and of the used classical clustering method [13]. Compared with the spectral clustering method based on the normalized cut, the proposed method has almost the same space complexity and computational complexity. Since the dissimilarity matrix can be obtained by directly using the similarity matrix, no additional space is required and its space complexity is still $O(n^2)$. The new method just involves the operations of the matrix addition and of the multiplication of a real number and a matrix, which would not change the computational complexity. Thus, the spectral clustering method based on SAD has almost the same computation complexity with the one based on the normalized cut.

Algorithm 1 Spectral clustering method based on similarity and dissimilarity

```
Input: The set of samples $X = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$ where
$\mathbf{x}_i \in R^d$, the number of clusters $k$, and the trade-off factor $m$
Output: $k$ disjoint subsets $\{X_j\}_{j=1}^{k}$
1. Calculate the similarity matrix $\mathbf{W}$ according to (2), and
   generate the dissimilarity matrix $\mathbf{Q}$ according to (5)
   and the matrix $\mathbf{D}$.
2. Generalized eigen-decompose $(\mathbf{D}-\mathbf{W})$ and
   $((1-m)\mathbf{D}-m\mathbf{Q})$. Sort the eigenvalues in the
   ascending order, or $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$. The
   corresponding eigenvectors are $\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n$ with
   $\mathbf{y}_i \in R^n$.
3. Keep the set of eigenvectors
   $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_k] \in R^{n\times k}$ corresponding to the
   smallest $k$ eigenvalues. The $i$ th row in $\mathbf{Y}$ is an
   image of $\mathbf{x}_i$ in the new feature subspace.
4. Adopt the classical clustering method on $\mathbf{Y}$, and get
   $k$ disjoint subsets $\{X_i\}_{i=1}^{k}$.
```

4 Experimental results

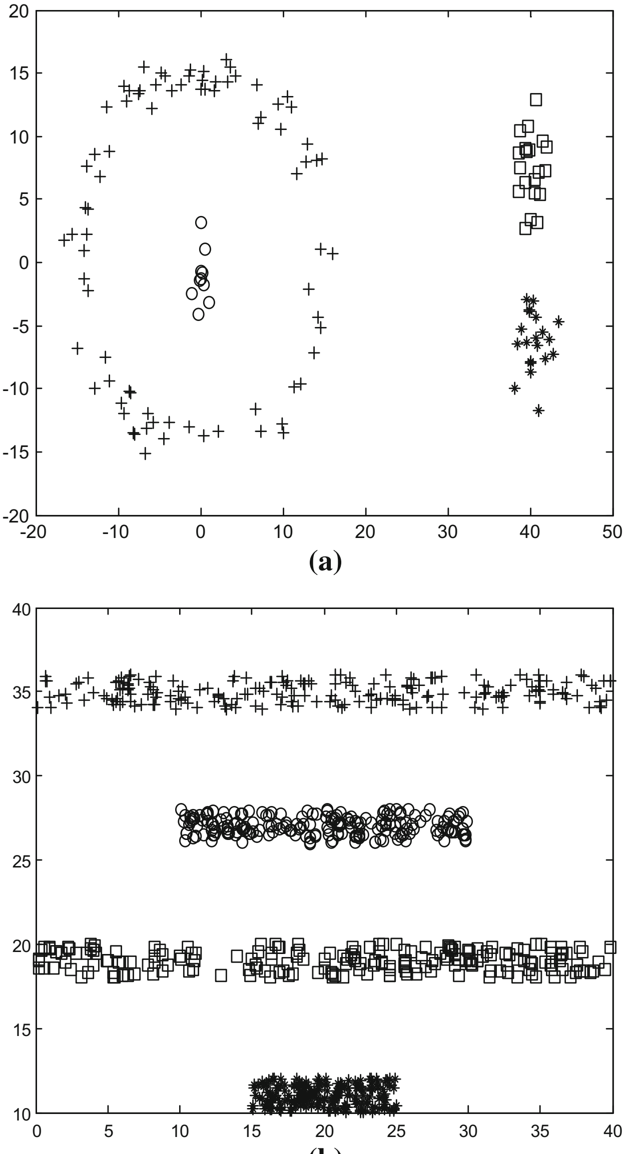
To validate the efficiency of the proposed method here, we perform experiments on artificial datasets, nine UCI datasets and one face dataset. Compared methods include the classical K-means method, the normalized cut-based spectral clustering (Ncut) method, and similarity and dissimilarity-based spectral clustering (SAD) method. In addition, K-means is also used in both Ncut and SAD as the subsequent clustering algorithm.

All numerical experiments are performed on the personal computer with a 1.8 GHz Pentium III and 2G bytes of memory. This computer runs on Windows XP, with MAT-LAB 7.1 and VC++6.0 compiler installed.

4.1 Artificial datasets

4.1.1 Data description

Consider two artificial datasets containing certain manifold structure. Dataset 1 being from Shi and Malik [18] is composed of 4 subsets with different distributions, as shown in Fig. 1a. Specifically, one subset consists of 80 data points forming a circle distribution, and the other three subsets are the data obeying different Gaussian



(a)

(b)

Fig. 1 Two artificial datasets. **a** Dataset 1, **b** dataset 2

distributions, respectively. For the other three subsets, the one locating in the circle consists 10 data points and the other two ones locating outside of circle both consist of 20 data points. Dataset 2 being from Zelnik-Manor and Perona [24] is also composed of 4 subsets with different distributions, as shown in Fig. 1b. Each subset consists of 200 data points coming from a uniformly random distribution taking value from a rectangle with different height and width.

