IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 4, 2017, accepted January 10, 2018, date of publication February 12, 2018, date of current version May 2, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2805365

DOE-AND-SCA: A Novel SCA Based on DNN With Optimal Eigenvectors and Automatic Cluster Number Determination

JINYIN CHEN, YANGYANG WU, XIANG LIN, AND QI XUAN
College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China

Corresponding authors: Jinyin Chen (chenjinyin@163.com) and Yangyang Wu (2111603080@zjut.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61502423 and Grant 61572439.

ABSTRACT Spectral clustering algorithm (SCA) is one of the widely used clustering algorithms (CAs), which is proved to be efficient in many applications including unsupervised image identification and gene prediction. However, most SCAs are confronted with several problems: 1) It is difficult for SCAs to handle multi-scale data sets; 2) It is difficult to set cluster number in advance for various applications; 3) It is also difficult to choose the most appropriate eigenvectors to reflect the data distribution; and 4) Moreover, SCAs are sensitive to the parameters. To handle these problems, we propose a novel SCA based on dynamic nearest-neighbors (DNN) with optimal eigenvector and automatic cluster number determination, namely DOE-AND-SCA. There into, first, we design a novel similarity function based on DNN for multi-scale data, making the similarity metric more accurate; Second, the cluster number is automatically determined, and the cluster centers are also automatically determined by normal fitting, based on the density and minimum distance distribution of the data points; Third, the optimal eigenvectors are selected on the basis of global and local features of the data set for more accurate data distribution reflection; Fourth, two main parameters, including the optimal density difference threshold and the number of intervals, are self-adaptive. The efficiency of DOE-AND-SCA is testified on abundant of simulation data sets, by comparing with other outstanding algorithms. And finally, DOE-AND-SCA is also applied to image recognition problems.

INDEX TERMS Spectral clustering algorithm, dynamic nearest-neighbors, automatic cluster number determination, optimal eigenvector, parameter self-adaptive, image recognition.

I. INTRODUCTION

Clustering is one of the essential problems in many research fields. Among proposed CAs, SCA [1]–[3], a basic CA which is based on spectral theory [4], frequently yields better performance comparing to the other CAs, for example, K-means algorithm [5].

Much effort has been devoted for developing novel SCAs in recent years. There are some classical SCAs, e.g., the Ng-Jordan-Weiss algorithm (NJW) [6] proposed by Ng *et al.* In NJW, optimal eigenvectors are selected based on K eigenvectors with largest eigenvalues of Laplacian matrix to reflect the responding data distribution between raw data original distribution and its feature space. Elhamifar and Vidal proposed the sparse subspace clustering algorithm [7] to cluster the data points that located in the low-dimensional subspaces. Fowlkeset *et al.* proposed the *Nyström* algorithm (NJWN) [8] which is based on the NJW algorithm. It can effectively reduce the complexity of SCA. But NJWN algorithm relies heavily on the selection of the initial points. Zhao et al. developed a fuzzy similarity measure for SCA [9], by utilizing the partition matrix, which is obtained by the fuzzy c-means clustering algorithm. It is quite effective and stable, but the space complexity of this algorithm is relatively high. Wang et al. designed an ascertainable clustering number algorithm using SCA [10]. It can automatically determine the suitable clusters number, but the proposed algorithm is sensitive to its parameters. Chen and Cai [11] proposed the landmark-based spectral clustering algorithm. This proposed algorithm selects some representative data points as the landmarks and represents the original data points as the linear combinations of these landmarks. Graph-based relaxed clustering, as one of the SCAs, is sensitive to the parameters of the adopted similarity measure. In order to overcome this shortcoming, Qian *et al.* proposed the fast graph-based relaxed clustering

algorithm [12] based on the constrained GRC developed by using the core-set-based minimal enclosing ball approximation. Xia *et al.* proposed the robust multi-view spectral clustering algorithm [13], which handles the noise points in the multi-view data set and constructs the transition probability matrix via sparse decomposition. Tremblay *et al.* [14] proposed a compressive spectral clustering algorithm. It speeds up the step of selecting eigenvectors and running K-means. However, the clustering effect of this algorithm is disappointing when processing multiple-scale data sets. Moreover, Passalis and Tefas [15] proposed a spectral bag-of-features clustering algorithm. It views the histogram space as an intermediate space between the feature and the spectral space. But it needs to manually determine the number of clusters as well.

Generally, although a number of improved versions of SCAs have been proposed, there are still several major challenges for these SCAs as follows.

- Multi-scale data sets are quite common in real applications, which usually consist of sparse clusters and dense clusters. For most SCAs, it is quite challenging to construct an appropriate similarity matrix to reflect the distribution of the multi-scale data set.
- Most SCAs need to manually determine the number of clusters, which may lead to bad clustering results in real applications.
- Currently, most SCAs choose eigenvectors purely depending on the eigenvalues of Laplacian matrix which, however, may not reflect the actual data distribution.
- Many SCAs are sensitive to the predefined parameters, which may lead to overfitting, and thus decrease their efficiency.

In order to handle these raised problems, in this paper, we propose a novel SCA based on DNN with optimal eigenvector and automatic cluster number determination, which is called DOE-AND-SCA. Firstly, we design a novel similarity function based on DNN for multi-scale data; Then, the cluster centers are automatically determined by constructing a normal distribution function for density-minimum distance; Next, the optimal eigenvectors are selected according to eigenvalues and Laplace scores; Finally, the optimal density difference threshold and interval number, which are two main parameters in DOE-AND-SCA, are self-adaptive.

The rest in this paper is organized as follows. In Sec. II, we introduce the previous works related to the SCA. In Sec. III, we describe our DOE-AND-SCA in detail. In Sec.IV, we experimentally evaluate the performance of the DOE-AND-SCA on several artificial data sets. We then apply the DOE-AND-SCA on several real data sets in Sec. V, and finally conclude this paper in Sec. VI.

II. RELATED WORKS

Given a set of n points $x_1, x_2, \cdots, x_n \in \mathbb{R}^m$, SCA first constructs an undirected graph according to its similarity matrix $S = (S_{ij})_{i,j=1}^n$, where $S_{ij} \geqslant 0$ is the similarity between x_i and x_j. The degree matrix D is a diagonal matrix, and its diagonal elements is represented by $D_{ii} = \sum_j S_{ij}$. Let $L = D^{-1/2}SD^{-1/2}$. Then, SCA selects the top K (clusters number K) eigenvectors based on the related eigenvalues. Finally, the K-means algorithm is applied to obtain the clustering result according to the selected eigenvectors. Although SCA work well when processing complex data sets, there are still some problems with it.

A. HANDLING MULTI-SCALE DATA SETS

Compared with the traditional CAs, SCA can better handle the data sets of complex shapes. It does not require estimating an explicit model of data distribution, but only needs a spectral analysis of point-to-point similarities. However, traditional SCAs may still be ineffective when clustering multi-scale data set.

In order to better handle multi-scale data sets, many SCAs optimized the similarity matrix. Yang *et al.* proposed a density sensitive spectral clustering algorithm [16], which can squeeze the distances in high density regions and widen the distances in low density regions. Zhang *et al.* proposed a SCA with local density adaptive similarity [17], which uses point-to-point local density difference to scale the Gaussian similarity function. The proposed similarity measure has effect of enlarging intra-cluster similarity and reducing inter-cluster similarity. Beauchemin [18] proposed a density-based similarity matrix construction for SCA. The idea of defining similarity from nonparametric density estimator was discussed in [19], where a link between graph-cut and kernel density estimation was established.

Although these SCAs have optimized the similarity matrix in various ways, the authors often neglect the division of boundary points between clusters in multi-scale data set.

B. DETERMINING THE NUMBER OF CLUSTERS

How to determine a suitable number of clusters is a common problem for almost all CAs, certainly including SCAs. The number of clusters is manually chosen for many SCAs, but there are also several approaches to automatically determine the cluster number.

Wang [20] proposed a novel SCA, which is determined the number of clusters from the slope difference distribution of the data set. This proposed method is composed of two parts: computation of the slope difference distribution from the data distribution and selection of the peaks of the slope distribution as the cluster centers. Manor and Perona proposed a SCA [21] that automatically computed the number of clusters by minimizing the cost of aligning a set of eigenvectors with a canonical coordinate system. Mur et al. proposed the spectral global silhouette algorithm [22]. GS uses SCA together with the Silhouette Validity index [23] and the concept of local scaling, which allows finding the number of clusters automatically. Wacquet *et al.* [24] proposed a new K-way semi-supervised spectral clustering method. The authors used a criterion based on an outlier number minimization to automatically determine the number of clusters. Borjigin et al. proposed

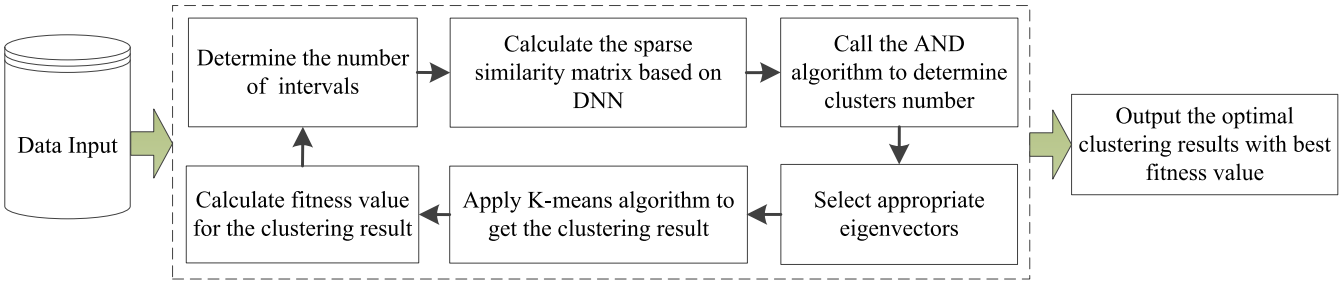


FIGURE 1. The overall framework of the DOE-AND-SCA algorithm.

a non-unique cluster number determination methods in SCA [25]. This algorithm utilized SCA to cluster data set for an initial the number of clusters K at first. Then, the standard, which is the ratio of the multiway normalized cut criterion of the obtained clusters and the sum of the leading eigenvalues of the stochastic transition matrix, is chosen to decide whether K is the optimal clusters number.

