Spectral clustering with eigenvector selection based on entropy ranking

Feng Zhao *, Licheng Jiao, Hanqiang Liu, Xinbo Gao, Maoguo Gong

Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Institute of Intelligent Information Processing, Xidian University, Xi’an, PR China

Abstract

Ng–Jordan–Weiss (NJW) method is one of the most widely used spectral clustering algorithms. For a K clustering problem, this method partitions data using the largest K eigenvectors of the normalized affinity matrix derived from the dataset. It has been demonstrated that the spectral relaxation solution of K-way grouping is located on the subspace of the largest K eigenvectors. However, we find from a lot of experiments that the top K eigenvectors cannot always detect the structure of the data for real pattern recognition problems. So it is necessary to select eigenvectors for spectral clustering. We propose an eigenvector selection method based on entropy ranking for spectral clustering (ESBER). In this method, first all the eigenvectors are ranked according to their importance on clustering, and then a suitable eigenvector combination is obtained from the ranking list. In this paper, we propose two strategies to select eigenvectors in the ranking list of eigenvectors. One is directly adopting the first K eigenvectors in the ranking list. Different to the largest K eigenvectors of NJW method, these K eigenvectors are the most important eigenvectors among all the eigenvectors. The other eigenvector selection strategy is to search a suitable eigenvector combination among the first $K_m$ ($K_m > K$) eigenvectors in the ranking list. The eigenvector combination obtained by this strategy can reflect the structure of the original data and lead to a satisfying spectral clustering result. Furthermore, we also present computational complexity reduction strategies for ESBER method to deal with large-scale datasets. We have performed experiments on UCI benchmark datasets, MNIST handwritten digits datasets, and Brodatz texture datasets, adopting NJW method for a baseline comparison. The experimental results show that ESBER method is more robust than NJW method. Especially, ESBER method with the latter eigenvector selection strategy can obtain satisfying clustering results in most cases.

1. Introduction

Clustering approach has been one of the most important research topics in pattern recognition and machine learning. The purpose of clustering is to partition a dataset into expected clusters such as data points in the same cluster, which are similar but in different clusters are dissimilar. In the past few decades, many clustering algorithms have been developed [12,19], which mainly contain hierarchical (Such as Single Link, Complete Link, etc.) and partitional (Such as K-means, Gaussian Mixture, Density Estimation and Mode Seeking, etc.) clustering. As the datasets become larger and more varied, many of the dimensions are often irrelevant. These irrelevant dimensions can confuse traditional clustering algorithms. In order to overcome this issue, subspace learning algorithm is proposed to project an original high-dimensional space to a low-dimensional space, and obtain a new data representation preserving well statistical properties. Subspace learning contains many research fields, such as the extensions of Fisher’s linear discriminant analysis [27], manifold learning [25,28], spectral analysis [18,32], kernel machine [20], tensor machine [33], and so on.

Spectral analysis methods have been successfully used to solve data clustering and graph partitioning problems. In recent years, spectral clustering [11,22,23,26] has attracted more and more interest due to their high performance in data clustering and simplicity in implementation. This kind of method has been successfully applied to parallel computations [17], VLSI design [1], image segmentation [26], speech separation [3], etc. Spectral clustering methods utilize the eigenvectors of the normalized affinity matrix derived from data to perform data partitioning. NJW method [22] is one of the most widely used spectral clustering algorithms. For a K clustering problem, this method always partitions data using the largest K eigenvectors of the

1 The largest K eigenvectors are the eigenvectors corresponding to the K largest eigenvalues.
normalized affinity matrix of a dataset. Although the spectral relaxation solution of normalized cut criteria lies in the subspace spanned by these eigenvectors [16], it is not guaranteed that the largest \( K \) eigenvectors can well detect the structure of the data. Xiang and Gong [30] were the first to use eigenvector selection to improve the performance of the spectral clustering method. The approach firstly finds the largest \( Km \) eigenvectors of the normalized affinity matrix of the data, denoted as \( e_1, e_2, \ldots, e_{Km} \). Then the relevance of each eigenvectors \( R_0 \) is estimated according to how well it can separate data into different clusters. Eventually the method preserves all the relevant eigenvectors (i.e., \( R_0 \geq 0.5 \)) and utilizes these eigenvectors to perform data clustering. This method behaves well in image segmentation. However, the eigenvector combination consisting of all the relevant eigenvectors can not always represent the data structure.

In this paper, we propose a novel eigenvector selection method based on entropy ranking for spectral clustering (ESBER). First all the eigenvectors are ranked according to their importance on clustering and then a suitable eigenvector combination is obtained from the ranking list. We introduce two strategies to select eigenvectors in the ranking list. One is directly adopting the first \( K \) eigenvectors in the ranking list. ESBER method based on this direct eigenvector selection strategy is short for ESBER-D. Although this method adopts the most important \( K \) eigenvectors, it may not well detect the structure of the data. So the performance of this method is only a little better than NJW method. Because we believe that the eigenvector selection for spectral clustering should be considered as a combinatorial optimization problem, we propose another eigenvector selection strategy which is searching an optimal eigenvector combination among the first \( Km \) (\( Km > K \)) eigenvectors in the ranking list. Based on the assumption that a sampling of data can retain the original clustering information in most cases, this strategy first draws a training set from the original dataset, and then extracts the corresponding part of the training data in the first \( Km \) (\( Km > K \)) eigenvectors of the ranking list and uses eigenvalue eigenvector combination evaluation criteria to find a suitable eigenvector combination. This strategy is called indirect eigenvector selection strategy, and ESBER method based on this strategy is short for ESBER-I. When \( K \) is not very large (e.g., \( K \leq 10 \)), it is concluded from experiments that \( Km = 10 \) is enough to determine a well eigenvector combination for spectral clustering. When \( K \) is larger than 10, we can set \( Km \) as an appropriate value larger than \( K \) and adopt a combinatorial optimization algorithm, e.g., Genetic algorithm (GA) [15], to search a suitable eigenvector combination for spectral clustering. In order to expand ESBER method to large-scale datasets, we also present a computational complexity reduction strategy for ESBER method.

The rest of the paper is organized as follows. In Section 2, we first review NJW method [22], then introduce the construction of the affinity matrix; eventually discuss the necessity of eigenvector selection for spectral clustering. In Section 3, we present the novel eigenvector selection method based on entropy ranking in detail, including the eigenvector ranking based on entropy, the eigenvector selection based on the ranking list, the computational complexity and reduction strategies. Experimental results on real-world datasets are given in Section 4, comparing our method with NJW method. In this section, we also present the impact of the parameters of our method on the clustering performance, the expansibility analysis of our method on the large-scale datasets, and an extension of ESBER-I to further improve the clustering performance. Finally, some concluding remarks and several issues for future work are given in the last section.

2. Spectral clustering algorithm and related problems

2.1. Ng–Jordan–Weiss (NJW) method

Spectral clustering methods are widely used graph-based approaches for data clustering. Given a dataset \( X=\{x_1, x_2, \ldots, x_n\} \) in \( \mathbb{R}^d \) with \( K \) clusters, we can define a \( n \times n \) affinity matrix \( A \) whose element \( A_{ij} \) can be viewed as the weight on the edge connecting the \( i \)th and \( j \)th data points. The element \( A_{ij} \) of the affinity matrix is measured by a typical Gaussian function

\[
A_{ij} = \begin{cases} \exp(-d^2(x_i, x_j)/\sigma^2) & i \neq j \\ 0 & i = j \end{cases}
\]

Furthermore, the degree matrix \( D \) is a diagonal matrix whose element \( D_{ii} (D_{ii} = \sum_j A_{ij}) \) is the degree of the point \( x_i \).

