

Self-Tuning p -Spectral Clustering Based on Shared Nearest Neighbors

Hongjie Jia^{1,2} · Shifei Ding^{1,2} · Mingjing Du^{1,2}

Received: 20 September 2014 / Accepted: 20 April 2015 / Published online: 1 May 2015
© Springer Science+Business Media New York 2015

Abstract Cognitive computing needs to handle large amounts of data and information. Spectral clustering is a powerful data mining tool based on algebraic graph theory. Because of the solid theoretical foundation and good clustering performance, spectral clustering has aroused extensive attention of academia in recent years. Spectral clustering transforms the data clustering problem into the graph partitioning problem. Cheeger cut is an optimized graph partitioning criterion. To minimize the objective function of Cheeger cut, the eigen-decomposition of p -Laplacian matrix is required. However, the clustering results are sensitive to the selection of similarity measurement and the parameter p of p -Laplacian matrix. Therefore, we propose a self-tuning p -spectral clustering algorithm based on shared nearest neighbors (SNN-PSC). This algorithm uses shared nearest neighbors to measure the similarities of data couples and then applies fruit fly optimization algorithm to find the optimal parameters p of p -Laplacian matrix that leads to better data classification. Experiments show that SNN-PSC algorithm can produce more balanced clusters and has strong adaptability and robustness compared to traditional spectral clustering algorithms.

Keywords Spectral clustering · Cheeger cut · p -Laplacian · SNN · FOA

Introduction

Different from traditional computing technologies, cognitive computing tries to achieve more natural human-computer interaction by mimicking the cognitive process of human [1, 2]. In the future, machines should have the ability to learn and their intelligence degree will increase as they are being used. As a representative of the cognitive computing system, “Watson” computer developed by IBM can realize autonomous learning like human brain through real-time computing and analysis on large data [3]. So cognitive computing centers on data, not on the processor. How to dig useful information from complex data has become a new challenge for cognitive computing. Clustering is an effective unsupervised learning technique in data mining field. According to the characteristics of data or certain rules, clustering aims to classify data objects into different sets (or clusters) so that the objects in the same set have high similarities, while the objects from different sets have relatively low similarities [4]. Classifying the unlabeled data objects into different categories will help reveal the real structure of original haphazard data and provide assistance for further data analysis. In our daily life, clustering techniques have been widely used in many applications, such as business customer analysis, biology data processing, image segmentation, speech recognition and medical diagnosis. [5].

The application of clustering techniques to cognitive computing may help solve the imprecise and uncertain issues in biological systems [6, 7]. Traditional clustering algorithms, such as k -means algorithm, FCM algorithm and

✉ Shifei Ding
dingsf@cumt.edu.cn

¹ School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China

² Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

EM algorithm (Expectation Maximization), are based on convex data space. When the spatial distribution of points is not convex, these clustering algorithms may easily fall into local optima. So they are not suitable for the clustering problem on arbitrary shaped data space [8]. But in real applications, many data points are distributed in anomalous shapes. In order to solve this problem, many academic researchers began to use spectral method to deal with non-convex distributed data in recent years [9]. The main idea of spectral method is converting the clustering problem on data space into a graph partitioning problem on graph G , where G is an undirected weighted graph. In graph G , every point in the data set is as a vertex, and the similarity value between each couple of points is as the weight of the edge. Most spectral clustering algorithms have three steps: firstly, define an affinity matrix to describe the similarities of pairwise data objects according to the given data set; secondly, construct Laplacian matrix and calculate its eigenvalues and eigenvectors; thirdly, select appropriate eigenvectors to group n data objects into k clusters.

In spectral clustering, the definition of graph cut criteria is crucial. Graph cut criteria are the principle of how to divide a weighted graph into several parts to make the objective function reach the optimal solution. Its ultimate goal is to maximize the connection weights within each sub-graph, while minimize the connection weights between sub-graphs [10]. The selection of graph cut criteria has a direct impact on the quality of clustering results. Minimum cut, ratio cut, normalized cut and average cut are classic graph partition criteria [11]. However, seeking the optimal solution of graph cut criteria is an NP-hard problem. Fortunately, spectral method can provide a loose solution within polynomial time for these optimization problems. Here, “loose” means relax the discrete optimization problem to the real number field, and then using some heuristic approach to re-convert it to a discrete solution [12]. The essence of graph partitioning can be summarized as the minimization or maximization problem of matrix trace. The completion of these minimizing or maximizing tasks relies on the spectral decomposition of graph Laplacian matrix.