Most SCAs tried to design evaluation criteria to determine the number of clusters by iteration, which may largely increase the time complexity of the algorithms.

C. SELECTING EIGENVECTORS

SCAs utilize the eigenvectors of the Laplacian matrix to cluster data set. NJW algorithm is one of the most popular SCAs. For a K clustering problem, NJW algorithm always partitions data using the K eigenvectors, which is selected according to the eigenvalues of the Laplacian matrix. Although the spectral relaxation solution of normalized cut criteria lies in the subspace spanned by these eigenvectors [26], it is not guaranteed that this subspace matches the structure of the data well.

Xiang and Gong [27] were the first to utilize eigenvectors selection to improve the clustering effect of SCA. The proposed algorithm firstly finds the K_k eigenvectors with the largest eigenvalues and the relevance of each eigenvector is estimated according to how well it can partition the data points. Eventually it preserves all the relevant eigenvectors. Jiang and Ren [28] proposed a novel eigenvector evaluation criterion based on the perturbation analysis. To evaluate the importance of a eigenvector, the authors perturbed the value of this eigenvector by introducing a perturbation factor to it for all the data points. In this proposed algorithm, if a small perturbation of one eigenvector causes a great disturbance of all the eigenvectors, this eigenvector is important for SCA. Hosseini and Azar proposed a novel strategy [29] of mitigating the undesired properties of high dimensionality to develop SCA. The proposed algorithm focused on introducing an objective function based on three measurement functions which evaluated the ability of each eigenvector in data set, i.e., the compactness of clusters, the interval between clusters, and the stability of clustering to recognize the best using eigenvectors. Zhao *et al.* [30] proposed an eigenvector selection method based on entropy ranking for SCA. In the proposed algorithm, according to the importance of eigenvectors on clustering, all the eigenvectors are ranked at first. And then a suitable eigenvector combination is obtained from the ranking list.

Similarly, these SCAs used the iterative way to determine the eigenvectors based on the evaluation index, and thus the time complexity is relatively high. In order to select the eigenvectors more quickly, we should evaluate how much information can be provided about each eigenvector.

D. PARAMETER DEPENDENCY

The clustering results of many SCAs are particularly dependent on the selection of parameters.

Fowlkeset *et al.* proposed an algorithm [8] which substantially reduces the computational requirements of grouping algorithms based on SCA. It allows one to extrapolate the complete grouping solution using only a small number data points. But this algorithm relies heavily on the selection of the initial sample points, and its clustering result is unstable. Nguyen et al. proposed an automatic unsupervised spike sorting method using the landmark-based spectral clustering algorithm (LSC) [31], which is based on the locality preserving projection technique that utilize to extract features. Before the LSC method can be performed, Gap statistics [32] is used to determine the number of clusters. But the LSC algorithm is sensitive to the parameter, which is the number of landmarks. Arias-Castro *et al.* [33] proposed a SCA based on the PCA method. After performing the PCA method [34] in selected neighborhoods, this algorithm builds a nearest-neighbor graph weighted according to a discrepancy between the principal subspaces, and then applies SCA. But it has many parameters that need to be determined manually, e.g., the neighborhood size and the projection scale. Tremblay *et al.* proposed a novel SCA [35] that avoids the computational bottleneck of extracting the Laplacian's eigenvectors. However, the impact of the error of the polynomial approximation on the algorithm is largely unknown.

III. DOE-AND-SCA ALGORITHM

In order to handle the four problems that many SCAs are confronted with, we proposed the DOE-AND-SCA

algorithm, with the overall framework shown in FIGURE 1. The main contributions of our method include the following four aspects:

- For multi-scale data sets, those data points at the boundaries of different clusters may have relatively high similarity of traditional metrics, which may lead to bad clustering results. We thus propose a novel similarity metric based on DNN, to reduce the similarity between the boundary points with large density difference, so as to better reflect the actual data distribution.
- We design a mechanism to automatically determine the number of clusters and the cluster centers. Based on the density and minimum distance distribution of the data points, we first identify the singular points by a normal fitting, and then filter the singular points according to certain conditions. The remaining singular points are set to cluster centers, and thus the number of clusters is also automatically determined.
- We choose the optimal eigenvectors based on the global and local features of the data set, to better reflect the data structure. In particular, the principal eigenvectors are chosen according to the eigenvalues to reflect the global data feature, while the required non-principal eigenvectors are chosen from the rest according to the Laplace scores to reflect the local data feature.
- We design a mechanism to make the two main parameters, including the optimal density difference threshold and the number of intervals, self-adaptive.

A. DNN BASED SIMILARITY METRIC

Typically, a multi-scale data set is of quite different distribution densities in different clusters. Most SCAs don't seriously consider the division of the boundary points of clusters, it is difficult for them to achieve satisfied clustering results on multi-scale data sets.

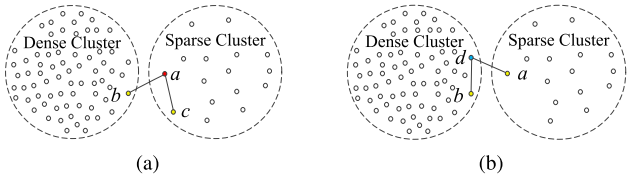


FIGURE 2. Examples of multi-scale data sets.

Here we give an illustration to show how the similarity metric based on DNN works. FIGURE 2 shows two multi-scale data sets, both consisting of a sparse cluster on the right and a dense cluster on the left. The data points a, b, c and d are the four boundary points.

Traditionally, the Gaussian similarity function of SCA is defined as

$$S_{ij} = \exp\left[\frac{-d^2(i,j)}{2\sigma^2}\right], \tag{1}$$

where $d(i,j)$ represents the distance between data points i and j, and σ is the scaling parameter.

In FIGURE 2 (a), assuming $d(a,b) = d(a,c)$, based on Eq. (1), we will have $S_{ab} = S_{ac}$. That is, the similarity between data points a and b is equal to the similarity between data points a and c. In fact, for any data point, the similarity with the data point in same cluster is higher than the similarity with the data point in different cluster. So, this is clearly inconsistent with the fact that a and c belong to the same cluster while a and b belong to different clusters. In other words, the similarity metric defined by Eq. (1) fails to distinguish such difference.

In order to solve this problem, Zelnik-Manor and Perona proposed a self-tuning spectral clustering algorithm (STSC) [21], where local-scale parameter, rather than global-scale parameter, is used in the Gaussian kernel function. The similarity matrix thus is expected to better reflect the data structure. The Gaussian similarity function of the STSC algorithm is defined as

$$S_{ij} = \exp\left[\frac{-d^2(i,j)}{2\sigma_i\sigma_j}\right], \tag{2}$$

where $\sigma_i = d(i,t)$ represents the distance from the data point i to its t-th nearest-neighbor.

According to the definition of the local-scale parameter, we can find that $\sigma_c > \sigma_b > 0$ and $\sigma_a\sigma_c > \sigma_a\sigma_b > 0$ in FIGURE 2 (a). Then, based on Eq. (2), we can find that $S_{ab} < S_{ac}$, consistent with the fact here.

In FIGURE 2 (b), assuming $d(d,b) = d(d,a)$, based on the definition of the local-scale parameter, we can find that $\sigma_a > \sigma_b > 0$ and $\sigma_a\sigma_d > \sigma_b\sigma_d > 0$. Thus, according to Eq. (2), we can find the similarity S_{ad} between a and d is larger than the similarity S_{bd} between b and d, which however is inconsistent with the fact that the similarity S_{bd} between b and d is larger than the similarity S_{ad} between a and d.

In other words, although the similarity metric defined by Eq. (2) seems better than that defined by Eq. (1), it could still be further improved. We thus propose a novel similarity metric based on DNN.

Definition 1: For each data i, its *density* is defined as

$$\rho_i = \sum_j f(d(i,j)), \tag{3}$$

$$f(x) = \begin{cases} 1 & x \in SD \\ 0 & x \notin SD \end{cases} \tag{4}$$

where the set SD is composed of the np_{ercent} smallest values in the distance matrix, p_{ercent} is the ratio of the average density of the data points.

By observing FIGURE 2, we can find the data points a and c should be grouped into the same cluster because a and c are two boundary points with similar density. Since the density difference between data points has an impact on the similarity between them, we propose a novel similarity metric based on DNN.

Definition 2: The *DNN set* T_i for each point i is defined

$$T_i = \{j \in N_i | d(i,j) < \min_{k \in G_i}(d(k,i))\}, \tag{5}$$

$$G_i = \{j \in N_i | |\rho_i - \rho_j| > \theta\}, \tag{6}$$

where N_i is the set, which is composed of initial nearest-neighbor for the data point i, θ is the density difference threshold.