4.1.2 Selection of similarity function parameter

Gaussian kernel parameters need to be set before calculating the similarity matrix. An adaptive method was presented to tune the parameter in Zelnik-Manor and Perona [24]. (2) is used to compute the similarity matrix, where σ_i is the square root of distance between the sample $\mathbf{x}_i$ and its pth neighbor. We consider how the parameter p affects the clustering performance, where $p \in \{1, 2,\ldots, 15\}$. For both Dataset 1 and Dataset 2, we randomly generate 50 training sets (for 50 experiments), respectively. Here, only Ncut is performed. The average results of 50 experiments are shown in Fig. 2. From the results, we can know that the method can obtain satisfying performance when setting the number of neighbors to be 7, just being the conclusion of Zelnik-Manor and Perona [24]. Thus, in the following experiments let $p = 7$.

4.1.3 Comparison of SAD with K-means and Ncut

In SAD, there is another parameter m. How to determine the value of m is discussed in the next subsection. Here, let m vary in the set $\{0, 0.1, 0.2,\ldots, 0.9, 1\}$. Under the same experimental conditions as before, we compare K-means, Ncut and SAD on the new 100 training sets, and the average error of 100 experiments is shown in Fig. 3. Observation on Fig. 3 indicates that SAD outperforms Ncut with different parameter m. In addition, the performance of SAD can be effected by the parameter m on these two artificial datasets.

In the experiment, we find that SAD with $m = 0$ does not have the same average performance as Ncut. The main reason is that we take K-means as the subsequent clustering method. It is well known that K-means is instable. From both theory and experiment, we can show that the feature vectors obtained by SAD with $m = 0$ are totally equivalent to the ones obtained by Ncut. Next, we would verify this from an experimental viewpoint.

In SAD, let $m = 0$. We perform experiments only on Dataset 1. Ncut and SAD, respectively, obtain their eigenvectors after spectral mapping, as shown in Fig. 4a. Since Dataset 1 contains 4 classes, the eigenvectors corresponding to the four smallest eigenvalues are selected.

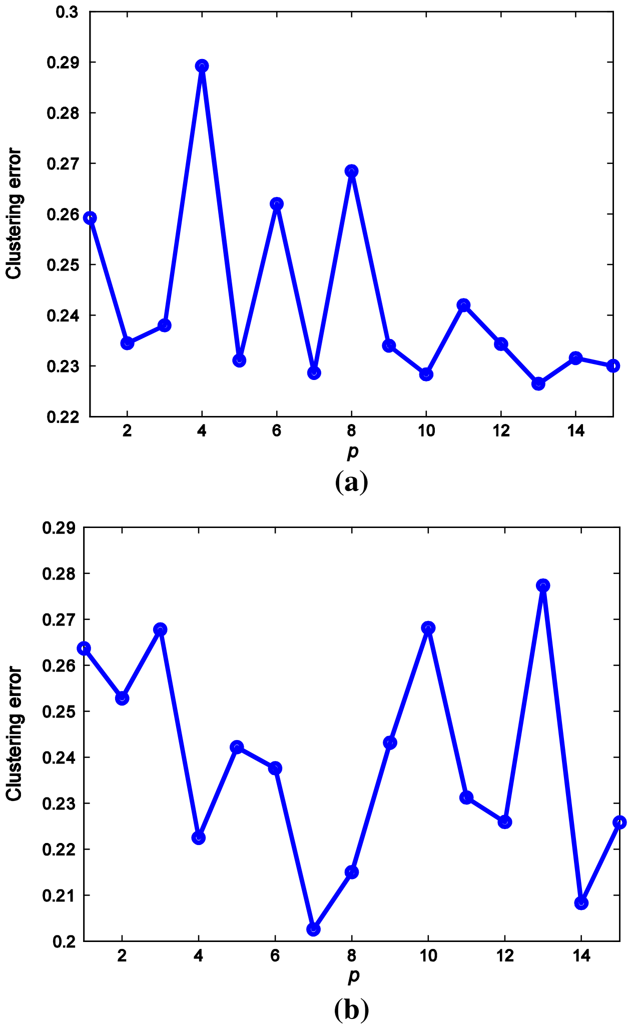


Fig. 2 Clustering error of Ncut vs. the number of neighbor. **a** Dataset 1, **b** dataset 2

From Fig. 4a, it is obvious that the two methods generate the same eigenvalues and eigenvectors. Of course, SAD with $m \neq 0$ would generate different eigenvectors with Ncut, see Fig. 4b with $m = 1$. In Fig. 4, the abscissa represents the index of samples, where the indexes from 1 to 80 belong to the first class, from 81 to 90 belong to the second one, from 91 to 110 is the third one, and from 111 to 130 the fourth one.