In above framework, clustering problem can be seen as a graph partitioning problem. As a spectral approach to graph partitioning problem, NJW method [16] uses the normalized affinity matrix as the Laplacian matrix and solves the optimization of the normalized cut criterion [18] through considering the eigenvectors associated with the largest eigenvalues. The idea of NJW method is to find a new representation of patterns on the first \( K \) eigenvectors of the Laplacian matrix. The details of NJW method are given as follows.

1. Form the affinity matrix \( A \in \mathbb{R}^{n \times n} \) defined by Eq. (1).
2. Compute the degree matrix \( D \) and the normalized affinity matrix \( L = D^{-1/2}AD^{-1/2} \).
3. Let \( 1 = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_K \) be the \( K \) largest eigenvalues of \( L \) and \( v^1, v^2, \ldots, v^K \) be the corresponding eigenvectors. Form the matrix \( V = [v^1, v^2, \ldots, v^K] \in \mathbb{R}^{n \times K} \) and here \( v^i \) is the column vector.
4. Form the matrix \( Y \) from \( V \) by renormalizing each of \( s \) rows to have unit length (i.e., \( Y_q = V_q / (\sum_j V_{jq}^2)^{1/2} \)).
5. Treat each row of \( Y \) as a point in \( \mathbb{R}^K \), and cluster them into \( K \) clusters via K-Means algorithm to obtain the final clustering of original dataset.

The success of NJW method is mainly based on the fact that it does not make strong assumptions in the form of data points. As we know, when the data points do not correspond to convex region, traditional clustering algorithms, such as K-Means algorithm, would not find the satisfactory clustering. But NJW method can obtain satisfying clustering results through embedding the data points into a new space (\( Y \)s rows) in which clusters are tighter. Therefore, NJW method outperforms K-Means algorithm in most cases.

2.2. Affinity matrix construction

As shown in Eq. (1), Gaussian function is widely used to construct the affinity matrix for spectral clustering. The scale parameter \( \sigma \) controls the affinity between two data points, which needs to be manually set by users. For a dataset, especially for a dataset with multiple scales, a fixed \( \sigma \) cannot always well scale the affinity between any two data points. It is known that parameter \( \sigma \) greatly affects spectral clustering results [6,31]. The spectral clustering results under different values of \( \sigma \) are very different from one another. Zelnik–Manor and Perona [31] proposed to calculate a local scale parameter \( \sigma_i \) for each data point \( x_i \) instead of the fixed scale parameter \( \sigma \). The local scale parameter \( \sigma_i \) for data point \( x_i \) is calculated as follows:

\[
\sigma_i = d(x_i, x_{ip})
\]
where $x_i$ is the $l$th neighbor of point $x_i$ and $d(a,b)$ denotes the Euclidean distance between the points $a$ and $b$. Then the affinity between a pair of points can thus be written as

$$A_{ij} = \exp(-d^2(x_i,x_j)/\sigma_i \sigma_j)$$

where the parameter $l$ is assigned to 7 in our method.

Furthermore, when the size of a dataset is not large, the fully connected graph is adopted to construct the affinity matrix $A$. For large-scale datasets, the fully connected graph is infeasible due to the storage of the matrix $A$. So we adopt the $k$-neighbor graph to construct the affinity matrix $A$ for large-scale datasets in this paper.

### 2.3. Necessity of eigenvector selection

In this section, we take Three-circle and Two-moon datasets, presented in Fig. 1, as examples to investigate the eigenvectors of the normalized affinity matrix (The Gaussian function using the local scaling parameter is adopted to construct the affinity matrix). Some eigenvector combinations for these two datasets are presented in Fig. 2. For all figures presenting eigenvectors in this paper, elements of each eigenvector corresponding to different clusters are mark coded according to the ground-truth label information.

For the Three-circle dataset, NJW method adopts the largest 3 eigenvectors of the normalized affinity matrix to perform

![Fig. 1. Synthetic datasets: (a) Three-circle and (b) Two-moon.](image)

![Fig. 2. The eigenvector combinations for the Three-circle and Two-moon datasets: (a) the largest 3 eigenvectors of Three-circle; (b) the 2nd largest eigenvector of Three-circle; (c) the largest 2 eigenvectors of Two-moon and (d) the 2nd and 4th largest eigenvectors of Two-moon.](image)
where \( \max_k \) and \( \min_k \) denote the maximum and minimum of the \( k \)th eigenvector \( \nu_k \), respectively, so \( (\max_k-\min_k) \) is the maximum interval of the \( k \)th eigenvector.

According to the definition of \( S_{ij} \), the similarity \( S_{ij} \) between \( V_i \) and \( V_j \) is high if these two data points are very close, and low if these two points are far away. Entropy \( E_{ij} \) is low if \( S_{ij} \) is either low or high and \( E_{ij} \) is high otherwise. Therefore, if the removal of an eigenvector \( \nu^l \) causes more disorder than another eigenvector \( \nu^v \), the entropy values satisfy \( E_{ij} > E_{ij'} \) \( (E_{ij} = -\sum_{V_i \in \mathcal{V}} \sum_{V_j \in \mathcal{V}} E_{ij}) \) and the eigenvector \( \nu^v \) is more important than the eigenvector \( \nu^l \) for spectral clustering. In order to obtain the ranking list of the eigenvectors, each eigenvector is removed in turn and the corresponding entropy is calculated. The ranking list of eigenvectors is denoted by \( \{R_{V_1}, R_{V_2}, ..., R_{V_n}\} \), which is a descending order according to the entropy of eigenvectors. Taking a dataset \( X = \{x_1, x_2, x_3, x_4, x_5\} \) for example, the ranking list is \( \{4, 1, 3, 2, 5\} \) when the entropy values of five eigenvectors satisfy \( E_{x_4} > E_{x_3} > E_{x_2} > E_{x_1} > E_{x_5} \). So the 4th eigenvector is the most important one among five eigenvectors.

3.2. Ranking-based eigenvector selection

3.2.1. Strategies for eigenvector selection

After obtaining the ranking list of eigenvectors, a simple eigenvector selection method is directly adopting the first \( K \) eigenvectors in the ranking list to perform data clustering. Different from the largest \( K \) eigenvectors of NJW method, these \( K \) eigenvectors \( \nu^{v_1}, \nu^{v_2}, ..., \nu^{v_K} \) are the most \( K \) important eigenvectors measured by entropy among all the eigenvectors. This eigenvector selection strategy is called direct eigenvector selection strategy.

The other proposed eigenvector selection strategy is to search a suitable eigenvector combination according to the eigenvector ranking list. It is known that all the data points of a dataset can be considered as independently sampled, so a sampling of data can retain the original clustering information in most cases \([9,13]\). In real applications, sometimes it is possible to obtain some data points with ground-truth label information for a dataset. In this paper, we first draw a training dataset with ground-truth label information from the original data, and then extract the corresponding part of the training data in the first \( Km \) \( (Km > K) \) eigenvectors in the ranking list and adopt an eigenvector combination evaluation criteria to find a suitable eigenvector combination among all the possible eigenvector combinations. We hold the point that the subspace spanned by this eigenvector combination generated from the training data can reflect the underlying structure of the original data. This eigenvector selection strategy is called indirect eigenvector selection strategy. We assume that this strategy can find better eigenvector combinations for spectral clustering than directly using the first \( K \) eigenvectors in the ranking list.