Thus, the proposed similarity function is defined as

$$S_{ij} = \begin{cases} \exp\left[\dfrac{-d^2(i,j)}{2\left(\max\{\bar{\sigma}_i, \bar{\sigma}_j\}\right)^2}\right] & j \in T_i \\ 0 & j \notin T_i \end{cases} \tag{7}$$

$$\bar{\sigma}_i = \sum_{j \in T_i} \frac{d(i,j)}{t_i} \tag{8}$$

where t_i is the number of data points in the set T_i. The local-scale parameter of each data point, defined in Eq. (8), is determined by the average distance with its DNNs. The maximum local-scale parameter for data points i and j is represented by $\max\{\bar{\sigma}_i, \bar{\sigma}_j\}$.

At this time, as we can see in FIGURE 2, b and d are located in dense clusters, while a and c are located in sparse clusters. Assuming that $|\rho_a - \rho_b| > \theta$, $|\rho_a - \rho_d| > \theta$, $|\rho_a - \rho_c| < \theta$, and $|\rho_d - \rho_b| < \theta$, Based on the definition of DNN set, we can conclude that $a \notin T_b$, $a \notin T_d$, $c \in T_a$, and $b \in T_d$. Therefore, according to Eqs. (7) and (8), we can get $S_{ac} > S_{ab} = 0$ and $S_{bd} > S_{ad} = 0$, and these results are consistent with the fact.

Another problem of SCAs is that the authors always store the whole similarity matrix, leading to relatively high space complexity. In this paper, we construct a sparse similarity matrix to solve this problem.

In particular, we only consider those significant relationships between DNNs. Three ways are adopted to shrink the storage space. First, the whole data sampling space is divided into multiple intervals. Only the distances between the data points in each interval and all the data points are calculated in each iteration. Second, for each data point, only its similarities with the DNNs will be stored instead of its similarities with all the rest data points in the same interval. Third, sparse matrix is adopted to keep all similarities. According to the similarity metric based on DNN and all interval sparse distance matrices, we can get the sparse similarity matrix based on DNN. Thus, the space complexity of the sparse similarity matrix based on DNN is $O(nb)$, where n is the number of data points and b is the number of intervals.

B. AUTOMATIC CLUSTER NUMBER DETERMINATION

Fast density clustering algorithm (FDC) [36], introduced by Rodriguez and Laio, has attracted much attention for its outstanding performance but simple operation. According to FDC, cluster centers, general cluster members and noise points could be distinguished through the distribution mapping of density and distance of the data points. As the authors described, cluster centers have relatively larger density and larger distance from each other, cluster centers could be found manually in different applications. Inspired by FDC, we propose a method to determine the cluster centers and the number of clusters, namely automatic cluster number determination (AND). Different from FDC where the clustering centers were observed or manually selected from density-distance mapping, in our method, the cluster centers and the number of clusters are automatically determined by constructing a normal distribution function for density-distance mapping to figure out all the singular points.

Definition 3: For each data point i, if the densities of data points in T_i are all smaller than itself, the data point i can be judged as the *candidate point*, otherwise, namely *non-candidate point*. The *minimum distance* δ_i of each candidate point i is defined as the minimum distance with the data point that has higher density.

$$\delta_i = \begin{cases} \min\{D_h(i)\} & \rho_i \neq \max(\rho) \\ \max(\delta) & \rho_i = \max(\rho), \end{cases} \tag{9}$$

where $D_h(i)$ is the set of distances between data point i and the data points of higher density in T_i, $\max(\rho)$ is the maximum density in data set, and $\max(\delta)$ is the maximum *minimum distance* of all the data points. The *minimum distance* of each *non-candidate point* i is defined as

$$\delta_i = \min\{D_n(i)\}, \tag{10}$$

where $D_n(i)$ is the set of distances between i and its DNNs of higher density.

In order to determine the cluster centers more accurately, we further define a variable for each data point that considers both the density and the minimum distance of the data points. For each data point i, according to definitions Eqs. (10) and (9), we define γ_i as:

$$\gamma_i = \rho_i \times \delta_i. \tag{11}$$

As shown in FIGURE 3, there are corresponding relationships among data original distribution, $\rho - \delta$ distribution, and density distribution of γ for Iris data set. In FIGURE 3, cluster centers are represented as $A1$, $A2$, and $A3$. They have relatively larger ρ, δ and γ values. The rest data points have smaller ρ or δ values. The curve of the density distribution of γ is fitted and it is found that the fitting curve of the density distribution of γ is similar to a normal distribution curve. The confidence interval can be easily determined according to the normal fit curve, and use this confidence interval to find singular points. As shown in FIGURE 3(c), it is obvious that these cluster centers are all singular points.

Suppose γ follows a Gauss distribution, with its mathematical expectation denoted by μ and the variance denoted by σ^2. First, we calculate the sample mean $\bar{x}$ and sample variance $\bar{S}$, then we can get μ and σ^2 according to the principle of moment estimation.

$$\mu = \bar{x}, \quad \sigma^2 = \frac{N-1}{N}\bar{S}. \tag{12}$$

However, when we analysis the density distribution of γ, we find that all the values of γ are non-negative, indicating that the density distribution of γ is not strictly normal distribution. Therefore, to accurately estimate the values of μ and δ, we need to compensate for the value of γ in the negative half axis based on symmetry.

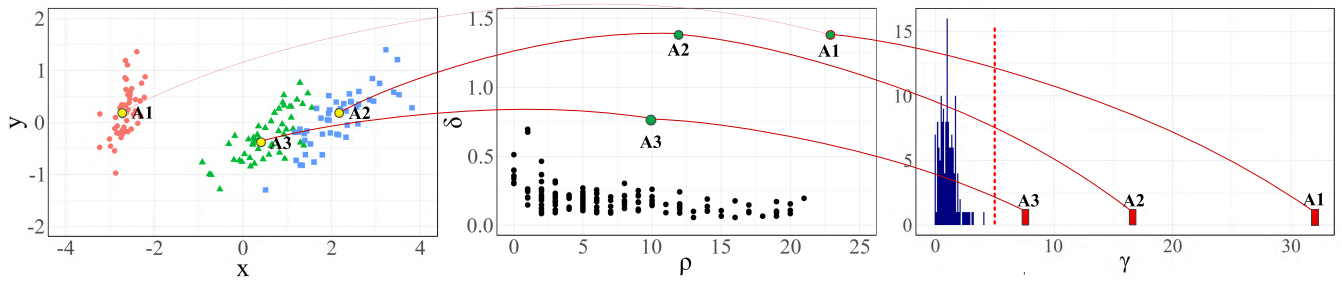


FIGURE 3. Mapping relationships among the data original distribution, $\rho - \delta$ plane, and density distribution of γ.

Thus, we first calculate the mean of all the data points, denoted by $\bar{x}_1$, and select the points in interval $[0, 2\bar{x}_1]$. Then, we obtain the mean of these points, denoted by $\bar{x}_2$. Again, we select the points in interval $[0, 2\bar{x}_2]]$, and obtain the mean of these points, denoted by $\bar{x}_3$, and so forth. This iteration process is terminated until the mean value keeps relatively stable. We denote the final mean value as $\bar{x}_0$. According to the principle of symmetry, we map the data points in the interval $[\bar{x}_0, \infty]$ to the interval $[-\infty, 0]$ by using $x = \frac{\bar{x}_0}{2}$ as the axis of symmetry. Finally, we calculate the sample variance $\bar{S}_0$ and use Eq. (12) to calculate the expectation μ and the variance σ^2.

Now let's determine the cluster centers. Typically, the singular points have greater γ value than normal data points. The experiments show that the number of singular points selected by this method is larger than the number of real clusters in the data set. So, we need to further filter the selected singular points. We find that some singular points have either relatively large ρ or relatively large δ. In combination with the physical meaning of the $\rho - \delta$ plane, it is easy to know that these points are either close to the clustering centers or just noises, i.e., they are not clustering centers. In other words, the true clustering centers can be found by filtering out these points. In reality, we first normalize the values of ρ and δ of data points, obtaining $\bar{\rho}$ and $\bar{\delta}$, and set the filter ratio ω. For data point i, if $1/\omega < \bar{\rho}_i/\bar{\delta}_i < \omega$, the singular point will be selected as a cluster center. Based on this idea, the AND algorithm is presented in **Algorithm 1**.

Algorithm 1 The AND Algorithm

Input: Interval distance matrices
Output: Cluster centers

1: According to interval distance matrices, calculate ρ, δ and γ for each point;
2: Draw $\rho - \delta$ plane and the density distribution of γ;
3: Calculate the mean of points, denoted by x_1, and select the data points in $[0, 2x_1]$ to calculate μ and σ about γ;
4: Make use of the normal distribution curve of γ through μ and σ, and find the confidence interval;
5: Use this confidence interval to find cluster centers.
6: **return** Cluster centers

C. OPTIMAL EIGENVECTOR SELECTION

When we obtain the sparse similarity matrix S and its Laplacian matrix L, a popular solver called *ARPACK* [37] can be adopted to quickly get the first K eigenvectors of L. But many researchers pointed out that the first K eigenvectors of the Laplacian matrix may be uninformative and inappropriate for SCA. In order to select the eigenvectors which can better reflect the data structure, we propose an optimal eigenvector selection algorithm (*OE*): First, according to the eigenvalues, the principal eigenvectors are selected, which can express the global feature of the data set. Then, the required non-principal eigenvectors are selected according to the Laplace score method, which can reflect the local features of the data set. The steps of *OE* algorithm are presented in **Algorithm 2**.

Algorithm 2 The OE Algorithm

Input: The DNN based sparse similarity metric S
Output: K eigenvectors;

1: Calculate degree matrix D and Laplacian matrix L;
2: Use ARPACK to obtain first $2K$ eigenvectors of L;
3: Select eigenvectors with eigenvalues of 1, its number is p;
4: Calculate the Laplace score of the remaining eigenvectors, and select the $K - p$ eigenvectors with least Laplacian score;
5: **return** K selected eigenvectors.