4.1.4 Selection of the parameter m for SAD

From the discussion before, the parameter m has an effect on the clustering error (CE) performance of SAD. However, we cannot select the optimal m according to the best clustering result which is unknown in clustering problems. Fortunately, many validity indices to measure the clustering performance have been proposed, such as the Davies-Bouldin (DB) index [5], the Dunn index [2], the SIL index

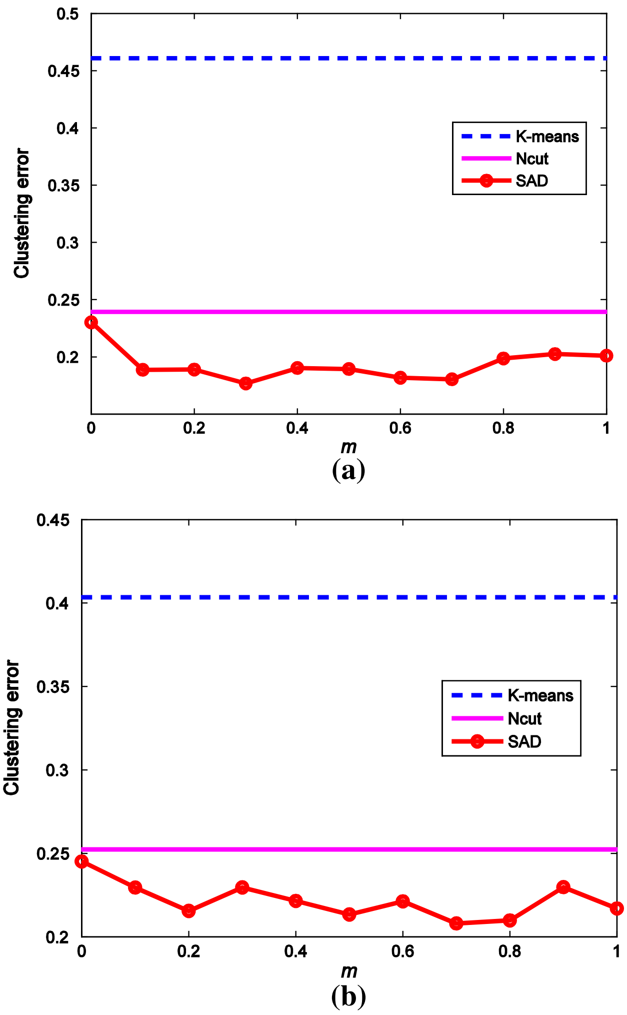


Fig. 3 Clustering error obtained by different clustering methods on two artificial datasets. **a** Dataset 1, **b** dataset 2

[16], and the negentropy index [12]. In theory, the best clustering partition is the one that minimizes the DB index, or minimizes the negentropy index, or maximizes the Dunn index, or maximizes the SIL index. It is easy to compute these indices for a given clustering partition. Thus, we can determine the optimal m as the corresponding one when these indices achieve their maximal or minimal value.

m varies in the set {0, 0.1, 0.2,…, 0.9, 1}. According to the partition obtained by SAD, we compute these indices and CE in one trial and show them in Fig. 5. In Fig. 5a, the best CE for Dataset 1 is obtained at $m = 0$, $m = 0.1$, $m = 0.2$, $m = 0.9$, and $m = 1$, respectively. At these points, the SIL index achieves its maximal value (or 0.0591), and the negentropy index also achieves its minimal value (or 1.0021). However, we cannot find the corresponding relationship between both the DB and the Dunn indices and CE, since these two indices are assuming that the the clusters are spherical [12]. Thus, the two indices may not be effective for the selection of m in Datasets 1 and 2 here. We have the same conclusion on Dataset 2, see Fig. 5b. When the maximal SIL index and the minimal negentropy index are obtained, the best partition is generated. Thus, we can use the SIL index or the negentropy index to select $m$for SAD.

4.2 Heart dataset

The Heart dataset contains 303 data points with 13-dimensional, which are divided into two classes with one is 164 data points and another is 139 data points. In SAD, m varies in the set $\{0.1, 0.2\ , \ldots, 0.9, 1\}$. The index values for SIL, negentropy, and CE are shown in Fig. 6. The corresponding relationship between the negenropy index and CE holds true on the Heart dataset. In other words, when the negentropy index achieves its minimum at $m = 0.1$, CE also achieves its minimum 23.10 % at the same parameter. However, the SIL index fails to select the optimal parameter. Among 50 trials, the best CE of K-means is 28.38 % and the best one of Ncut is 24.42 %.

It is known that if we choose spectral clustering, the clustering result greatly depends on two factors, one of which is the representation of dataset through eigenvectors and the other one is the performance of classical clustering method in the feature space. We hope the difference between SAD and Ncut is the result of the first factor, not the instability of K-means. In other word, we desire the eigenvectors obtained by SAD is a better representation so as to be helpful for the subsequent clustering.

We, respectively, use Ncut and SAD to perform generalized eigen-decomposition and then only get the eigenvector corresponding to the second minimal eigenvalue because the dataset is only divided into two classes. Figure 7 shows the distribution of data points after decomposition with $m = 0.1$ for SAD. The abscissa denotes the index of samples, and the indexes from 1 to 164 are of the first cluster, and from 165 to end are of the second cluster. The ordinate is the mapping value.