Through amounts of experiments, we find a few eigenvectors which are crucial for data clustering. The first \( Km \) \( (K \leq Km \leq n) \) eigenvectors in the ranking list are considered as the most important \( Km \) eigenvectors for clustering. So we aim to obtain suitable eigenvector combinations among the first \( Km \) eigenvectors in the ranking list. When \( K \) is not very large (e.g., \( K \leq 10 \)), the number of the crucial eigenvectors for a dataset is much fewer, so we think that \( Km \) with an assignment of 10 is enough to find a good eigenvector combination for spectral clustering. We have verified in the experimental section that it is of no use assigning \( Km \) to a higher value when \( K \) is less than or equal to 10. In most cases, the clustering performance of a larger \( Km \) is not quite different from that of \( Km = 10 \). When \( K \) is larger than 10, we can...
set \( Km \) an appropriate value larger than \( K \) and adopt GA [15] to search a suitable eigenvector combination for spectral clustering.

3.2.2. Indirect eigenvector selection for spectral clustering

The goal of the indirect eigenvector selection is to determine a suitable eigenvector combination to achieve satisfying clustering performance. Therefore, the evaluation criteria should assess the validity of eigenvector combination for clustering. Given a set of eigenvector combinations \( \text{Eigen}_\text{com} = \{e_1, e_2, \ldots, e_m\} \), in which \( e_i (1 \leq i \leq m) \) is an eigenvector combination, the evaluation criteria is defined as

\[
f(e_i) = 10^4 \text{Acc}_{\text{train}} + \lambda \text{len}_{\text{remain}}
\]

where \( \lambda = 100/Km \), \( \text{Acc}_{\text{train}} \) is the clustering accuracy of the training data using the eigenvector combination \( e_i \), and \( \text{len}_{\text{remain}} \) corresponds to the number of eigenvectors not selected. The evaluation criterion aims to maximize the clustering accuracy of the training data while select fewer eigenvectors. The \( \text{Acc}_{\text{train}} \) term ranges from 0 to 1, thus, the first term in Eq. (6) assumes values from 0 to 10,000. The \( \text{len}_{\text{remain}} \) term ranges from 0 to \( Km-1 \), thus, the second term in Eq. (6) assumes values below 100. Based on the weights \( 10^4 \) and \( \lambda \), the \( \text{Acc}_{\text{train}} \) term dominates the evaluation criterion value. The indirect eigenvector selection for spectral clustering is performed by the following algorithm.

Algorithm 1: Indirect eigenvector selection using exhaustive search

**Input:** The training dataset from the original dataset \( X = \{x_1, x_2, \ldots, x_n\} \); the matrix \( V \in \mathbb{R}^{n \times Km} \) consisting of the first \( Km \) eigenvectors in the ranking list of all the eigenvectors.

**Output:** The optimal eigenvector combination \( e^* \).

1. Extract the corresponding part of the training data in \( V \) and denote it by \( V_{\text{train}} \in \mathbb{R}^{t \times Km} \), where \( t n \) is the number of the training data \( i = 1 \).
2. Generate the set of eigenvector combinations \( \text{Eigen}_\text{com} = \{e_1, e_2, \ldots, e_m\} \) from the first \( Km \) eigenvectors in the ranking list.
3. Form the matrix \( U_{\text{train}} \in \mathbb{R}^{t \times Km} \) from \( V_{\text{train}} \) by only adopting the eigenvectors in the \( e_i \), where \( I \) is the number of the eigenvectors in the \( e_i \).
4. Form the matrix \( Y_{\text{train}} \) from \( U_{\text{train}} \) by renormalizing each of \( U_{\text{train}} \)'s rows to have unit length.
5. Treat each row of \( Y_{\text{train}} \) as a point in \( I \), then cluster them into \( I \) clusters via K-Harmonic Means algorithm, eventually obtain the clustering result of the training data.
6. Adopt the clustering accuracy to measure the clustering result of the training data, denoted as \( \text{Acc}_{\text{train}} \). Then compute the evaluation criteria value for \( e_i \) using Eq. (6).
7. \( i = i + 1 \) if \( i \leq m \), go to step (3), otherwise, conclude this algorithm and output the eigenvector combination \( e^* \) when \( e^* = \arg \max_{e_i, 1 \leq i \leq m} f(e_i) \).

When \( K \) is not very large (e.g., \( K \leq 10 \)), \( Km = 10 \) is enough to determine suitable eigenvector combinations for spectral clustering. The number of all the possible eigenvector combinations \( m \) is \( 2^{Km-1} = 1023 \) (we assume that an eigenvector combination contains an eigenvector at least). So it is feasible to evaluate each eigenvector combination for spectral clustering. When \( K \) is larger than \( 10, Km \) should be set by an appropriate value larger than \( K \) and the complexity of Algorithm 1 is very high. In this case, we adopt GA [15] to search a suitable eigenvector combination among the first \( Km \) eigenvectors in the ranking list. The indirect eigenvector selection using GA is presented in Algorithm 2 in the Appendix A.

3.2.3. Design procedure for our method

Based on the above descriptions, the procedure for eigenvector selection based on entropy ranking is given as follows:

Algorithm 3: Eigenvector selection based on entropy ranking (ESBER)

**Input:** A dataset \( X = \{x_1, x_2, \ldots, x_n\} \)

**Output:** The optimal eigenvector combination \( e^* \) for spectral clustering.

1. Form the affinity matrix \( A \in \mathbb{R}^{n \times n} \) and assign the diagonal elements \( A_{ii} = 0, 1 \leq i \leq n \).
2. Compute the degree matrix \( D \) and the normalized affinity matrix \( L = D^{-1/2}AD^{-1/2} \).
3. Obtain all the eigenvectors of \( L = \{v^1, v^2, \ldots, v^n\} \). Then all the eigenvectors are ranked according to their entropy values, as described in Section 3.1. The ranking list of eigenvectors is \( \{Rv_1, Rv_2, \ldots, Rv_n\} \).
4. If the direct eigenvector selection strategy is adopted, we utilize the first \( K \) eigenvectors in the ranking list to form the matrix \( U = \{v^{Rv_1}, v^{Rv_2}, \ldots, v^{Rv_{Km}}\} \in \mathbb{R}^{n \times Km} \) and then go to step (8). Otherwise, go to step (5).
5. Generate a training dataset from the original dataset \( X \).
6. If \( K \leq 10 \), set \( Km = 10 \) and adopt Algorithm 1 to obtain the optimal eigenvector combination \( e^* \). Otherwise, set \( Km \) an appropriate value larger than \( K \) and use the Algorithm 2 to obtain the optimal eigenvector combination \( e^* \).
7. The matrix \( U \in \mathbb{R}^{n \times Km} \) is formed by all the eigenvectors in the \( e^* \), where \( I \) is the number of eigenvectors in the \( e^* \).
8. Form the matrix \( Y \) from \( U \) by renormalizing each of \( s \) rows to have unit length (i.e., \( Y_{j} = \sum_{i} U_{ij} / \sqrt{\sum_{i} U_{ij}^2} \)).
9. Treat each row of \( Y \) as a point and cluster them into \( K \) clusters via K-Harmonic Means algorithm.
10. Assign the original point \( x_i \) to cluster \( c_i \) if and only if the corresponding row \( i \) of the matrix \( Y \) was assigned to cluster \( c_i \).