Since Laplacian matrix L is block diagonal, its eigenvalues and eigenvectors are the union of the eigenvalues and eigenvectors of its blocks. The formula for the calculation of the block Laplacian matrix is shown in Eq. (17), where D denotes the degree matrix and S^{ij} denotes the block similarity matrix between the data points in the ith region and in the jth region. Assume the data set has n data points and K clusters. The data could be divided into p $(1 \leq p \leq K)$ regions. Since the distances between the data points in the ith $(i = 1, \cdots, p)$ region and the data points in other regions are relatively large, we have the block similarity matrix $S^{ij} \approx 0\,(i \neq j)$. According to Eq. (17), we have the block Laplacian matrix $L^{ij} = 0$ $(i \neq j)$. It can be observed that L^{ij} has a strictly positive principal eigenvector with eigenvalue 1. This principal eigenvector

can help to distinguish the data points in the ith region from the data points in other regions [38].

$$S = \begin{bmatrix} S_{11} & \cdots & S_{1n} \\ \vdots & \ddots & \vdots \\ S_{11} & \cdots & S_{nn} \end{bmatrix} = \begin{bmatrix} S^{11} & \cdots & S^{1p} \\ \vdots & \ddots & \vdots \\ S^{p1} & \cdots & S^{pp} \end{bmatrix} \tag{13}$$

$$D = \begin{bmatrix} D_{11} & & \\ & \ddots & \\ & & D_{nn} \end{bmatrix} \tag{14}$$

$$L = \begin{bmatrix} L^{11} & \cdots & L^{1p} \\ \vdots & \ddots & \vdots \\ L^{p1} & \cdots & L^{pp} \end{bmatrix} \tag{15}$$

$$D_{ii} = \sum_j S_{ij} \tag{16}$$

$$L^{ij} = (D_{ii})^{-\frac{1}{2}} S^{ij} (D_{jj})^{-\frac{1}{2}} \tag{17}$$

Moreover, we need to select the non-principal eigenvectors which can better reflect the local features of the divided regions. Here, Laplace score method [39] is used to select non-principal eigenvectors. The Laplace score method can represent the local retention ability of the non-principal eigenvectors to the DNNs. The Laplace score of the $r-$th eigenvector L_r is defined as:

$$L_r = \frac{\sum_{i,j}(f_{ri} - f_{rj})^2 S_{ij}}{\sum_i (f_{ri} - u_r)^2 D_{ii}}, \tag{18}$$

where f_{ri} is the rth feature of data point i, u_r is the mean value of f_{ri} $(i = 1, \cdots, p)$, S_{ij} is the similarity between data points i and j. The smaller Laplace score of the eigenvector, the better the eigenvector can preserve local structure information.

D. OPTIMAL DENSITY DIFFERENCE THRESHOLD

In DOE-AND-SCA, different density difference thresholds will result in different DNN sets. Moreover, it also has a certain effect on the structure of the sparse similarity matrix and the clustering result. So, the selection of the density difference threshold is very important.

As usual, we assume that a better clustering effect corresponds to the smaller distance within the cluster while the larger distance between clusters. As described in Eqs. (19) and (20), the mean distance between all the data points in each cluster and the corresponding cluster center are adopted to represent the global average intra-cluster distance. And the mean distance between different cluster centers is used to represent the global average inter-cluster distance.

$$\zeta_i = \sum_{j \in C_i} \frac{d(j, \alpha_i)}{|C_i|} \tag{19}$$

$$\eta(C_i, C_j) = d(\alpha_i, \alpha_j) \tag{20}$$

where α_i is the clustering center of cluster C_i, and $|C_i|$ is the number of data points in the cluster.

We design a fitness function, as presented in Eq. (21), to measure clustering effect.

$$Fitness = \frac{2}{(K-1)K} \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} \frac{\zeta_i + \zeta_j}{\eta(C_i, C_j)} \tag{21}$$

Generally, give a density difference threshold, we can calculate the above fitness value. Smaller fitness value is, the better the clustering effect for the density difference threshold. Thus, we can determine the optimal density difference threshold.

E. INTERVALS NUMBER DETERMINATION

In order to reduce the space complexity of the algorithm, we divide the sampling space into multiple intervals. However, increasing the number of intervals will be accompanied by increasing the time complexity of the algorithm. So, we need to consider the balance between time complexity and space complexity when determining the number of intervals.

The time complexity of the step, which is related to the number of intervals in the DOE-AND-SCA algorithm, is $O(nmb + nb\log t)$, while the space complexity of our algorithm is $O(n^2/b)$, where n is the number of data points, m is the dimensionality of each data point, b is the number of intervals, and t is the average number of nearest-neighbors for all data points. Due to the limited storage space of the computer, the space complexity of the DOE-AND-SCA algorithm should be less than the space complexity threshold $O(S_{max})$. Thus, we can prove that the number of intervals is determined as:

$$b = \begin{cases} \dfrac{n^2}{S_{max}} & \left(n^3\,(m + \log(t))\right)^{\frac{1}{2}} > S_{max} \\ \left(\dfrac{n}{m + \log(t)}\right)^{\frac{1}{2}} & \left(n^3\,(m + \log(t))\right)^{\frac{1}{2}} \le S_{max} \end{cases} \tag{22}$$

Proof: Denoting by $S_{pace} = nmb + nb\log t$ and $T_{ime} = n^2/b$, we define the mixed complexity function M_{cf} as:

$$M_{cf} = S_{pace} + T_{ime} = nmb + nb\log(t) + \frac{n^2}{b} \tag{23}$$

Then, the second-order partial mixed derivative of M_{cf} with respect to b can be calculated by

$$\frac{\partial^2 M_{cf}}{\partial b^2} = 2\frac{n^2}{b^3} \tag{24}$$

Since $\frac{\partial^2 M_{cf}}{\partial b^2} > 0$, the mixed complexity $O(S_{max})$, which is composed of the time complexity T_{ime} and the space complexity S_{pace}, is minimal when the number of intervals is $b = b_{best} = \left(\frac{n}{m+\log(t)}\right)^{\frac{1}{2}}$. Note that, if $b_{best} \ge \frac{n^2}{S_{max}}$, the space complexity of our algorithm will be lower than the space complexity threshold $O(S_{max})$ when we set the number of intervals $b = b_{best}$. This makes the mixed complexity M_{cf} minimum. If $b_{best} < \frac{n^2}{S_{max}}$, we will set the number of intervals $b = \frac{n^2}{S_{max}}$ to ensure that the space complexity

TABLE 1. The time complexities of various algorithms.

Algorithm	Time complexity
DOE-AND-SCA	$O(iter(3K^2(9K+2n)+n(t+mb+blog(t))))$
SCA	$O(n^3)$
NJWN	$O(nl+l^3+nlk+nK^2)$
STSC	$O(n^3+n^2\log(t))$
FDC	$O(2n^2+n(m+6))$

TABLE 2. The space complexities of various algorithms.

Algorithm	Space complexity
DOE-AND-SCA	$O(nb)$
SCA	$O(n^2)$
NJWN	$O(ln)$
STSC	$O(n^2)$
FDC	$O(n^2)$

S_{pace} of our algorithm is lower than the space complexity threshold $O(S_{max})$.

F. COMPLEXITY ANALYSIS

In this section, we provide a complexity analysis of DOE-AND-SCA algorithm. Suppose we randomly select n data points, each data has t DNNs on average, the number of intervals is b, the number of clusters is K and the number of iterations to select the optimal density difference is $iter$. Let's investigate each step.

1) To construct the sparse similarity matrix, we need to find the DNNs of each point. Thus, the time complexity of this step is $O(nmb + nb\log t)$.
2) The calculation of ρ and δ takes up most time complexity of the AND algorithm, and its time complexity is $O(nt)$.
3) The time complexity of selecting eigenvectors is composed of using *ARPACK* to obtain the first $2K$ eigenvectors of L and calculating the Laplace score. Thus, the time complexity of this step is $O(27K^3 + 6nK^2)$.
4) The time complexity of selecting the optimal density difference is related to the number of iterations $iter$.

The space complexity of the DOE-AND-SCA algorithm lies mainly on the space complexity required to calculate the interval distance matrix. So, its space complexity is $O(nb)$.

The steps of the DOE-AND-SCA algorithm is listed in **Algorithm 3**. By comparison, the time complexities and the space complexities of various algorithms are presented in TABLE 1 and TABLE 2, respectively.

Algorithm 3 The DOE-AND-SCA

Input: Input data set
Output: Output clustering result

Determine the number of intervals;
for *iter* **do**
Calculate the sparse similarity matrix based on DNNs;
Call the AND algorithm to determine the number of clusters;
Call the *OE* algorithm to select the appropriate eigenvectors;
Apply K-means to get the clusters;
Calculate the fitness value and update the density difference threshold;
return The clustering results with lowest Fitness value.

TABLE 3. The summary of real-world data sets.

Data Set	Instances	Dimensions	Classes
PenDigits	10992	16	10
Aggregation	788	2	7
BreastCancer	699	10	2
Jain	373	2	2
Dermatology	366	34	6
Spiral	312	2	3
Haberman	290	3	3
Flame	240	2	2
Seeds	210	7	3
Wine	178	13	3
Iris	150	4	3

TABLE 4. The summary of artificial multi-scale data sets.