Instead of using K-means as the subsequent clustering method, the simplest threshold method is used here. The mean of all the samples in the feature space is set as the threshold; then here we have threshold -0.0526 for Ncut and 0.1665 for SAD, respectively. We count the number of samples in two regions by taking the threshold as the division point and report them in Table 1. There are 120 correct data points in the first cluster and 106 correct data points in the second cluster for Ncut. For SAD, we can get more correct points, 126 in the first cluster and 111 in the second cluster, respectively. That is to say, SAD increases the number of samples of the clusters and would result in better clustering performance. Therefore, SAD can map most data points in the same cluster into the same region in the feature space, which is conducive to the subsequent clustering in the feature space.

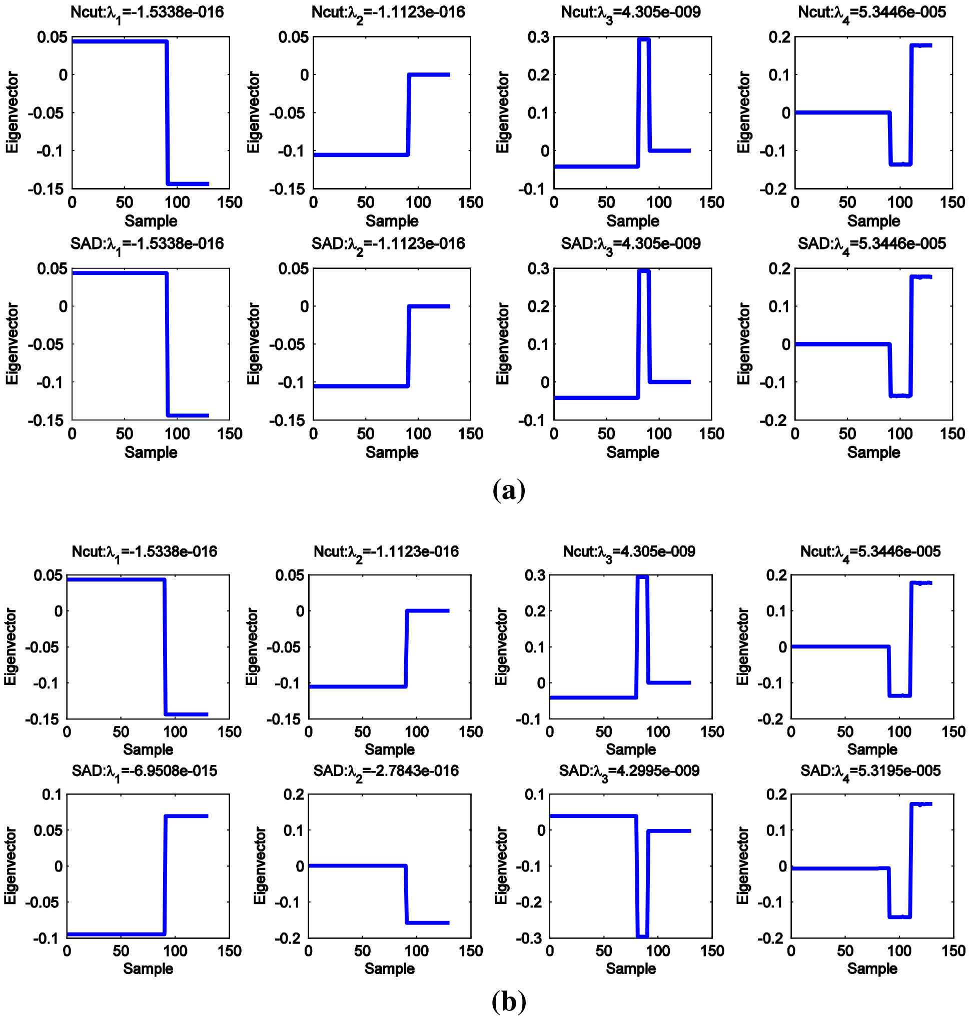


Fig. 4 Comparison of eigenvectors after spectral mapping for Ncut and SAD. **a** Ncut and SAD with $m = 0$, **b** Ncut and SAD with $m = 1$

4.3 More UCI dataset

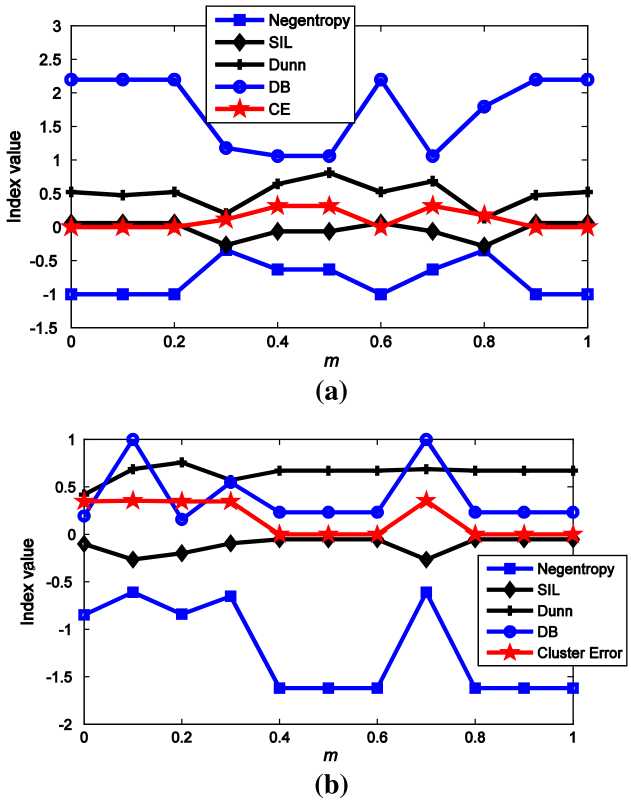
We also test our method on more UCI datasets, which are described in Table 2. Since the labels of samples are given in these UCI datasets, the clustering error rate can be calculated and taken as the evaluation performance. Due to the instability of K-means, we perform 50 times for each dataset and report the best results of K-means and Ncut in Table 3. We also list the results on the Heart dataset in this table. For SAD, we use the negentropy index to select the optimal m and report the corresponding CE.