3.3. Complexity analysis and reduction strategies

3.3.1. Complexity analysis

The complexity of our method mainly consists of three parts. The first part is obtaining all the eigenvectors of the normalized affinity matrix \( L \). Its complexity is \( O(n^3) \). The second part is ranking all the eigenvectors. Its complexity is also \( O(n^3) \). The third part of complexity of our method is the eigenvector selection. There are two conditions to be considered. When the direct eigenvector selection strategy is adopted, that is, the first \( K \) eigenvectors in the ranking list of eigenvectors are selected, this complexity can be ignored. In the case of using the indirect eigenvector selection strategy, there are \( 2^{Km-1} \) eigenvector combinations to be evaluated and the complexity is \( O(2^{Km}) \). When \( K \) is not very large (e.g., \( K \leq 10 \)), \( Km = 10 \) is enough to find a satisfying eigenvector combination for our method. In this condition, the number of all the possible eigenvector combinations is 1023. When \( K \) is larger than \( 10, Km \) is set by an appropriate value larger than \( K \) and GA [15] is adopted to search a suitable eigenvector combination. In this condition, the \( O(2^{Km}) \) complexity is replaced by the complexity of GA. The parameters used in GA are given as follows: \( Go \) is the evolutionary generation number, \( ps \) the population size, \( Prob_m \) and \( Prob_c \) are the mutation probability and crossover probability, respectively, the complexity of GA is \( Ga(Ops + ps \times Proc_m + ps \times Proc_c) + Ops \times \log_2(ps) \), which is given in Appendix A. In a word, when \( Km = 10 \), the computational complexity of ESBER_I method (i.e.,
ESBER method based on the indirect eigenvector selection strategy is $O(n^2)+O(n^2)+O(1023)=O(2n^2+1023)$. When $Km$ is larger than 10, the computational complexity of ESBER-I method becomes $O(2n^2)+Ga(O(ps+ps \times Proc_m+ps \times Proc_c)+O(ps \times \log_2 ps))$.

3.3.2. Complexity reduction

Based on amounts of experiments, we find it unnecessary to solve and rank all the eigenvectors of the normalized affinity matrix $L$. Besides, it is also infeasible to solve and rank all the eigenvectors for large-scale datasets. Here, we first obtain the largest $kn(Km \leq kn \leq n)$ eigenvectors and then rank them according to their entropy values. In the experiment part, we have demonstrated that the first $Km$ eigenvectors in the ranking list of $n$ eigenvectors are almost the same to the corresponding part in the ranking list of $kn$ eigenvectors. In this condition, the complexity of eigen-decomposition and eigenvector ranking both become $O(\log2(\frac{n}{kn} \times q^2))$.

Furthermore, we introduce scalable ranking method [9] to further reduce the ranking complexity, which was used to rank features of data. Firstly, $kn$ eigenvectors are ranked 0. Secondly, random samplings are generated (Random samplings are performed on the data consisting of $kn$ eigenvectors, not the original data). Thirdly, ranking is run over each sampling to produce the ranking of eigenvectors, in which $kn$ eigenvectors are ranked 1, 2, ..., $kn$ according to their importance measured by entropy. Finally, the eigenvector rankings are added correspondingly, and the ranking list of the $kn$ eigenvectors is obtained by sorting the sum of eigenvector rankings. When the sampling number is $p$ and the data number of each sampling is $q$, the ranking complexity is $O(p \times kn \times q^2)$.

Using the above two complexity reduction solutions can greatly reduce the complexity of our method. When $Km=10$, the computational complexity of ESBER-I method becomes $O(\log2(\frac{n}{kn} \times q^2))+O(p \times kn \times q^2)+O(1023)$. When $Km$ is larger than 10, the computational complexity of ESBER-I method is $O(\log2(\frac{p}{kn} \times q^2))+O(p \times kn \times q^2)+Ga(O(ps+ps \times Proc_m+ps \times Proc_c)+O(ps \times \log_2 ps))$.

4. Experimental results and analysis

4.1. Experimental settings

In this paper, ESBER method is testified on a number of datasets from UCI Machine Learning Repository [2], MNIST handwritten digits database and Brodatz's texture album [4], respectively. NJW method is performed for a baseline comparison. For the datasets from the first two databases, Eq. (3) is adopted to compute the affinity matrix, as introduced in Section 2.2. For the datasets from Brodatz's texture album, the local scale parameters in Eq. (3) fail to measure well the affinity between any two data points, so we adopt Eq. (4) to compute the affinity matrix and the scale parameter $\sigma$ is set to 3. K-Harmonic Means algorithm is used as the final clustering algorithm in both NJW and ESBER, and the maximal iteration number of this algorithm is set to 500 in this paper. When the indirect eigenvector selection strategy is adopted, ESBER method needs the training data to guide the eigenvector selection. For all the datasets, we adopt 10% of the data (Draw 10% from every class of the data) as the training data. It is certain that the eigenvectors selected by our method under different training data are not identical, so we perform 10 independent runs on each dataset. We adopt the clustering accuracy [29] to evaluate the clustering quality, which is also used to compute $Acc_{\text{train}}$ in Eq. (6). The clustering accuracy is defined as

$$\text{Acc} = \frac{1}{n} \sum_{i=1}^{n} \delta(y_i, \text{map}(c_i))$$

4.2. Eigenvector selection experiments

4.2.1. Experiments on UCI benchmark datasets

We first perform experiments on four UCI benchmark datasets [2]. Clustering results are presented in Table 1. Acc and $\text{Acc}_d$ denote the clustering accuracy of NJW and ESBER-D methods, respectively. Although ESBER-I method adopts 10% data points with ground-truth label information from the original dataset as the training data to determine an eigenvector combination, they are not used to cluster the original data. So we still compute the clustering accuracy of the original dataset for ESBER-I method. $\text{Acc}_\text{min}$, $\text{Acc}_\text{max}$, and $\text{Acc}_\text{ave}$ denote the minimum, maximum and average clustering accuracy of ESBER-I method for 10 independent runs, respectively.

Furthermore, in order to make a fair comparison with NJW method to some degree, we design a new statistical term $\text{Acc}_1$ for NJW method. The 10% data points with ground-truth label information, which have been used in ESBER-I method, are denoted as $\text{XS} = \{x_1, x_2, \ldots, x_p\}$ ($3 = \lceil n/10 \rceil, 1 \leq j \leq n, 1 \leq r \leq s$) with the ground-truth label $Y = \{y_1, y_2, \ldots, y_p\}$ ($y_r \in \{1, 2, \ldots, K\}$). When the cluster label obtained by NJW method is $C = \{c_1, c_2, \ldots, c_n\}$ ($C \in \{1, 2, \ldots, K\}$), we modify the label $C$ with

$$C(i) = \begin{cases} y_i & x_i \in \text{XS} \\ C(i) & \text{else} \end{cases}$$

and compute the corresponding clustering accuracy, denoted as $\text{Acc}_1$.