Data Set	Instances	Dimensions	Classes
ZM3	303	2	3
ZM5	622	2	5
ZM9	238	2	3

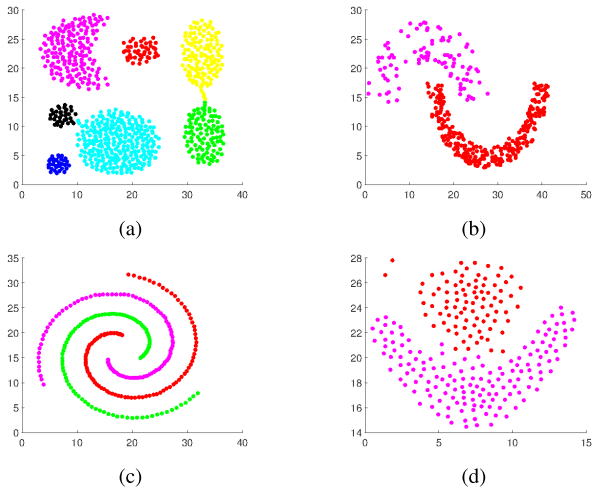


FIGURE 4. The clustering results of the DOE-AND-SCA on: (a) Aggregation, (b) Jain, (c) Spiral, and (d) Flame.

IV. EXPERIMENTS

The operating system for experiments is Windows 7, the integrated development environment is Matlab2012a, CPU is Intel Core I5 2.5GHz and the memory is 4GB. The maximum number of elements allowed in a matrix on this version of MATLAB is 2.1475×10^9. So, we can set the space complexity threshold $O(S_{max}) = 10^9$.

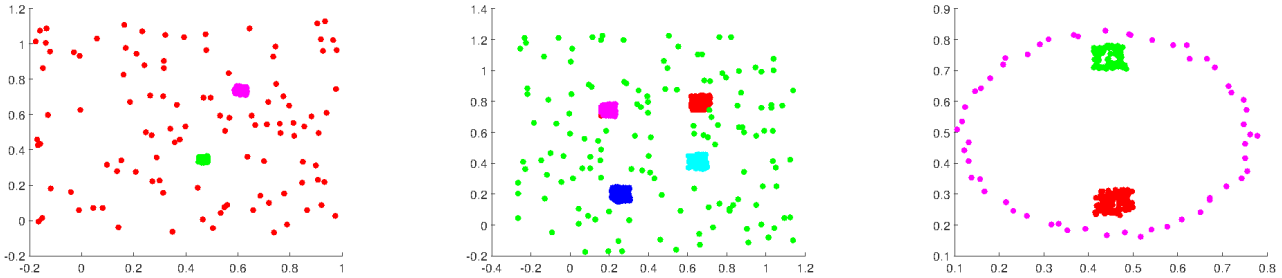


FIGURE 5. The clustering results of the D-AND-SCA algorithm on: (a) ZM3, (b) ZM5, and (c) ZM9.

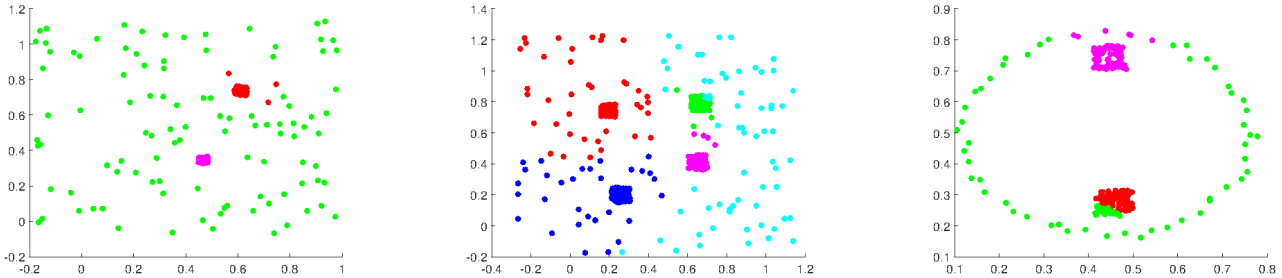


FIGURE 6. The clustering results of the STSC algorithm on: (a) ZM3, (b) ZM5, and (c) ZM9.

Simulations are carried out on 11 real-world data sets from UCI database and their learning libraries. TABLE 3 provides a summary description about these real-world data sets. TABLE 4 shows three 2-D multi-scale data sets introduced in Zelnik-Manor.[1] These artificial data sets are challenging and selected here due to their multiple scales.

Clustering accuracy and *clustering purity* are adopted to evaluate the algorithm's performance. Clustering accuracy [40] is defined as

$$Accuracy = \frac{\sum_{i=1}^{K} a_i}{n} \times 100 \tag{25}$$

where a_i is the number of data points which have been correctly classified to cluster C_i, K is the number of clusters, n is the number of total data points; while the clustering purity is defined as

$$Purity = \sum_{i=1}^{K} \frac{|P_i^d|}{K|P_i|} \times 100 \tag{26}$$

where $|P_i^d|$ is the number of data points which have been correctly classified, $|P_i|$ represents the total number of data points in cluster i.

Several two-dimensional artificial data sets, including Aggregation, Jain, Spiral and Flame, are used to demonstrate the performance of our DOE-AND-SCA algorithm. We find that, indeed, DOE-AND-SCA can achieve expected clustering results for the data sets of various shapes, as shown in FIGURE 4. Moreover, since several new operations were added in the DOE-AND-SCA algorithm, we would like to investigate their contributions one by one by simulations.

[1] http://www.vision.caltech.edu/lihi/Demos/SelfTuningClustering.html

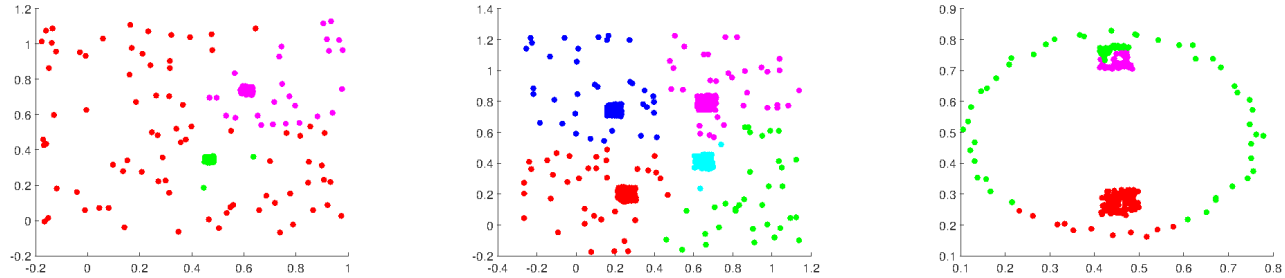
A. NUMERICAL ANALYSIS OF SIMILARITY FUNCTION

Here, we use the multi-scale data sets ZM3, ZM5, ZM9. Comparing the clustering results obtained by D-AND-SCA (the simplified version of DOE-AND-SCA, which only select the K eigenvectors with the largest eigenvalues), STSC, and SCA, as shown in FIGUREs 5-7, we find that D-AND-SCA with similarity function based on the DNNs can achieve relatively better clustering results.

B. NUMERICAL ANALYSIS OF AND

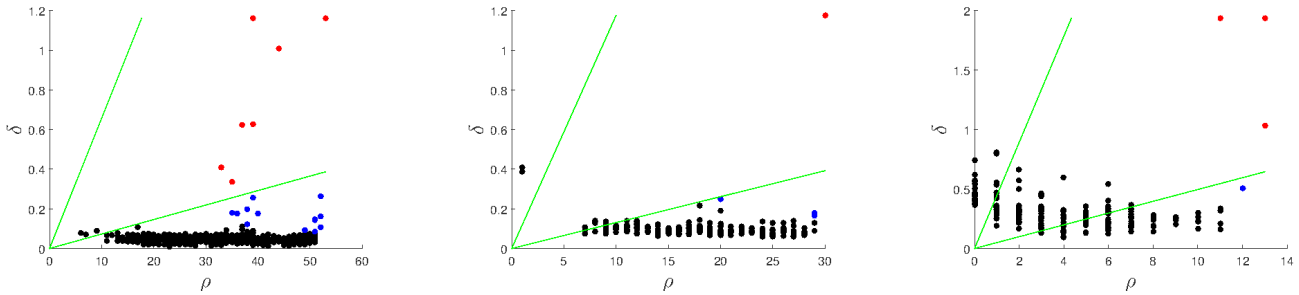
Automatic cluster number determination plays an important role in DOE-AND-SCA. Here, we will testify how it works. FIGUREs 8-10 show the overall process to determine the number of clusters. We set filter ratio $\omega = 3$ here, because through the analysis of some experimental data sets, we find when $\omega = 3$, we can obtain the true cluster centers with higher probability. The specific experimental steps are:

- First, plot the $\rho - \delta$ plane and the density distribution of γ, as shown in FIGUREs 8 and 9.
- Second, obtain the normal distribution curves fitted according to the density distribution of γ and get the confidence interval, as shown in FIGURE 10.



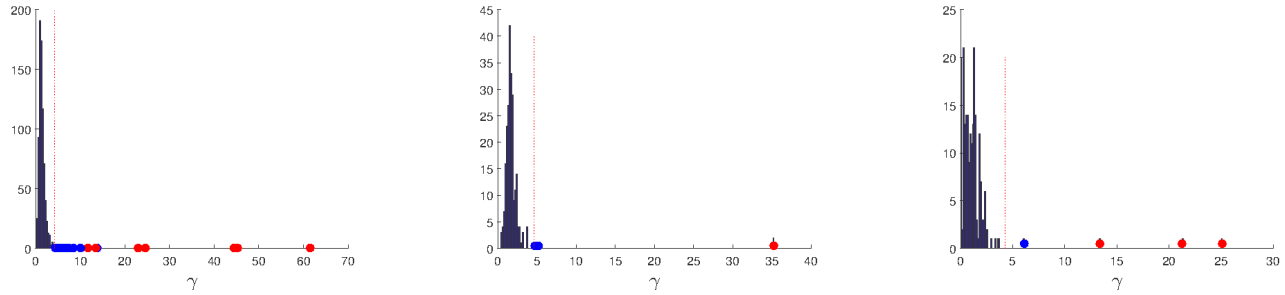
(a) (b) (c)

FIGURE 7. The clustering results of the SCA algorithm on:(a) ZM3, (b) ZM5, and (c) ZM9.