From Table 3, SAD outperforms K-means in five out of nine data sets, especially in the Heart dataset. Compared with Ncut, SAD is better in five datasets and has the same results on the two datasets. The advantage on the Ionosphere dataset shows SAD is more promising than Ncut.

We expect that the minimum negentropy index is corresponding to the best CE. However, we find that it happens only on six datasets: Liver, Heart, Musk, Sonar, Wine, and Wpbc. In other words, we can select an appropriate parameter m for SAD on these six datasets. For other three datasets, Pima, Wdbc, and Ionoshpere, the negentropy index does not work well. Figure 8 shows the negentropy index and CE vs. m. The best CE on Pima obtained by SAD should be 31.77 % at $m = 0.4$, which is supposed to be the best performance among three clustering methods. However, according to the minimal negentropy index, 34.38 % at $m = 0.3$ is listed in Table 3. We encounter the same situation on the Wdbc dataset. The best CE 6.33 % corresponds to the maximal negentroy index, which is unreasonable. Thus, we pick up the wrong m, which leads to a bad performance of SAD.

4.4 UMIST face dataset

Here, we apply the clustering algorithm to the UMIST face dataset [10], which is a multiview database and consists of 574 cropped gray-scale images of 20 subjects, each covering a wide range of poses from profile to frontal views as well as race, gender, and appearance. Each image in the



(a)

(b)

Fig. 5 Five indices vs. *m* on **a** dataset 1, and **b** dataset 2

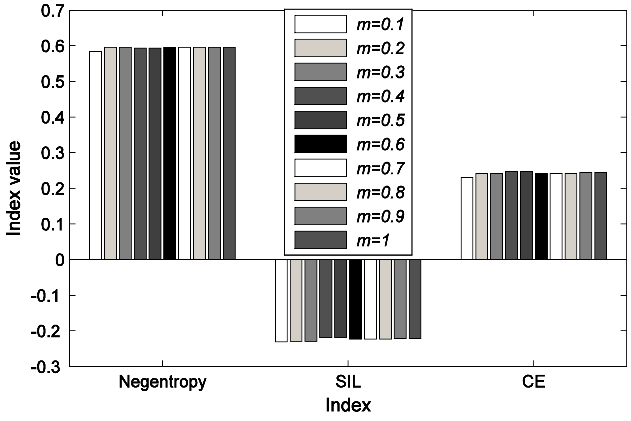


Fig. 6 Three indices vs. *m* on the heart dataset

database is resized into 32 × 32, and the resulting standardized input vectors are of dimensionality 1024. Figure 9 depicts some sample images of a typical subject in the UMIST database.

Empirically, let $m = 0.5$ for SAD. The other experimental setting is the same as Sect. 4.3. We list the maximum, minimum, and average clustering errors among 20 runs in Fig. 10, respectively. In Fig. 10, “max” means the maximum value of 20 runs, “min” means the minimum value of 20 runs, and “average” means the average value

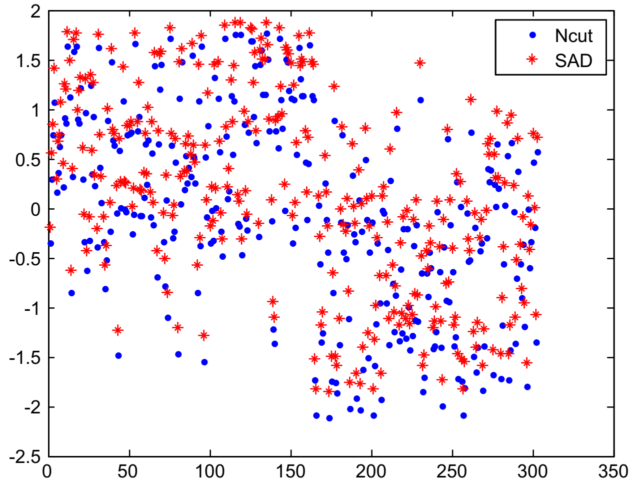


Fig. 7 Distribution of data after projection on the heart dataset

Table 1 Sample number in different clusters on the heart dataset

	Ncut		SAD	
	≥−0.0526	<−0.0526	≥0.1665	<0.1665
Cluster one data	120	44	126	38
Cluster two data	33	106	28	111

Table 2 Feature description of Eight UCI datasets

Dataset	Feature	Class	Size
Liver	6	3	345
Ionosphere	34	2	351
Musk	66	2	476
Sonar	60	2	208
Pima	8	2	768
Wpbc	33	2	198
Wdbc	30	2	569
Wine	13	3	170

on 20 runs. Obviously, SAD is superior to Ncut and K-means. Note that the face data are a small size sample data since the feature number is greater than the sample number. The best clustering error is only 42.33 % obtained by SAD. It is hard to get a satisfied clustering result for these clustering algorithms.