For the Iris and Wine datasets, the maximum accuracy of ESBER-I method is higher than the Acc and $\text{Acc}_1$ values of NJW method, but the average and minimum accuracy values are lower than the Acc and $\text{Acc}_1$ values of NJW method. For the Ionosphere and Breast-w datasets, the minimum, maximum and average accuracy values of ESBER-I method are all higher than the Acc and $\text{Acc}_1$ values of NJW method. As referred before, the eigenvector combinations of ESBER-I method under different training data are
different from one another, so are the clustering accuracy values. For the four UCI benchmark datasets, Fig. 3(a) presents the clustering accuracy of ESBER_I method for 10 independent runs under different training data. From Fig. 3(a), we can find that most of the 10 runs of ESBER_I method can obtain satisfying clustering results. Take the Iris dataset for example, 7 of 10 runs can obtain the accuracy of 0.9400, and the clustering accuracy of the other 3 runs are 0.8333, 0.8933 and 0.7933, which cause the average and minimum accuracy of ESBER_I method to be lower than the Acc and Acc_min of NJW method. For the Wine dataset, only one among 10 runs obtains a bad clustering result and causes the average and minimum accuracy of ESBER_I method to be lower than the Acc and Acc_min of NJW method. For the Ionosphere and Breast-w datasets, only one among 10 runs obtains an apparently lower clustering accuracy compared with the other runs, but the accuracy obtained by this run of ESBER_I method is still higher than the Acc and Acc_min of NJW method.

Here, we discuss and analyze the eigenvector combination selected by ESBER_I method. We take the Iris dataset for example to display the distribution of eigenvectors. NJW method utilizes the largest 3 eigenvectors to perform clustering and Fig. 4(a) shows the data projection on these three eigenvectors. For ESBER_I method, the eigenvector combinations of 10 independent runs all include the 2nd and 3rd largest eigenvectors (See Fig. 4(b) and (c)), which are the first two eigenvectors in the ranking list of eigenvectors (See Table 3). The eigenvector combination composed of the 2nd and 3rd largest eigenvectors, presented in Fig. 4(d), occurs 7 times with the clustering accuracy of 0.9400. For the Wine dataset, 10 eigenvector combinations all contain the 2nd largest eigenvector, and 6 eigenvector combinations include the 3rd largest eigenvector. These two eigenvectors are the first two eigenvectors of the ranking list (See Table 3). The eigenvector combination constituted by 2nd, 3rd and 5th eigenvectors occurs 5 times with a clustering accuracy of 0.9693. Because NJW method utilizes the largest 2 eigenvectors for clustering, without using the 3rd largest eigenvector, its performance on this dataset is not satisfying. From this experiment, we can verify that it is not always reasonable for NJW method to utilize the largest $K$ eigenvectors for clustering. Though ESBER_D method adopts the first $K$ eigenvectors in the ranking list to perform clustering, it shares the same problem of NJW method. As shown in Table 1, the accuracy of ESBER_D is lower than the Acc and Acc_min of NJW method for the Iris and Wine datasets, and higher than the Acc of NJW method. From the experimental results, we find that ESBER_I can provide more reasonable eigenvector combinations for spectral clustering than the other methods.

4.2.2. Experiments on MNIST handwritten digit datasets

We further perform experiments on the MNIST handwritten digits database which consists of 10 classes (0–9). There are 60,000 training examples and 10,000 test examples which have been size normalized and centered to $28 \times 28$ gray-level images. We use the subsets {5, 8}, {8, 9}, {0, 3, 8} and {1, 2, 3, 4} to test our method. They are randomly extracted from the test examples and the number of each digit is 200. The clustering results are presented in Table 2.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Examples</th>
<th>Attributes</th>
<th>Classes</th>
<th>NJW Acc</th>
<th>ESBER Acc</th>
<th>ESBER Acc_d</th>
<th>ESBER Acc_min</th>
<th>ESBER Acc_max</th>
<th>ESBER Acc_ave</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>150</td>
<td>4</td>
<td>3</td>
<td>0.9267</td>
<td>0.9327</td>
<td>0.8067</td>
<td>0.7933</td>
<td>0.9400</td>
<td>0.9100</td>
</tr>
<tr>
<td>Wine</td>
<td>178</td>
<td>13</td>
<td>3</td>
<td>0.9663</td>
<td>0.9702</td>
<td>0.9382</td>
<td>0.6742</td>
<td>0.9775</td>
<td>0.9292</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>351</td>
<td>34</td>
<td>2</td>
<td>0.7322</td>
<td>0.7573</td>
<td>0.7521</td>
<td>0.7806</td>
<td>0.8889</td>
<td>0.8467</td>
</tr>
<tr>
<td>Breast-w</td>
<td>683</td>
<td>9</td>
<td>2</td>
<td>0.7189</td>
<td>0.7463</td>
<td>0.9444</td>
<td>0.8961</td>
<td>0.9652</td>
<td>0.9594</td>
</tr>
</tbody>
</table>

**Fig. 3.** The clustering accuracy values of ESBER_I method for 10 independent runs: (a) UCI benchmark datasets and (b) MNIST handwritten digits datasets.
For the {5, 8}, {0, 3, 8} and {1, 2, 3, 4} datasets, the minimum, maximum and average accuracy values of ESBER_I method are higher than the Acc and Acc_l values of NJW method. For the {8, 9} dataset, the maximum accuracy of ESBER_I method is higher than the Acc and Acc_l of NJW method, and the minimum accuracy of ESBER_I method is lower than the Acc and Acc_l of NJW method. The average accuracy of ESBER_I method is a little lower than the Acc_l of NJW method, but higher than the Acc of NJW method. Except for the {5, 8} dataset, the accuracy of ESBER_D method is higher than the Acc and Acc_l of NJW method. For the {5, 8} dataset, the accuracy of ESBER_D method is lower than the Acc and Acc_l of NJW method. For the {5, 8} and {0, 3, 8} datasets, there are 3 and 1 among 10 runs obtaining apparently lower clustering accuracy compared with the other runs, respectively, but these runs still obtain higher clustering accuracy than NJW method. For the {8, 9} dataset, the clustering accuracy values of 10 runs of ESBER_I method are not quite different from one another. The same is true with the {0, 3, 8} dataset.

The first 10 eigenvectors in the ranking list of the {5, 8} dataset are the 2nd, 3rd, 4th, 7th, 5th, 17th, 9th, 18th, 30th and 124th eigenvectors. For ESBER_I method, the eigenvector combinations of 10 runs are all including the 3rd and 4th largest eigenvectors and the eigenvector combination composed of the 3rd and 4th largest eigenvectors occur 6 times with accuracy of 0.9325. The distribution of eigenvectors for this dataset is presented in Fig. 5.

### Table 2

Performance comparison on MNIST handwritten digits datasets.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Examples</th>
<th>Attributes</th>
<th>Classes</th>
<th>NJW</th>
<th>ESBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Acc</td>
<td>Acc_l</td>
<td>Acc_d</td>
<td>Acc_min</td>
<td>Acc_max</td>
</tr>
<tr>
<td>{5, 8}</td>
<td>0.5750</td>
<td>0.6158</td>
<td>0.54</td>
<td>0.7550</td>
<td>0.9325</td>
</tr>
<tr>
<td>{8, 9}</td>
<td>0.8975</td>
<td>0.9083</td>
<td>0.93</td>
<td>0.8750</td>
<td>0.9275</td>
</tr>
<tr>
<td>{0, 3, 8}</td>
<td>0.5467</td>
<td>0.5917</td>
<td>0.6733</td>
<td>0.8683</td>
<td>0.8767</td>
</tr>
<tr>
<td>{1, 2, 3, 4}</td>
<td>0.6713</td>
<td>0.6896</td>
<td>0.8325</td>
<td>0.7800</td>
<td>0.8762</td>
</tr>
</tbody>
</table>

For the {5, 8}, {0, 3, 8} and {1, 2, 3, 4} datasets, the minimum, maximum and average accuracy values of ESBER_I method are higher than the Acc and Acc_l values of NJW method. For the {8, 9} dataset, the clustering accuracy values of 10 runs of ESBER_I method are not quite different from one another. The same is true with the {0, 3, 8} dataset.