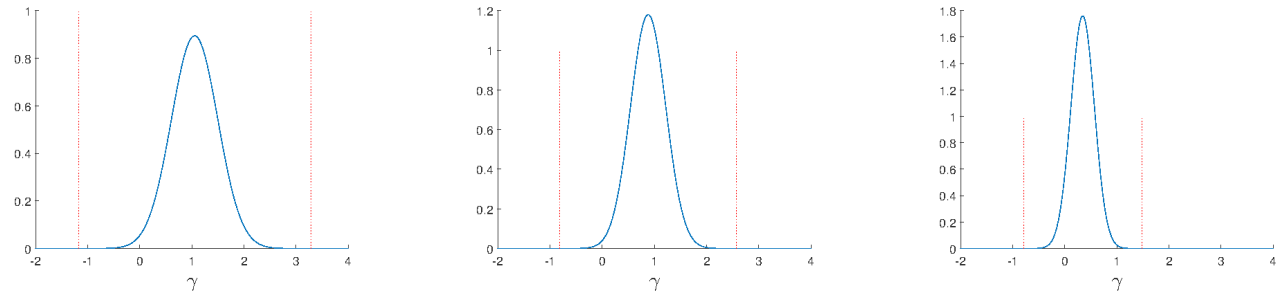
(a) (b) (c)

FIGURE 8. The $\rho - \delta$ planes for: (a) Aggregation, (b) Flame, and (c) Seeds.



(a) (b) (c)

FIGURE 9. The density distribution of γ for: (a) Aggregation, (b) Flame, and (c) Seeds.



(a) (b) (c)

FIGURE 10. The normal distribution curve about γ for: (a) Aggregation, (b) Flame, and (c) Seeds.

- Third, find out the data points falling outside the confidence interval as the singular points in the density distribution of γ.
- Finally, use the screening method to filter out selected singular points, and get the cluster centers.

TABLE 5. Comparison of the clustering accuracy of various algorithm for different data sets (%).

$DataSet$	$DOE-AND-SCA$	SCA	$NJWN$	$STSC$	FDC
$PenDigits$	**81.52**	76.55	73.94	78.65	80.26
$Aggregation$	**99.62**	92.34	71.07	98.61	96.26
$BreastCancer$	**99.86**	89.32	95.99	96.99	96.84
$Jain$	**100**	80.21	88.47	86.33	**100**
$Dermatology$	**93.44**	86.21	88.25	90.98	89.54
$Spiral$	**100**	80.24	41.35	88.14	**100**
$Haberman$	**100**	86.41	75.51	82.41	77.59
$Flame$	**99.17**	90.55	85.83	98.33	**100**
$Seeds$	**93.81**	89.21	91.90	90.95	90.00
$Wines$	**96.63**	71.79	96.07	94.94	95.86
$Iris$	**96.67**	92.67	93.33	94.00	94.67

TABLE 6. Comparison of the clustering purity of various algorithm for different data sets (%).

$DataSet$	$DOE-AND-SCA$	SCA	$NJWN$	$STSC$	FDC
$PenDigits$	**80.25**	77.52	74.26	77.85	79.64
$Aggregation$	**99.85**	92.68	72.34	98.83	98.52
$BreastCancer$	**99.74**	90.42	96.21	96.86	95.26
$Jain$	**100**	82.57	86.26	88.62	**100**
$Dermatology$	**94.25**	88.26	90.45	94.98	90.34
$Spiral$	**100**	82.96	50.25	89.26	**100**
$Haberman$	**100**	85.42	77.58	86.44	81.25
$Flame$	**99.20**	92.72	86.18	99.12	**100**
$Seeds$	**94.12**	90.24	92.56	90.15	92.26
$Wines$	**96.83**	74.62	96.26	95.84	95.82
$Iris$	**96.97**	94.74	94.06	94.25	95.52

In FIGUREs 8-10, the green lines are used to filter the ineligible singular; the red dotted lines represent the confidence intervals; the blue data points are the ineligible singular points; and the red data points are the final cluster centers.

C. NUMERICAL ANALYSIS OF OE

In order to investigate the contribution of OE, we compare the clustering results of various algorithms, including D-AND-SCA, DOE-AND-SCA, STSC algorithm, OE-STSC algorithm (the STSC with optimal eigenvector selection), NJWN algorithm, OE-NJWN algorithm (the NJWN with optimal eigenvector selection), as shown in FIGURE 11.

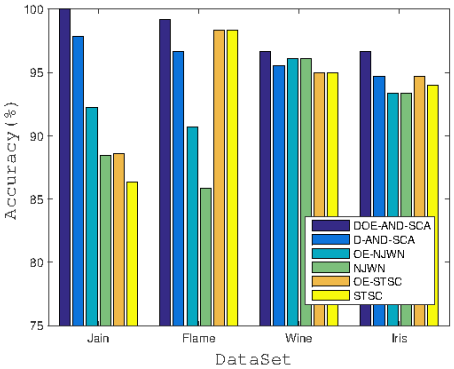


FIGURE 11. Comparison of the three algorithms with and without the OE operation on clustering accuracies.

We can find that, in most cases, the algorithms with OE operation have higher clustering accuracy. This is because the eigenvectors selected by the OE algorithm can better capture the structure of the data set.

D. NUMERICAL ANALYSIS OF OPTIMAL DENSITY DIFFERENCE THRESHOLD

For a given data set, the density difference between nearest-neighbors is relatively small, as described in [3] and [14]. FIGURE 12 shows the relationships between fitness values, clustering accuracy, and density difference threshold. General, the fitness values and the clustering accuracy have the opposite trend with density difference threshold, and the clustering accuracy is maximized when the fitness function takes the minimum value.

E. PARAMETER SENSITIVITY ANALYSIS

Here, we present how the number of data points in the initial nearest-neighbor set affect the algorithm performance. Here, the number of data points in the initial nearest-neighbor set is the maximum number of data point in the DNN set for each data point. The results are shown in FIGURE 13. We find that the clustering accuracy increases first and then decreases with the number of the initial nearest-neighbors for all the four considered data sets. This is reasonable because too small number of the initial nearest-neighbors may cause that the DNN set could not fully express its local characteristics, while too large number of the initial nearest-neighbors may let DNN set contain many data originally belong to other clusters, both of which may lead to poor clustering accuracy.

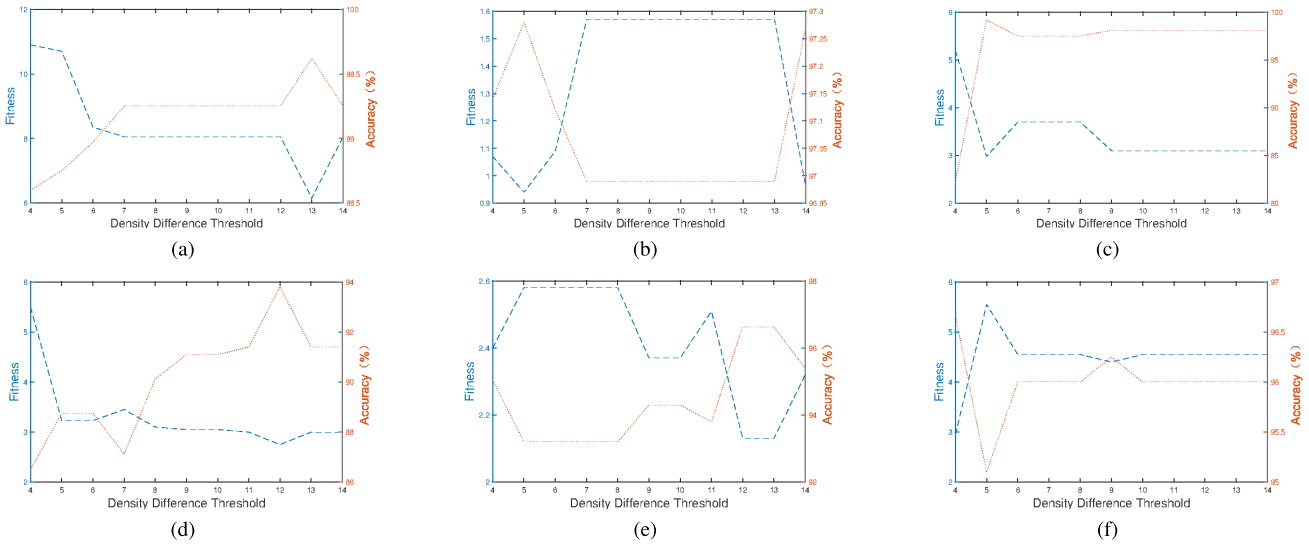


FIGURE 12. The relationships between fitness values, clustering accuracy, and density difference threshold for: (a) Aggregation, (b) Breast Cancer, (c) Flame, (d) Seeds, (e) Wine, and (f) Iris.