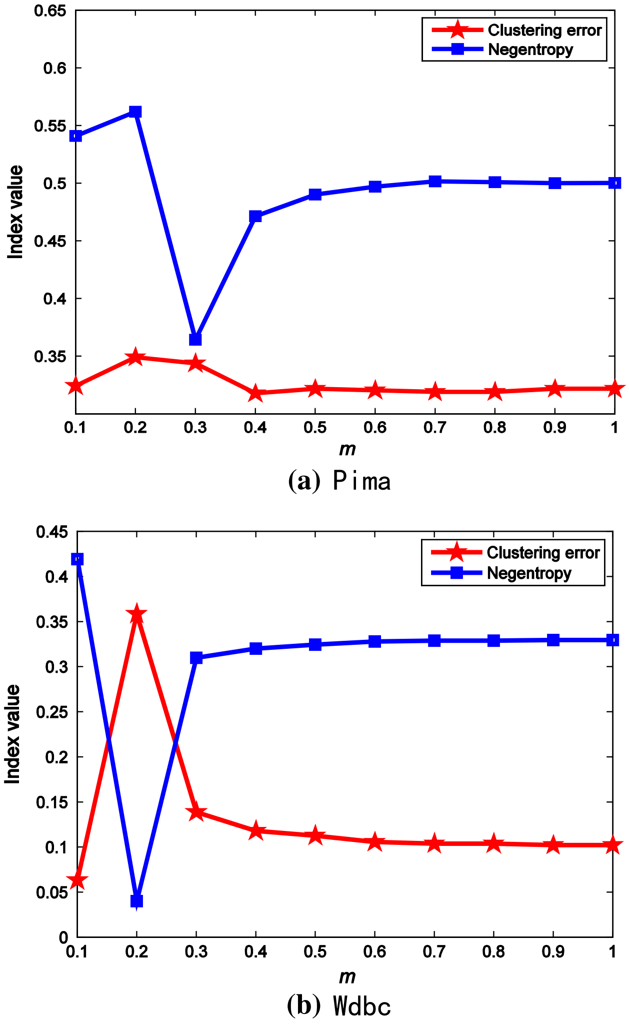
4.5 Statistical comparisons over 12 datasets

In the previous section, we have conducted experiments on 12 datasets, including 2 toy datasets, 9 UCI datasets, and one face dataset. Statistical tests on multiple datasets for the three algorithms are preferred for comparing different algorithms over multiple datasets. Here, we conduct

Table 3 Clustering error rate on the UCI dataset (%)

Dataset	K-means	Ncut	SAD
Liver	44.35	43.48	**42.90**
Ionosphere	28.77	35.61	**26.78**
Musk	**45.17**	46.85	45.80
Sonar	**44.23**	46.63	46.63
Pima	33.20	**32.94**	34.38
Wpbc	39.90	32.32	**31.31**
Wdbc	**7.21**	10.02	35.85
Wine	4.49	**3.37**	**3.37**
Heart	28.38	24.42	**23.10**

Bold values are used to indicate the best performance or lowest error rate in the same datasets with different methods, SAD outperforms K-means in five out of nine data sets, especially in the Heart dataset. Compared with Ncut, SAD is better in five datasets and has the same results on the two datasets. The advantage on the Ionosphere dataset shows SAD is more promising than Ncut



(a) Pima

(b) Wdbc

Fig. 8 Two indices vs. *m* on **a** Pima, and **b** Wdbc

Fig. 9 Some image samples of one person from the UMIST dataset

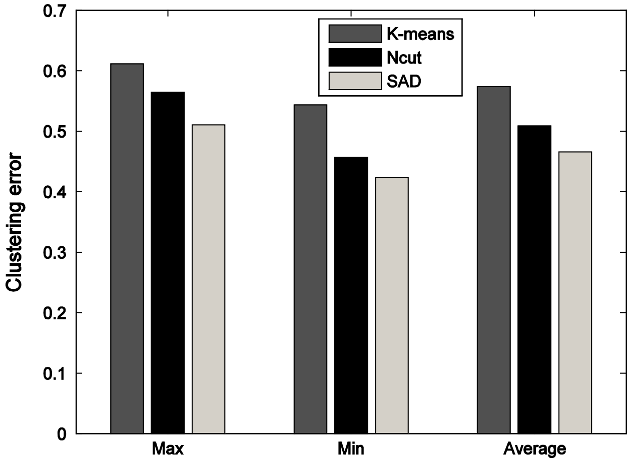


Fig. 10 Clustering error of 20 trials on the UMIST dataset. "max" means the maximum value of 20 runs, "min" means the minimum value of 20 runs, and "average" means the average value on 20 runs

statistical tests over 12 datasets by using the Friedman test (Friedman 1998; [6] with the corresponding post hoc tests.