Recently, an improved version of normalized cut named Cheeger cut has aroused much attention [13]. Research shows that Cheeger cut is able to produce more balanced clusters by calculating the second eigenvector of p -Laplacian matrix [14]. p -Laplacian matrix is a nonlinear generalization form of graph Laplacian matrix. However, during the clustering process, it is hard to build a high-quality similarity matrix that can properly describe the intrinsic relationship between data points; besides, the parameter p of p -Laplacian matrix is sensitive and often requires empirical knowledge to determine its value. In order to solve these problems, this paper defines a novel similarity

measurement that takes advantage of the local density information embedded in shared nearest neighbors (SNN), then uses fruit fly optimization algorithm (FOA) to automatically determine the optimum value of parameter p , and proposes a self-tuning p -spectral clustering algorithm based on shared nearest neighbors (SNN-PSC).

This paper is organized as follows: Section 2 reviews typical graph cut criteria and describes the concepts about p -Laplacian matrix and then analyzes the relationship between p -Laplacian and Cheeger cut; Section 3 introduces SNN similarity measure and fruit fly optimization method to improve the performance of p -spectral clustering and presents the specific scheme of SNN-PSC algorithm; Section 4 verifies the effectiveness of SNN-PSC algorithm through the experiments on both artificial data sets and real-world data sets; the final part is a summary of this paper and discusses some future research prospects.

Materials

Graph Cut Criteria

The idea of spectral clustering comes from spectral graph partition theory. Donath and Hoffman established a link between graph partitioning problem and the eigenvectors of similarity matrix. Fiedler proved that the bipartition of a graph is closely related to the second eigenvector of Laplacian matrix [15]. According to Rayleigh–Ritz theory, we can use the eigenvectors of Laplacian matrix to approximate the optimal solution of graph cuts [16].

The definition of graph cut criteria aims to find an optimal partition that the similarities within a cluster are as large as possible and the similarities between clusters are as small as possible. Given a data set, we can construct an undirected weighted graph $G = (V, E)$, where V is the set of vertices represented by data points, E is the set of edges weighted by the similarities between the edge’s two vertices. Then the simplest graph partitioning problem based on graph G is described as follows:

$$\text{cut}(A_1, \dots, A_k) = \sum_{i=1}^k \text{cut}(A_i, \bar{A}_i) \quad (1)$$

where $\text{cut}(A, B) = \sum_{i \in A, j \in B} w_{ij}$, $\bar{A}_i = \{v_p | v_p \in V \text{ and } v_p \notin A_i\}$ is the complementary set of sub-graph A_i . A good graph partition is to minimize objective function (1), but this objective function only considers the external connections of clusters, neglecting the density distribution within each cluster. In addition, it is sensitive to the impact of outliers in the data set, which will result in unbalanced data partition.

In order to get more balanced clusters, Hagen and Kahng presented radio cut criterion, denoted as R_{cut} [17]:

$$\text{Rcut}(A_1, \dots, A_k) = \sum_{i=1}^k \frac{\text{cut}(A_i, \bar{A}_i)}{|A_i|} \quad (2)$$

Rcut objective function introduces cluster size as a balancing item to minimize the similarity between clusters, which can avoid unbalanced partition to some extent. But more vertices are not a necessary and sufficient condition of large similarity within clusters, so Shi and Malik proposed normalized cut, denoted as Ncut [18]:

$$\text{Ncut}(A_1, \dots, A_k) = \sum_{i=1}^k \frac{\text{cut}(A_i, \bar{A}_i)}{\text{vol}(A_i)} \quad (3)$$

where $\text{vol}(A_i) = \sum_{p \in A_i, q \in V} w_{pq}$. Minimizing the objective function means minimizing the total weights between sub-graphs and maximizing the inner weights of each sub-graph at the same time, so it is able to generate better classification results. However, as a nonlinear function of the projection matrix, this objective function may fall into local minimum.

Cheeger et al. [19] improved normalized cut and proposed Cheeger cut, denoted as Ccut:

$$\text{Ccut}(A_1, \dots, A_k) = \sum_{i=1}^k \frac{\text{cut}(A_i, \bar{A}_i)}{\min(A_i)} \quad (4)$$

The objective function of Ccut wants to make the data within the cluster be more compact, which means that the correlation between the cluster size (the number of data points in a cluster) and the total cut weight of the cluster is as small as possible after clustering. But according to the Rayleigh quotient principle, calculating the optimal Cheeger cut is an NP-hard problem. Next we will try to get an approximate solution of Cheeger cut by introducing p -Laplacian into spectral clustering.