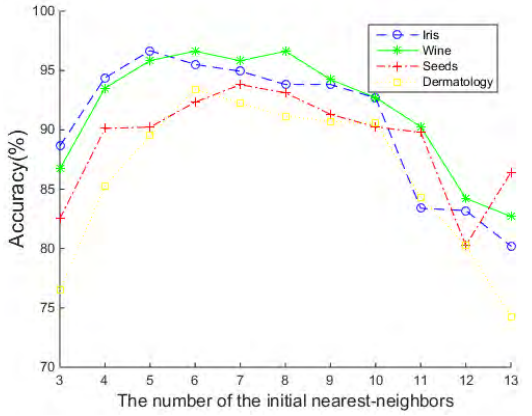


FIGURE 13. The relationship between the clustering accuracy and the number of initial nearest-neighbors.

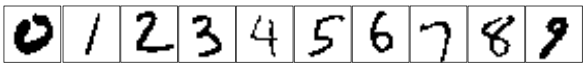


FIGURE 14. Examples of handwritten image.

FIGURE 15. Examples of Frontal face image.

F. PERFORMANCE COMPARISONS OF SCAS

In this section, we focus on the following data sets: PenDigits, Aggregation, BreastCancer, Jain, Dermatology, Spiral, Haberman, Flame, Seeds, Wine, and Iris. DOE-AND-SCA is compared with SCA, NJWN, STSC and FDC in clustering accuracy and clustering purity, as shown in the TABLE 5 and the TABLE 6.

TABLE 5 gives the simulation results carried out on eleven data sets about the clustering accuracy. Clustering accuracies of DOE-AND-SCA are higher than those of other algorithms, i.e., DOE-AND-SCA, SCA, NJWN, STSC and FDC give average clustering accuracy of 96.43%, 85.05%, 81.92%, 90.94% and 92.82% when dealing with these data sets. DOE-AND-SCA is more 14.51% and 3.61% accurate than the NJWN and FDC algorithm, respectively. As shown in TABLE 6, there are the experimental results carried out on eleven data sets about the clustering purity. The average clustering purity of these comparison algorithms are 96.47%, 86.56%, 83.31%, 92.02% and 93.51%. DOE-AND-SCA is more 13.16% and 2.96% accurate than the NJWN and FDC algorithm, respectively.

The reason that our algorithm has better clustering effect is as follows: (1) Similarity function based on DNN can more rationally divide the boundary points. (2) The eigenvectors selected by the optimal eigenvector selection algorithm can better express the data structure of the data set.

V. APPLICATIONS

Supervised methods have gained great success in image recognition area, especially CNN and its improved versions. In practical applications, however, a large number of images are created by minutes, and most of them are not labelled. How to identify those images without labels? Unsupervised method such as clustering may provide an efficient solution. Here, we use our DOE-AND-SCA to cluster images to improve recognition rate.

A. EXPERIMENTAL DATA SET

MNIST data set is consisted of 70000 number handwriting images. Randomly select 100 handwritten pictures for

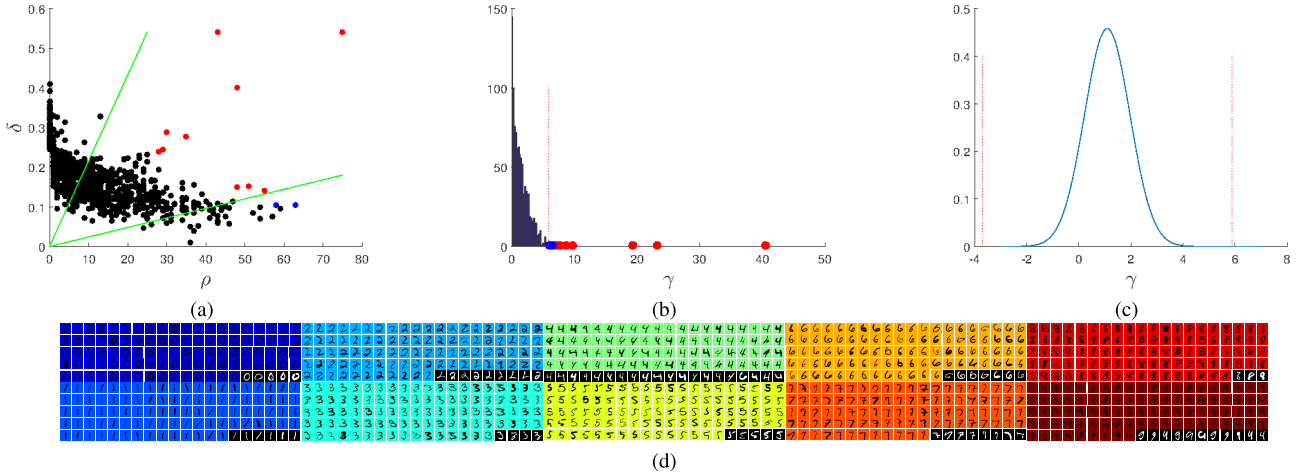


FIGURE 16. Clustering process of MNIST data set: (a) $\rho - \delta$ decision graph (b) Density distribution of γ (c) Normal distribution curves (d) Clustering result of MNIST data set (The same color images belong to the same cluster, the black image represents misplaced images).

each number to form 1000 handwritten images data set. The selected MNIST image for each number is shown in FIGURE 14.

B. IMAGE RECOGNITION BASED ON DOE-AND-SCA

Since the image similarity cannot be measured by the ordinary distance, SSIM [41] is used for calculating the relative distance between images. The specific steps for image recognition based on DOE-AND-SCA are shown as **Algorithm 4**.

Algorithm 4 Image Recognition Based on DOE-AND-SCA

Input: Input image data set
Output: Output clustering result
Determine the number of intervals;
for *iter* **do**
Calculate interval sparse distance matrix based on SSIM algorithm;
Calculate the sparse similarity matrix based on DNNs;
Call the AND algorithm to determine the number of clusters;
Use the optimal eigenvector selection algorithm to select appropriate eigenvectors;
Apply K-means to get the clusters;
Calculate the Fitness value and update density difference threshold;
return The clustering results with lowest Fitness value.

C. CLUSTERING RESULT ANALYSIS

The experimental results of those main steps for MNIST data set and Frontal face data set clustering based on DOE-AND-SCA are shown in FIGURE 16 and FIGURE 17, respectively.

Frontal face data set is collected by Markus Weber at California Institute of Technology. We select 10 people, with each having 15 face images.

TABLE 7. Comparison of the clustering accuracy of various algorithms on MNIST data set and Frontal face data set (%).

Algorithm	*MNIST*	*Frontal*
$DOE-AND-SCA$	**92.20**	**92.00**
SCA	67.26	71.20
$NJWN$	53.70	76.25
$STSC$	83.78	78.90
FDC	87.75	86.24

TABLE 8. Comparison of the clustering purity of various algorithms on MNIST data set and Frontal face data set (%).

Algorithm	*MNIST*	*Frontal*
$DOE-AND-SCA$	**91.40**	**92.60**
SCA	69.64	73.40
$NJWN$	54.60	78.75
$STSC$	82.78	79.82
FDC	89.65	88.85

As shown in FIGURE 16 (a)-(c), ten cluster centers are automatically selected by DOE-AND-SCA. And the experimental results show that the ten clustering centers represent ten images of different numbers, respectively, which testify that DOE-AND-SCA can accurately get the number of clusters automatically. FIGURE 16 (d) shows the experimental results of DOE-AND-SCA for MNIST data set. The clustering accuracy of DOE-AND-SCA is 92.2% and clustering purity is 91.4%, better than the other methods, as presented in the TABLE 7 and the TABLE 8, respectively.

As shown in FIGUREs 17 (a)-(c), ten clusters are also automatically selected by DOE-AND-SCA for the Frontal face data set. And the experimental results show that the ten clustering centers represent ten images of different people, respectively. This time, the clustering accuracy of DOE-AND-SCA is 92% and clustering purity is 92.6%, also

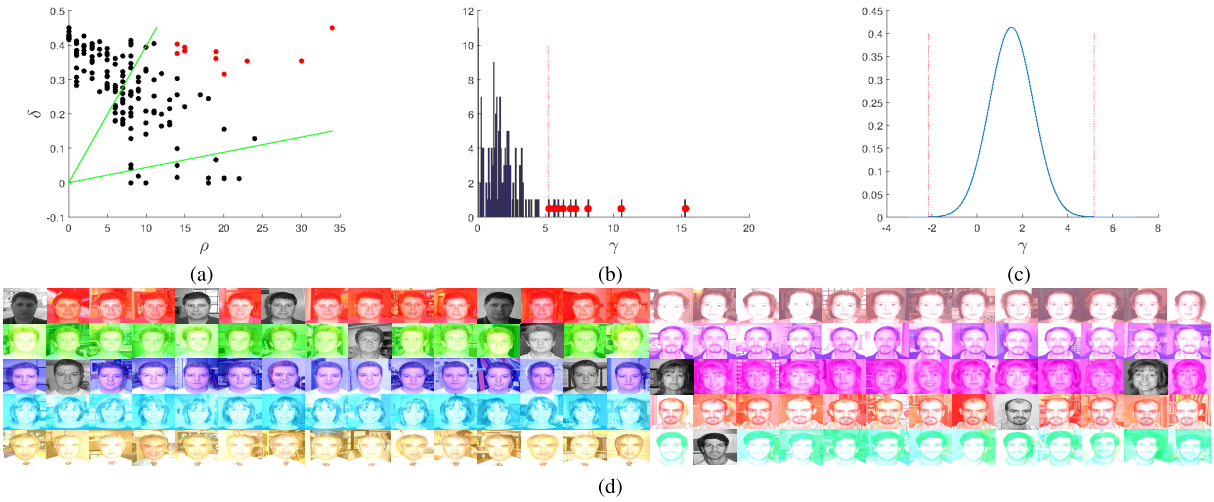


FIGURE 17. Clustering process of Frontal data set: (a) $\rho - \delta$ decision graph (b) Density distribution of γ (c) Normal distribution curves (d) Clustering result of Frontal data set (The same color images belong to the same cluster, the gray image represents misplaced images).

better than the other methods, as presented in the TABLE 7 and the TABLE 8, respectively.