The Friedman test is a nonparametric equivalence of the repeated-measures analysis of variance (ANOVA) under the null hypothesis that all the algorithms are equivalent and so their ranks should be equal [6]. According to the results described above, we can get the average ranks of three algorithms, 2.42 for K-means, 2.17 for Ncut, and 1.42 for SAD. The p value of the Friedman test is 0.034, which is less than 0.05. In other words, we can reject the null hypothesis and proceed with a post hoc test. Here, the Bonferroni-Dunn test is taken as post hoc tests. The performance of pairwise algorithms is significantly different if the corresponding average ranks differ by at least the critical difference (CD)

$$\mathrm{CD} = q_\alpha \sqrt{\frac{k(k+1)}{6N}}, \tag{14}$$

where k is the number of algorithms, N is the number of datasets, and q_α is the critical value which can be found in Demšar [6]. In our case, $k = 3$, $N = 12$, and $q_{0.1} = 1.96$ where the subscript 0.10 is the threshold value. Then, we have $\mathrm{CD} = 0.80$. The difference of average ranks between

SAD and the other two algorithms is 1.00 and 0.75, respectively. Thus, we have the following conclusion:

- SAD is significantly better than K-means since the rank difference between them is 1.00, which is greater than the CD value.
- There is no significant difference between SAD and Ncut. Note that the rank difference between them is 0.75, which approaches to the CD value. Thus, the advantage of SAD is still obvious compared with Ncut.

5 Conclusions

This paper proposes a spectral clustering method based on the similarity and dissimilarity criterion, which is an extension of the spectral clustering method of the normalized cut criterion. The new clustering method can increase the within-scatter.

Similarity and the between-scatter dissimilarity. Experimental results on the artificial datasets, UCI datasets, and one face dataset show that the new method indeed works well. We also conduct statistical tests on these datasets for the compared algorithms. Statistical results indicate that there is no significant difference between SAD and Ncut, but SAD is significantly better than K-means.

In the experiments, we find that the parameter m has an effect on the performance of SAD and use the negentropy index to select m. In theory, SAD always could get better CE than Ncut when $0 < m \leq 1$. However, the negentropy index is not very efficient for some datasets, such as Pima and Wdbc. We would further research how to adaptively select a proper value. Besides, the study about spectral clustering also includes the process of large-scale data [4], semi-supervised clustering with constrains [3, 13, 19, 20, 25], and non-negative sparse spectral clustering [14]. How to combine our method with these researches is also the next work we will consider. We expect that semi-supervised clustering can improve the clustering performance on the UMIST dataset.

Appendix A

Proof: To prove Theorem 1, we need to prove that the weight of the between-cluster similarity in $(1-m)\mathbf{y}^T\mathbf{Dy} - m\mathbf{y}^T\mathbf{Qy}$ is less than that in $\mathbf{y}^T\mathbf{Dy}$. Thus, the between-cluster similarity would have less effect on the maximization of the within-cluster similarity.

Without loss of generality, assume that a graph G can be partitioned into two disjoint sub-graphs X_1 and X_2. Consider the continuous and loose form of the indicator vector $\mathbf{y}$, and let $y_i \in \{1, -b\}$ with $b = \frac{\sum_{z_i>0} D_i}{\sum_{z_i<0} D_{ii}}$ and indicator vector $\mathbf{z} = [z_1,\ldots, z_n]$. First, we expand the denominator of the objective in (4) and simplify it, and have

$$\begin{aligned}\mathbf{y}^T\mathbf{Dy} = & \left(\sum_{x_i\in X_1}\sum_{x_j\in X_1} W_{ij} + b^2 \sum_{x_i\in X_2}\sum_{x_j\in X_2} W_{ij}\right)\\ & + \left(\sum_{x_i\in X_1}\sum_{x_j\in X_2} W_{ij} + b^2 \sum_{x_i\in X_2}\sum_{x_j\in X_1} W_{ij}\right)\\ & = Sim_w + (1+b^2)Sim_b,\end{aligned} \tag{15}$$

where Sim_w is the sum of the first two terms in (15) which can be viewed as the within-cluster similarity, and $Sim_b = \sum_{x_i\in X_2}\sum_{x_j\in X_1} W_{ij}$ denotes the between-cluster similarity. Next, we expand the denominator of the objective in (12), and have

$$\begin{aligned} & (1-m)\mathbf{y}^T\mathbf{Dy} - m\mathbf{y}^T\mathbf{Qy}\\ & = (1-m)\left(\sum_{x_i\in X_1}\sum_{x_j\in X_1} W_{ij}\right)\\ & \quad + (1-m)\left(b^2\sum_{x_i\in X_2}\sum_{x_j\in X_2} W_{ij} + (1+b^2)\sum_{x_i\in X_2}\sum_{x_j\in X_1} W_{ij}\right)\\ & \quad - m\left(\sum_{x_i\in X_1}\sum_{x_j\in X_1} Q_{ij} + b^2\sum_{x_i\in X_2}\sum_{x_j\in X_2} Q_{ij} - 2b\sum_{x_i\in X_1}\sum_{x_j\in X_2} Q_{ij}\right)\end{aligned} \tag{16}$$