Graph p -Laplacian

Spectral graph theory is inextricably linked with graph Laplacian matrix. Rayleigh quotient principle indicates that the optimization of graph cut problem is equivalent to calculating the eigenvalues of Laplacian matrix, which provides a relaxed solution of the theoretically optimal graph partition. Hein et al. [20] defined the inner product form of standard graph Laplacian Δ_2 as follows:

$$\langle f, \Delta_2 f \rangle_i = \frac{1}{2} \sum_{i,j=1}^n w_{ij} (f_i - f_j)^2 \quad (5)$$

where f is the eigenvector of Laplacian matrix. Assuming the Laplacian operator is generalized to Δ_p , where $p \in (1, 2]$, then Δ_p can be denoted as:

$$\langle f, \Delta_p f \rangle_i = \frac{1}{2} \sum_{i,j=1}^n w_{ij} (f_i - f_j)^p \quad (6)$$

Laplacian matrix L has un-normalized form and normalized form. Un-normalized Laplacian matrix is represented as $L = D - W$, and the normalized form is represented as $L = I - D^{-1}W$, where W is the affinity matrix, D is a diagonal matrix. It is not difficult to derive the corresponding un-normalized and normalized form of p -Laplacian matrix:

$$(\Delta_p^{(u)} f)_i = \sum_{j \in V} w_{ij} \varphi_p(f_i - f_j) \quad (7)$$

$$(\Delta_p^{(n)} f)_i = \frac{1}{d_i} \sum_{j \in V} w_{ij} \varphi_p(f_i - f_j) \quad (8)$$

where $d_i = \sum_{j=1}^n w_{ij}$ is the degree function of graph, $\varphi_p(x) = |x|^{p-1} \text{sign}(x)$, and when $p = 2$, $\varphi_2(x) = x$. Taking un-normalized p -Laplacian matrix, for example, next we give the definition of p -Laplacian's eigenvalue λ_p .

Definition 1 If there is a real number λ_p satisfying formula (9), λ_p is the corresponding eigenvalue of eigenvector f .

$$(\Delta_p^{(u)} f)_i = \lambda_p \varphi_p(f_i), \quad \forall i = 1, \dots, n \quad (9)$$

In matrix operation, the eigenvector corresponding to the smallest eigenvalue is expected to contain important discrimination information. This property is also play an important role for clustering algorithm. Related studies have shown that [21], in linear operation, if λ is the smallest eigenvalue of an operator Δ , λ should satisfy formula (10) to reach the lower bound.

$$\lambda = \arg \min_{f \in R^n} \frac{\langle f, \Delta f \rangle}{\|f\|^2} \quad (10)$$

where $\|f\|^2 = \sum_{i=1}^n |f_i|^2$. Definition 2 applies this feature to nonlinear p -Laplacian operator.

Definition 2 If λ_p is the smallest eigenvalue of p -Laplacian operator Δ_p , λ_p should satisfy formula (11) to reach the lower bound.

$$\lambda_p = \arg \min_{f \in R^n} \frac{\langle f, \Delta_p f \rangle}{\|f\|^p} \quad (11)$$

According to formula (11), the objective function of un-normalized p -Laplacian operator is constructed as:

$$F_p(f) = \frac{\langle f, \Delta_p^{(u)} f \rangle}{\|f\|^p} = \frac{\frac{1}{2} \sum_{i,j=1}^n w_{ij} |f_i - f_j|^p}{\|f\|^p} \quad (12)$$

The above analysis indicates that the lower bound value of $F_p(f)$ is related to the eigenvalues and eigenvectors of Δ_p . As the category information of data points is implicit in the eigenvectors of p -Laplacian matrix, the eigen-decomposition of p -Laplacian matrix is an essential part for spectral clustering. Similarly, the objective function of normalized p -Laplacian matrix can be obtained in the same way.