VI. CONCLUSION

Most SCAs are confronted with several challenges, such as multi-scale data, the determination of the number of clusters, appropriate selection of eigenvectors, and parameter sensibility for practical applications. In this paper, we put forward three mechanisms to solve these problems and proposed a novel clustering method, namely DOE-AND-SCA. This method can handle multi-scale data, determine the number of clusters automatically, select optimal eigenvectors, and adjust parameters self-adaptively. Abundant of simulations and application experiments are carried out to testify its performances. The results show that our DOE-AND-SCA behaves better, by comparing with other excellent SCAs.

However, the DOE-AND-SCA still has some limitations, for example, the time complexity of the DOE-AND-SCA is relatively high. In order to reduce the time complexity of the DOE-AND-SCA, we will consider reducing the time complexity of the algorithm from following two parts: determining the cluster centers and selecting the optimal eigenvectors. And we will apply the proposed algorithm to other real-world data sets.

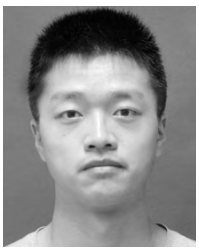
REFERENCES

[1] U. von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.
[2] H.-C. Huang, Y.-Y. Chuang, and C.-S. Chen, "Affinity aggregation for spectral clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 773–780.
[3] Y. Song, W. Y. Chen, H. Bai, C. J. Lin, and E. Y. Chang, "Parallel spectral clustering," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2008, pp. 374–389.
[4] Q. Liu, Z. Dong, and E. Wang, "Moment-based spectral analysis of large-scale generalized random graphs," *IEEE Access*, vol. 5, pp. 9453–9463, 2017.
[5] F. Bu, "A high-order clustering algorithm based on dropout deep learning for heterogeneous data in cyber-physical-social systems," *IEEE Access*, vol. 6, pp. 11687–11693, 2017.
[6] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Int. Conf. Neural Inf. Process. Syst., Natural Synth.*, 2001, pp. 849–856.
[7] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.
[8] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the Nyström method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 214–225, Feb. 2004.
[9] F. Zhao, H. Liu, and L. Jiao, "Spectral clustering with fuzzy similarity measure," *Digital Signal Processing*, vol. 21, no. 6, pp. 701–709, Jun. 2011.
[10] W. Chongjun, L. W. Jun, D. Lin, T. Juan, and C. Shifu, "Image segmentation using spectral clustering," *Proc. IEEE Int. Conf. Tools Artif. Intell.*, Nov. 2005, p. 678.
[11] X. Chen and D. Cai, "Large scale spectral clustering with landmark-based representation," in *Proc. AAAI Conf. Artif. Intell.*, 2011, pp. 313–318.
[12] P. Qian, F.-L. Chung, S. Wang, and Z. Deng, "Fast graph-based relaxed clustering for large data sets using minimal enclosing ball," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 3, pp. 672–687, Jun. 2012.
[13] R. Xia, Y. Pan, L. Du, and J. Yin, "Robust multi-view spectral clustering via low-rank and sparse decomposition," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 2149–2155.
[14] N. Tremblay, G. Puy, R. Gribonval, and P. Vandergheynst, "Compressive spectral clustering," in *Proc. ICML*, 2016, pp. 1002–1011.
[15] N. Passalis and A. Tefas, "Spectral clustering using optimized bag-of-features," in *Proc. Hellenic Conf. Artif. Intell.*, 2016, p. 19.
[16] P. Yang, Q. Zhu, and B. Huang, "Spectral clustering with density sensitive similarity function," *Knowl.-Based Syst.*, vol. 24, no. 5, pp. 621–628, 2011.
[17] X. Zhang, J. Li, and H. Yu, "Local density adaptive similarity measurement for spectral clustering," *Pattern Recognit. Lett.*, vol. 32, no. 2, pp. 352–358, 2011.
[18] M. Beauchemin, "A density-based similarity matrix construction for spectral clustering," *Neurocomputing*, vol. 151, no. 151, pp. 835–844, 2015.
[19] R. Jenssen, D. Erdogmus, J. C. Principe, and T. Eltoft, "The Laplacian classifier," *IEEE Trans. Signal Process.*, vol. 55, no. 7, pp. 3262–3271, Jul. 2007.
[20] Z. Wang, "Determining the clustering centers by slope difference distribution," *IEEE Access*, vol. 5, pp. 10995–11002, 2017.
[21] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 1601–1608.
[22] A. Mur, R. Dormido, N. Duro, J. Vega, and S. Dormido-Canto, "Determination of the optimal number of clusters using a spectral clustering optimization," *Expert Syst. Appl.*, vol. 65, pp. 304–314, Dec. 2016.

[23] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, Nov. 1987.
[24] G. Wacquet, M. Poisson-Caillault, and P.-A. Hébert, *Semi-Supervised K-Way Spectral Clustering with Determination of Number of Clusters*. Berlin, Germany: Springer, 2013.
[25] S. Borjigin, "Non-unique cluster numbers determination methods based on stability in spectral clustering," *Knowl. Inf. Syst.*, vol. 36, no. 2, pp. 439–458, 2013.
[26] M. Gu, H. Zha, C. Ding, X. He, and H. Simon, "Spectral relaxation models and structure analysis for k-way graph clustering and bi-clustering," Dept. Comput. Sci. Eng., Pennsylvania State Univ., State College, PA, USA, Tech. Rep. CSE-01-007, 2001.
[27] T. Xiang and S. Gong, "Spectral clustering with eigenvector selection," *Pattern Recognit.*, vol. 41, no. 3, pp. 1012–1029, Mar. 2008.
[28] Y. Jiang and J. Ren, "Eigenvector sensitive feature selection for spectral clustering," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2011, pp. 114–129.
[29] M. Hosseini and F. T. Azar, "A new eigenvector selection strategy applied to develop spectral clustering," *Multidimensional Syst. Signal Process.*, vol. 28, no. 4, pp. 1227–1248, 2017.
[30] F. Zhao, L. Jiao, H. Liu, X. Gao, and M. Gong, "Spectral clustering with eigenvector selection based on entropy ranking," *Neurocomputing*, vol. 73, no. 1012, pp. 1704–1717, 2010.
[31] T. Nguyen, A. Khosravi, A. Bhatti, and D. Creighton, "Neural signal analysis by landmark-based spectral clustering with estimated number of clusters," in *Proc. Int. Joint Conf. Neural Netw.*, 2014, pp. 4042–4049.
[32] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Statist. Methodol.*, vol. 63, no. 2, pp. 411–423, 2004.
[33] E. Arias-Castro, G. Lerman, and T. Zhang, "Spectral clustering based on local PCA," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 253–309, 2017.
[34] K. Dan, M. Galun, and A. Brandt, "Fast multiscale clustering and manifold identification," *Pattern Recognit.*, vol. 39, no. 10, pp. 1876–1891, 2006.
[35] N. Tremblay, G. Puy, P. Borgnat, R. Gribonval, and P. Vandergheynst, "Accelerated spectral clustering using graph filtering of random signals," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2016, pp. 4094–4098.
[36] A. Rodriguez and A. Laio, "Machine learning. Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, p. 1492, 2014.
[37] R. B. Lehoucq, D. C. Sorensen, and C. Yang, "ARPACK users," *Guide: Solution of Large-Scale Eigenvalue Problems With Implicitly Restarted Arnoldi Methods*. Philadelphia, PA, USA: SIAM, 1998.
[38] F. R. Chung, *Spectral Graph Theory*. Providence, RI, USA: American Mathematical Society, 1997, p. 92.
[39] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 507–514.
[40] Z. Huang, "Clustering large data sets with mixed numeric and categorical values," in *Proc. 1st Pacific–Asia Conf. Knowl. Discovery Data Mining (PAKDD)*, Singapore, 1997, pp. 21–34.
[41] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

JINYIN CHEN is currently pursuing the Ph.D. degree with the Institute of Information Engineering, Zhejiang University of Technology. She is currently an Associate Professor with the Institute of Information Engineering, Zhejiang University of Technology. Her research interest covers intelligent computing, optimization, and data mining.

YANGYANG WU received the bachelor's degree from the Zhejiang University of Technology in 2016, where he is currently pursuing the master's degree with the Institute of Information Engineering. His research interest covers data mining and applications, and clustering analysis.



XIANG LIN received the bachelor's degree from the Zhejiang University of Technology, where he is currently pursuing the master's degree with the Institute of Information Engineering. His research interest covers data mining and applications, and clustering analysis.

QI XUAN received the B.S. and Ph.D. degrees in control theory and engineering from Zhejiang University, Hangzhou, China, in 2003 and 2008, respectively.

He was a Post-Doctoral Researcher with the Department of Information Science and Electronic Engineering, Zhejiang University, from 2008 to 2010, and also a Research Assistant with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong, in 2010 and 2017. From 2012 to 2014, he was a Post-Doctoral Fellow with the Department of Computer Science, University of California at Davis, Davis, CA, USA. He is currently a Professor with the College of Information Engineering, Zhejiang University of Technology, Hangzhou. His current research interests include network-based algorithm design, social network data mining, social synchronization and consensus, reaction-diffusion network dynamics, machine learning, and computer vision.