The first 10 eigenvectors in the ranking list of the {5, 8} dataset are the 2nd, 3rd, 4th, 7th, 5th, 17th, 9th, 18th, 30th and 124th eigenvectors. For ESBER_I method, the eigenvector combinations of 10 runs are all including the 3rd and 4th largest eigenvectors and the eigenvector combination composed of the 3rd and 4th largest eigenvectors occur 6 times with accuracy of 0.9325. The distribution of eigenvectors for this dataset is presented in Fig. 5. NJW method utilizes the largest 2 eigenvectors to perform clustering and the data projection on these two eigenvectors is presented in Fig. 5(a). We can find that the data projection is unseparated between two classes, so the Acc and Acc_l values of NJW method are very low. While the dataset is mapped to the 3rd

Fig. 4. The distribution of eigenvectors for the Iris dataset: (a) the largest 3 eigenvectors; (b) the 2nd largest eigenvector; (c) the 3rd largest eigenvector and (d) the 2nd and 3rd largest eigenvectors.
and 4th largest eigenvectors, the data overlapping between two classes is much alleviated (See Fig. 5(d)). For the (8, 9) dataset, the first 10 eigenvectors in the ranking list are the 2nd, 4th, 3rd, 5th, 1st, 17th, 9th, 8th, 51st and 6th eigenvectors. The eigenvector combinations of 10 runs all include the 2nd largest eigenvector and provide very close clustering accuracy values. NJW method utilizes the largest 2 eigenvectors to perform data clustering and the Acc and Acc_l values of this method are very close to the average accuracy of ESBER_I method. The first 10 eigenvectors in the ranking list for the (0, 3, 8) dataset are the 2nd, 11th, 1st, 6th, 7th, 4th, 32nd, 12th, 3rd and 8th eigenvectors. The eigenvector combinations of 10 independent runs all contain the 2nd and 4th largest eigenvectors and offer almost the same clustering accuracy values, which are much higher than the Acc and Acc_l values of NJW method. Because NJW method utilizes the largest 3 eigenvectors to perform clustering, without using the 4th largest eigenvector, the Acc and Acc_l values are very low. For the (1, 2, 3, 4) dataset, the first 10 eigenvectors in the ranking list are 2nd, 1st, 5th, 7th, 3rd, 13th, 73rd, 38th, 27th and 16th eigenvectors. All the eigenvector combinations of 10 runs are including the 2nd and 3rd largest eigenvectors. Although NJW method also adopts the 2nd and 3rd largest eigenvectors, it also includes the 4th largest eigenvector, which is not an important eigenvector and does not appear in the first 10 eigenvectors in the ranking list of eigenvectors. So the performance of NJW method is very poor for this dataset. The above four MNIST handwritten digits datasets again demonstrate that ESBER_I method can find more reasonable eigenvector combinations than ESBER_D and NJW methods.

### 4.2.3. Robustness analysis

In this section, we use the robustness analysis method proposed by Geng et al. [14] to evaluate the robustness of our method. The relative performance of the algorithm \( \alpha \) on a dataset is represented by the ratio \( R_\alpha \) of its clustering accuracy to the highest clustering accuracy among all the compared methods.

\[
R_\alpha = \frac{\text{Acc}_\alpha}{\max_j \text{Acc}_j}
\]  

(8)

So the best method \( \alpha^* \) on that dataset has \( R_{\alpha^*} = 1 \), and all the other methods have \( R_\alpha \leq 1 \). The larger the value of \( R_\alpha \), the better the performance of the method \( \alpha \) on that dataset. Thus the sum of \( R_\alpha \) over all datasets offers a good measure of the robustness of the method \( \alpha \). A large value of the sum indicates good robustness. For ESBER_I method, we adopt the average clustering accuracy \( \text{Acc}_\text{ave} \) of 10 independent runs as \( \text{Acc}_\alpha \) in Eq. (8) to compute \( R_\alpha \). 

Fig. 6 presents the distribution of \( R_\alpha \) of four methods (including NJW, ESBER_D, ESBER_I methods, and NJW method based on the accuracy term Acc_l, called NJW_L) over the eight datasets in Section 4.2. For each method, the eight values of \( R_\alpha \) are stacked, and the sum is given on top of the stack. It is shown from Fig. 6 that ESBER_I method has the highest sum value. Actually, the \( R_\alpha \) values of ESBER_I method on the eight datasets are 0.9749, 0.9577, 1, 1, 0.9720, 1 and 1, respectively, which indicate that ESBER_I method performs well on each dataset. Therefore ESBER_I method is the most robust method among all the methods.
4.3. Effect of parameters on the clustering performance

4.3.1. Percentage of training data

In the former experiments, for each dataset we adopt 10% of data for training to achieve a suitable eigenvector combination. In this section, we take the UCI benchmark datasets for example to inspect the effect of the percentage of training data on the algorithm performance. The percentage varies from 5% to 90%, and we perform 10 independent runs under each percentage. For these four UCI benchmark datasets, four curves of clustering accuracy with the percentage of training data are shown in Fig. 7. For each dataset, the minimum, average and maximum clustering accuracy values of 10 runs are presented. Fig. 7 shows that the maximum clustering accuracy values are almost invariant with the increasing percentage of training data for these four datasets. It can also be seen from Fig. 7 that these four curves of average clustering accuracy present an apparent increase from 5% to 10%, and the average clustering accuracy presents a slow and fluctuant ascending trend from 10% to 90%. So 10% of data is adopted for training to guide ESBER_I method to determine suitable eigenvector combinations for all the datasets used in Section 4.2.

4.3.2. Sensitivity analysis of Km

For all the datasets used in the above experiments, Km is set to 10 due to the cluster number of data K less than 10. In this section, Km is assigned to 20 and GA [15] is adopted to find suitable eigenvector combinations among the first 20 eigenvectors in the ranking list. We also take the UCI benchmark datasets for example and adopt 10% of data as training data. For all the datasets, the evolutionary generation, population size, mutation probability and crossover probability of GA are 200, 100, 0.1 and 0.8, respectively. For each dataset, we perform 10 independent runs and the average clustering accuracy of 10 independent runs is presented in Fig. 8. We can find that the average clustering accuracy of Km=20 is very close to the one of Km=10. So it is unnecessary to adopt a higher value of Km(Km > 10) in ESBER_I method when the cluster number of a dataset K ≤ 10. When K ≥ 10, Km=10 is enough to find a satisfying eigenvector combination for ESBER_I method.

4.3.3. Number of eigenvectors ranked

For all the datasets in Section 4.2, all the eigenvectors of the normalized affinity matrix L of data are obtained and ranked, and the first Km eigenvectors in the ranking list are adopted to obtain...
suitable eigenvector combinations for spectral clustering. In this section, we take the UCI benchmark datasets for example to validate whether it is necessary to solve and rank all the eigenvectors of the matrix $L$ or not. About 10% of each dataset is drawn as training data and $K_m$ is assigned to 10. Table 3 demonstrates the first 10 eigenvectors in the ranking list under different number of eigenvectors ranked (denoted as $k_n$). In Table 3, Bold is adopted to emphasize the eigenvectors, which are different from the first 10 eigenvectors in the ranking list of all the eigenvectors ($k_n=n$). For each UCI benchmark dataset, we find that there are almost the same first 10 eigenvectors in the ranking list under different $k_n$ values, and the number of different eigenvectors is not larger than 3. These different eigenvectors are always the unimportant ones, which are always the latter ones in the first 10 eigenvectors of the ranking list. For these four UCI benchmark datasets, $k_n=50$ can lead to a satisfying ranking list, in which the first 10 eigenvectors are very close to those in the ranking list of $n$ eigenvectors.