The Relationship Between p -Laplacian and Cheeger Cut

Cheeger cut can enhance the relationships within clusters and weaken the relationships between clusters, so it is able to produce more balanced sub-graphs. Since seeking the optimal solution of Cheeger cut is an NP-hard problem, we introduce p -Laplacian operator to make the original problem be solved in polynomial time. Related researches give evidence that the graph can be recursively divided into two parts using the second eigenvector $v_p^{(2)}$ of p -Laplacian matrix until the number of sub-graphs meets the requirements.

Theorem 1 For a segmentation problem (A_i, \bar{A}_i) , if the objective function of Cheeger cut is $Ccut(A_1, \dots, A_k) = \sum_{i=1}^k \frac{cut(A_i, \bar{A}_i)}{\min(A_i)}$ and the objective function of p -Laplacian matrix is $F_p(f)$, then the solution of $F_p(f)$ is approximate to the relaxed solution of Cheeger cut.

Proof First define an associated partition (f, A_i) , expressed as:

$$(f, A_i) = \begin{cases} \frac{1}{|A_i|^{p-1}} & i \in A_i, p \in (1, 2] \\ \frac{1}{|\bar{A}_i|^{p-1}} & i \in \bar{A}_i, p \in (1, 2] \end{cases} \quad (13)$$

In order to obtain the minimum eigenvalue of objective function, we use formula (13) to transform formula (12) into the following expression:

$$F_p(f, A_i) = \frac{\sum_{ij} w_{ij} \left| \frac{1}{|A_i|^{p-1}} - \frac{1}{|\bar{A}_i|^{p-1}} \right|^p}{\|f, A\|_p^p} \quad (14)$$

where $\|f, A\|_p^p = \sum_i |f_i, A|^p = \frac{1}{|A_i|^{p-1}} - \frac{1}{|\bar{A}_i|^{p-1}}$, then formula (14) can be converted to

$$\begin{aligned} F_p(f, A_i) &= \frac{\sum_{ij} w_{ij} \left| \frac{1}{|A_i|^{p-1}} - \frac{1}{|\bar{A}_i|^{p-1}} \right|^p}{\left| \frac{1}{|A_i|^{p-1}} - \frac{1}{|\bar{A}_i|^{p-1}} \right|^p} = \sum_{ij} w_{ij} \left| \frac{1}{|A_i|^{p-1}} - \frac{1}{|\bar{A}_i|^{p-1}} \right|^{p-1} \\ &\leq \sum_{ij} w_{ij} \left| \frac{2}{\min\{|A_i|, |\bar{A}_i|\}} \right|^{p-1} = 2^{p-1} \frac{cut(A_i, \bar{A}_i)}{\min\{|A_i|, |\bar{A}_i|\}} \end{aligned}$$

Comparing the above inequality with the objective function of Cheeger cut, we can see that

$$\lim_{p \rightarrow 1} F_p(f, A_i) = Ccut(A_i, \bar{A}_i) \quad (15)$$

So the solution of $F_p(f)$ is a relaxed approximate solution of Cheeger cut. Thus the optimal minimized objective function can be expressed as:

$$\lambda_p = \arg \min_{p \rightarrow 1} F_p(f) \quad (16)$$

where λ_p is the eigenvalue corresponding to eigenvector f .

The foregoing analysis shows that the problem of Cheeger cut can be solved by constructing p -Laplacian matrix and calculating its eigenvectors. Specifically, the second eigenvector $v_p^{(2)}$ of p -Laplacian matrix will lead to a bipartition of the graph by setting an appropriate threshold [8]. The optimal threshold value should minimize its corresponding Cheeger cut.

Un-normalized graph p -Laplacian matrix $\Delta_p^{(u)}$ corresponds to ratio Cheeger cut $RCC(C, \bar{C}) = \frac{cut(C, \bar{C})}{\min\{|C|, |\bar{C}|\}}$, whose second eigenvector $v_p^{(2)}$ should satisfy formula (17):

$$\begin{aligned} &\arg \min_{A_i = \{i \in V | v_p^{(2)}(i) > \tau\}} RCC(A_i, \bar{A}_i) \quad (17) \end{aligned}$$

Normalized graph p -Laplacian matrix $\Delta_p^{(n)}$ corresponds to normalized Cheeger cut $NCC(C, \bar{C}) = \frac{cut(C, \bar{C})}{\min\{vol(C), vol(\bar{C})\}}$, whose second eigenvector $v_p^{(2)}$ should satisfy formula (18):

$$\begin{aligned} &\arg \min_{A_i = \{i \in V | v_p^{(2)}(i) > \tau\}} NCC(A_i, \bar{A}_i) \quad (18) \end{aligned}$$