Since $Q_{ij} = 1 - W_{ij}$, we substitute it into (16) and get

$$\begin{aligned} & (1-m)\mathbf{y}^T\mathbf{Dy} - m\mathbf{y}^T\mathbf{Dy}\\ & = \left(\sum_{x_i\in X_1}\sum_{x_j\in X_1} W_{ij} + b^2\sum_{x_i\in X_2}\sum_{x_j\in X_2} W_{ij}\right)\\ & \quad + \left(\left((1+b^2) - m(1+b^2)\right)\sum_{x_i\in X_1}\sum_{x_j\in X_2} W_{ij}\right)\\ & = Sim_w + \left((1+b^2) - m(1+b^2)\right)Sim_b\end{aligned} \tag{17}$$

Comparing (15) with (17), we can see that the difference between them is the weight of the between-cluster similarity. In addition, it is obvious that the following inequality

$$\left((1+b^2) - m(1+b^2)\right) \leq (1+b^2) \tag{18}$$

holds true. If and only if $m = 0$, the inequality becomes equality.Thus, the between-cluster similarity in the denominator of (12) has a smaller weight than that of (4).

This completes the proof of Theorem 1.

References

1. Barreto A, Araujo AA, Kremer S (2003) A taxonomy for spatiotemporal connectionist networks revisited: the unsupervised case. Neural Comput 15:1255–1320
2. Bezdek JC, Pal RN (1998) Some new indexes of cluster validity. IEEE Trans Pattern Recognit Mach Intell 28(3):301–315
3. Chen W, Feng G (2012) Spectral clustering: a semisupervised approach. Neurocomputing 77(1):229–242
4. Chen WY, Song Y, Bai H, Lin C-J, Chang E (2011) Parallel spectral clustering in distributed systems. IEEE Trans Pattern Recognit Mach Intell 33(3):568–586
5. Davies DL, Bouldin DW (1979) A cluster separation measure. IEEE Trans Pattern Recognit Mach Intell 1(4):224–227
6. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
7. CHQ Ding, X He, H Zha, M Gu, HD Simon (2001) A min-max cut algorithm for graph partitioning and data clustering. In: Proceedings of the first IEEE International Conference on Data Mining (ICDM), Washington. DC, USA, pp 107–114
8. Duda R, Hart P, Stork D (2000) Pattern classification. Wiley-Interscience, London
9. Friedman M (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J Am Stat Assoc 32:675–701
10. Graham DB, Allinson NM (1998) Characterizing virtual Eigensignatures for general purpose face recognition. Face Recognit: From Theory Appl, NATO ASI Ser F, Comput Syst Sci 163:446–456
11. Hagen L, Kahng AB (1992) New spectral methods for ratio cut partitioning and clustering. IEEE Trans Comput Aided Des Integr Circuits Syst 11(9):1074–1085
12. Lago-Fernández LF, Corbacho F (2010) Normality-based validation for crisp clustering. Pattern Recogn 43(3):782–795
13. Z Lu, M Carreira-Perpinan (2008) Constrained spectral clustering through affinity propagation. In: Proceedings of CVPR, Anchorage, Alaska, USA, pp 1–8
14. Lu H, Fu Z, Shu X (2014) Non-negative and sparse spectral clustering. Pattern Recogn 47(1):418–426
15. Luxburg U (2007) A tutorial on spectral clustering. Statistics and computing 17(4):395–416
16. Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 20:53–65
17. Serkar S, Soundararajan P (2000) Supervised learning of large perceptual organization: graph spectral partitioning and learning automata. IEEE Trans Pattern Anal Mach Intell 22(5):504–525
18. Shi J, Malik J (2000) Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell 22(8):888–905
19. Wacquet G, Caillault EP, Hamad D, Ĥebert P-A (2013) Constrained spectral embedding for k-way data clustering. Pattern Recogn Lett 34(9):1009–1017
20. X Wang, I Davidson (2010) Flexible constrained spectral clustering. In: The 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington DC, USA, pp 563–572
21. Wang S, Siskind J (2003) Image segmentation with ratio cut. IEEE Trans Pattern Anal Mach Intell 25(6):675–690
22. Wu Z, Leahy R (1993) An optimal graph theoretic approach to data clustering: theory and its application to image segmentation. IEEE Trans Pattern Anal Mach Intell 15(11):1101–1113
23. AY Ng, MI Jordan (2002) On spectral clustering: analysis and an algorithm. In: Advances in neural information processing systems. Vancouver, British Columbia, Canada, pp 849–856
24. L Zelnik-Manor, P Perona (2004) Self-tuning spectral clustering. In: Saul LK, Weiss Y, Bottou L (eds) The 18th annual conference on neural information processing systems, Vancouver, British Columbia, Canada, pp 1601–1608
25. HS Zou, WD Zhou, L Zhang, CL Wu, RC Liu, LC Jiao (2009) A new constrained spectral clustering for sar image segmentation. In: Proceedings 2009 2nd Asian-Pacific Conference on Synthetic Aperture Radar, Xian, China, pp 680–683