### 4.4. Expansibility analysis on large-scale datasets

For large-scale datasets, we adopt the complexity reduction strategies introduced in Section 3.3.2 to decrease the complexity of our method. In this section, the step (3) of Algorithm 3 is replaced by the following two operations. Firstly, the largest $k_n$ eigenvectors of the normalized affinity matrix $L$ are computed, and then the scalable ranking method is adopted to rank the $k_n$ eigenvectors.

#### 4.4.1. Experiments on large MNIST handwritten digits datasets

For all the datasets in the above experiments, the data number of each dataset is less than 1000. In this section, we generate eight large digit subsets with 2000–10,000 data points from the test dataset of MNIST handwritten digits database. Because the data number of each subset is large, so we adopt the $k$-neighbor graph to substitute the fully connected graph, which is used in the above experiments. We adopt 10% of the data as training data. $k_n$, $k_m$ and $K_m$ is assigned to 10, 50 and 10. For the scalable ranking method, the number of sampling is 10 and the data number of each sampling is $0.2n$, that is, each sampling contains 20% of the data points. For each dataset, we perform 10 independent runs and the clustering result is presented in Table 4.

Table 4 reveals that except the $\{3, 5, 8\}$, $\{0, 2, 3, 4, 5, 7, 8, 9\}$ and $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ datasets, ESBER_I method outperforms NJW method. For the $\{3, 5, 8\}$ dataset, the maximum and average clustering accuracy of ESBER_I method are higher than the Acc and Acc_l of NJW method, and the minimum clustering accuracy of ESBER_I method is lower than the Acc and Acc_l of NJW method. For the $\{0, 2, 3, 4, 5, 7, 8, 9\}$ dataset, the minimum, maximum and average clustering accuracy of ESBER_I method are

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Examples</th>
<th>Attributes</th>
<th>Classes</th>
<th>NJW</th>
<th>ESBER</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Acc</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Acc_l</td>
<td>Acc_d</td>
</tr>
<tr>
<td>$(3, 5, 8)$</td>
<td>2876</td>
<td>784</td>
<td>3</td>
<td>0.8140</td>
<td>0.8327</td>
</tr>
<tr>
<td>$(1, 2, 3, 4)$</td>
<td>4159</td>
<td>784</td>
<td>4</td>
<td>0.8959</td>
<td>0.9065</td>
</tr>
<tr>
<td>$(0, 1, 3, 5, 7)$</td>
<td>5045</td>
<td>784</td>
<td>5</td>
<td>0.7140</td>
<td>0.7423</td>
</tr>
<tr>
<td>$(0, 3, 4, 6, 8, 9)$</td>
<td>5913</td>
<td>784</td>
<td>6</td>
<td>0.7938</td>
<td>0.8144</td>
</tr>
<tr>
<td>$(2, 3, 5, 6, 7, 8, 9)$</td>
<td>6903</td>
<td>784</td>
<td>7</td>
<td>0.7629</td>
<td>0.8172</td>
</tr>
<tr>
<td>$(0, 2, 3, 4, 5, 7, 8, 9)$</td>
<td>7907</td>
<td>784</td>
<td>8</td>
<td>0.8633</td>
<td>0.7835</td>
</tr>
<tr>
<td>$(1, 2, 3, 4, 5, 6, 7, 8, 9)$</td>
<td>9020</td>
<td>784</td>
<td>9</td>
<td>0.7169</td>
<td>0.7451</td>
</tr>
<tr>
<td>$(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)$</td>
<td>10000</td>
<td>784</td>
<td>10</td>
<td>0.7406</td>
<td>0.7671</td>
</tr>
</tbody>
</table>
lower than the Acc_I value of NJW method, but higher than the Acc value of NJW method. For the {1, 2, 3, 4, 5, 6, 7, 8, 9} dataset, the maximum and average clustering accuracy of ESBER_I method are higher than the Acc and Acc_I of NJW method, but the minimum clustering accuracy of ESBER_I method is lower than the Acc_I value of NJW method and higher than the Acc value of NJW method. For the first five datasets, ESBER_D outperforms NJW method. For the other three datasets, NJW outperforms ESBER_D method. We can conclude from these experiments that ESBER method, especially ESBER_I method, can be successfully applied to large-scale datasets.

4.4.2. Experiments on Brodatz texture datasets

In this section, we testify that our method can perform on datasets with more than 10,000 datasets and with the cluster no. K > 10. We use 16 textures (D6, D9, D19, D20, D21, D24, D29, D53, D55, D57, D78, D80, D83, D84, D85, D92) obtained from Brodatz’s texture album [4], as shown in Fig. 9. Each image is of size 512 × 512 pixels with 256 gray levels. A dataset of n (n = 10,000, 20,000,...,100,000) image regions of 16 texture classes is constructed by dividing each 512 × 512 image into n/16 nonoverlapping 128 × 128 regions. These ten datasets are called Texture1, Texture2, Texture3, Texture4, Texture5, Texture6, Texture7, Texture8, Texture9 and Texture10. Due to computational simplicity and efficiency, stationary wavelet [21] is used to extract texture features. For each image region, we perform 3-level stationary wavelet decomposition and compute the subband energies using Eq. (9) to produce the feature vector that describes the texture. Therefore, we can obtain ten energy features of each 128 × 128 image

\[
EF_i = \frac{1}{\#R_i} \sum_{(k,l) \in R_i} |c(k,l)|
\]

where \( \#R \) denotes the number of coefficients in the subband \( R_i \).

Due to the cluster number \( K \) of these datasets more than 10, Algorithm 2 is used to determine the optimal eigenvector combination for the indirect eigenvector selection strategy. For all the datasets, \( Km \) and \( Kn \) are assigned to 30 and 100, respectively. The evolutionary generation, population size, mutation probability and crossover probability of GA are 200, 100, 0.1 and 0.8, respectively. Other experimental settings are similar to those of Section 4.4.1. The average clustering accuracy values of 10 runs of NJW, NJW_L, ESBER_D and ESBER_I methods are shown in Fig. 10. For the Texture1, Texture2, Texture6, Texture8 and Texture9 datasets, NJW method obtains the worst clustering result among these four methods. For the other datasets, ESBER_D method achieves the worst clustering performance. Moreover, the average accuracy value of NJW_L method is higher than that of NJW method for all these ten datasets. Although the accuracy of NJW_L method is slightly higher than that of ESBER_I method for the Texture2 and Texture5 datasets, ESBER_I method obtains the best performance among these four methods for the other eight datasets. From these four curves, we can also find that ESBER_I method is the most stable method, and the changing trend of average accuracy value against the data amount of the dataset is not rapid. Therefore, ESBER_I method can obtain stable and effective clustering results when processing large and multi-class datasets.

4.5. Extension of ESBER_I

In the above experiments, ESBER_I method only utilizes the 10% data points with ground-truth label information to choose a suitable eigenvector combination for spectral clustering. In order to utilize the label information fully, we further use the label information to guide the data clustering. In detail, we apply a support vector machine (SVM) classifier [5] to classify the data consisting of the optimal eigenvectors selected by ESBER_I, and call this method as ESBER_SVM. In this section, we take the UCI benchmark datasets for example to verify the performance of ESBER_SVM, and adopt SVM as a comparative method. For SVM, we use the 10% data points with ground-truth label information offered for ESBER_I as the training dataset and adopt the original data as the testing dataset. In this experiment, we conduct SVM with Gaussian radial-basis-function (RBF) kernel and perform 5-fold cross validation for parameter tuning. The average clustering accuracy values of 10 independent runs of ESBER_I, SVM and ESBER_SVM are recorded in Table 5. It is found from Table 5 that ESBER_SVM
can achieve better performances than ESBER_I for these four UCI datasets. Except the Iris dataset, ESBER_SVM outperforms SVM in the average clustering accuracy values.

5. Conclusions

In this paper, we present a novel eigenvector selection method based on entropy ranking for spectral clustering (ESBER). We give two versions of ESBER method. One is a direct version (ESBER_D) and the other is an indirect version (ESBER_I). The experiments on real datasets show that two versions of ESBER method are more robust than NJW method. For ESBER_I method, we draw 10% samples from the dataset as the training data and experiments have demonstrated that 10% data for training can bring suitable eigenvector combinations for spectral clustering. Furthermore, ESBER method can also perform well on large-scale datasets with the cluster no. K > 10.

Even though our method can achieve promising performances, there is still much work for us to undergo further research. In this paper, we adopt entropy to rank eigenvectors, which is a simple feature ranking method. We plan to utilize other feature ranking methods to rank eigenvectors, especially an ensemble of multiple rankings of eigenvectors. Furthermore, the computational complexity of ESBER method, especially ESBER_I method, is higher than that of NJW method. Other eigenvector selection methods for spectral clustering with a computational complexity comparable to that of NJW method also deserves to be studied. We will pursue these research directions in the future work.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant nos. 60703108, 60702062, 60970067), the National High Technology Research and Development Program (863 Program) of China (Grant no. 2009AA12Z210), the National Basic Research Program (973 Program) of China (Grant no. 2006CB705707) and the Program for Cheung Kong Scholars and Innovative Research Team in University (Grant no. IRT0645).

Appendix A. Indirect eigenvector selection using Genetic algorithm

Genetic algorithm (GA) [15] has emerged as a practical optimization and search method. It is a robust search method that requires little information to search effectively in a large or poorly understood search space [7]. In each generation, three basic operators, that is, reproduction, crossover, and mutation, are performed to generate a new population with a constant population size.

The search space of GA is all the column subsets of Vtrain. In our encoding scheme, each chromosome represents an eigenvector combination. The chromosome is a bit string consisting of Km genes (one for each eigenvector). Each gene in the string contains either 0 or 1. If the jth bit is 1, then the jth eigenvector is selected, otherwise, that eigenvector is ignored. The fitness function is also defined by Eq. (6).

Moreover, the reproduction, crossover, and mutation operators are defined as follows:

- **Reproduction**: This operator is based on the principle of the survival of the fitness. In the population A = {a1, a2,..., am}, those individuals with large fitness f(ak) have a large number of copies (k) in the new generation. For example, in roulette wheel selection, the ith individual with a high fitness value is given a proportionately high probability of reproduction. Once the individuals are reproduced, a new population (k) = (i(k), i = 1, 2,...m) is generated in the next generation.
- **Crossover**: In crossover, the individuals exchange information via probabilistic decisions. A commonly used method for crossover is called multi-point crossover, which provides more hybridizing opportunity for the optimized individuals. For multi-point crossover, p crossover positions are randomly chosen with no duplicates and sorted into ascending order. Then, the genes between successive crossover points are exchanged between the two parents to produce two new offsprings.
- **Mutation**: When the jth position of the ith individual is chosen, the mutation operator simply changes the state from 0 to 1, or vice versa. Mutation should be a very low probability operator, as GA with a high mutation probability degrades to a random search [8,24].

The procedure for indirect eigenvector selection method using GA is given as follows:

**Algorithm 2. Indirect eigenvector selection using Genetic algorithm**

**Input**: The training dataset from the original dataset X = {x1, x2,...,xn}; the matrix V ∈ ℝn×Km consisting of the first Km eigenvectors in the ranking list of all the eigenvectors.

**Output**: The optimal eigenvector combination ec*

1. Extract the corresponding part of the training data in V and denote it by Vtrain ∈ ℝn×Km, where tn is the number of the training data.
2. Generate random population A(0) = {a1(0), a2(0),...,am(0)} and set the number of iteration k = 0.
3. Form the matrix Utrain ∈ ℝtn×l from Vtrain by only adopting the eigenvectors in the ai(k). Compute the corresponding fitness value f(ai(k)) (1 ≤ i ≤ m) of the population using Eq. (6).
4. Judge the closing condition until a suitable number of iteration is obtained. If it is satisfying the term, the current population is considered as the optimal one and go to step (6). Otherwise the next step is executed.
5. Use reproduction, crossover, and mutation to generate new chromosomes in the next generation, k = k + 1, then go to step (3).
6. Select the best individual from the optimal population which obtains the high fitness value as the optimal eigenvector combination ec*.

The notations of GA are given as follows: Ga is the maximum iterative number in GA, ps the population size Prob_m and Prob_c are the mutation probability and crossover probability, respectively. The computational complexity of this algorithm mainly consists of the encoding, reproduction, crossover, mutation, and fitness evaluation. At per iteration of GA, the complexity of reproduction, crossover and mutation is O(ps), O(ps × Proc_m) and O(ps × Proc_c). We utilized the 2-way merge sort as the fitness value sorting algorithm whose time complexity is O(ps × log2 ps). Therefore, the time complexity of the entire GA in our method is Ga(O(ps + ps × Proc_m + ps × Proc_c) + O(ps × log2 ps)).

References

Feng Zhao is currently a Ph.D. student of Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China at Xidian University, He received his B.Sc. degree from Heilongjiang University, in 2004, and the M.Sc. degree in Institute of Intelligent Information Processing from Xidian University in 2007. His current research interests include image processing, machine learning, computational intelligence and data mining.

Licheng Jiao received the B.S. degree from Shanghai Jiao Tong University, Shanghai, China, in 1982, and the M.Sc. and the Ph.D. degree from Xi'an Jiao Tong University, Xi'an, China, in 1984 and in 1990, respectively. His research interests include signal and image processing, natural computation, and intelligent information processing. He is an IEEE senior member, a member of the IEEE Xi'an Section Executive Committee and the chairman of its Awards and Recognition Committee, and an executive committee member of the Chinese Association of Artificial Intelligence.

Hanjiang Liu is a Ph.D. student of Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China at Xidian University. He received his B.Sc. degree from Heilongjiang University, in 2004, and the M.Sc. degree in Institute of Intelligent Information Processing from Xidian University in 2007. His current research interests include SAR image processing, machine learning, computational intelligence and data mining.

Xinbo Gao received the B.Sc., M.Sc., and Ph.D. degrees in signal and information processing from Xidian University, China, in 1994, 1997, and 1999, respectively. He joined the Department of Electric Engineering, Xidian University, as a lecturer in 1999 and is currently Professor, Director of the VIPs Lab, and Director of the International Office, Xidian University. His research interests include computational intelligence, machine learning, information processing and analysis, pattern recognition, and artificial intelligence.

Maoguo Gong is currently an associate professor with the Intelligent Information Processing Innovative Research Team of the Ministry of Education of China at Xidian University, Xi'an, China. He received his B.Sc. degree in electronic engineering from Xi'an University, Xi'an, China, in 2003 with the highest honor. His areas of special interest include artificial immune systems, evolutionary computation, image understanding, data mining, optimization, and some other related areas.