Methods

How to select a proper distance measurement that can truly describe the internal structure of data set is a critical factor to p -spectral clustering. Data points in the same group should have high similarity and space consistency. The quality of similarity matrix will make a great influence on the performance of spectral clustering [22]. The most common method of measuring the similarities between data points is Gaussian kernel function. However, in Gaussian kernel, the scale parameter σ is usually fixed, so the pairwise data similarity only depend on their Euclidean distance, lacking adaptability to their surrounding points [23]. When dealing with complex multi-scale data sets, the similarity simply based on Euclidean distance cannot accurately reflect the distribution of data, which would significantly reduce the performance of spectral clustering, resulting in poor clustering partition.

Shared-Nearest-Neighbors Similarity Measure

This paper presents a novel similarity measure, which can adaptively adjust the similarities between data points according to their local density. Researchers found that if two points belong to the same cluster, they should locate in the same area with a relatively high density. And between these two points, there will be many intermediate points to bond them together. In order to reflect the glue effect between two points, we define “shared nearest neighbors” (SNN) and use it to extend the self-tuning Gaussian kernel to enlarge the similarities within a cluster.

$SNN(a, b)$ is the number of data points in the intersection of the ε -neighborhood of point a, b . The ε -neighborhood of a point is a spherical region with this point as the sphere center and ε as the radius. $SNN(a, b)$ reflects the local density of point a and point b , helping to distinguish the points in the same cluster or in different clusters. For example, in Fig. 1, point a, b has more common neighbors than point a, c , so $SNN(a, b) > SNN(a, c)$. According to this fact, SNN can guide the algorithm to find the right cluster members.

As SNN is adaptive to the local density of data, it can be used to improve the pairwise similarity of data points, and the mathematical expression is given as follows:

$$w_L(x_i, x_j) = \begin{cases} \exp\left(-\frac{d(x_i, x_j)^2}{\sigma_i \sigma_j (SNN(x_i, x_j) + 1)}\right) & i \neq j \\ 0 & i = j \end{cases} \quad (19)$$

where σ_i and σ_j are, respectively, the Euclidean distance from point x_i, x_j to their m -th nearest neighbors. σ_i and σ_j can automatically and timely adjust themselves according to the sparse or dense distribution of data points within the neighborhoods of point i, j . When two points are located in a relatively sparse cluster, we may enlarge their similarity appropriately by adjusting the scale parameter σ . In this

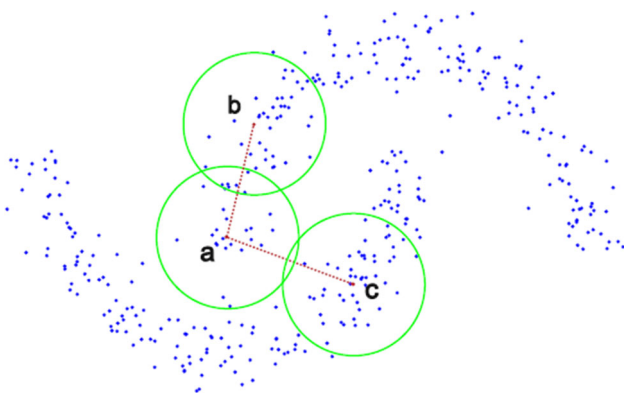


Fig. 1 Common neighbors on Two-Moon data set

way, the points of sparse cluster will have more chance to be grouped together.

This density adaptive similarity measurement has the following features:

1. If $d(x_i, x_j) \geq 2\varepsilon$, then $SNN(x_i, x_j) = 0$, $w_L(x_i, x_j) = \exp(-d(x_i, x_j)^2/\sigma_i \sigma_j)$, which means that the threshold ε is a local parameter and will not affect distant data points.
2. For two pairs of points x_i, x_j and x_m, x_n , assume $d(x_i, x_j) = d(x_m, x_n) < 2\varepsilon$, but in fact, x_i, x_j are in the same dense region, while x_m, x_n are in different dense regions, then there is a great possibility $w_L(x_i, x_j) > w_L(x_m, x_n)$. For example, in Fig. 1, $w_L(a, b) > w_L(a, c)$.

It can be seen from feature (2) that this measurement is density sensitive. It can enlarge the data similarities within the same cluster and weaken the similarities between different clusters. Since clusters are a few dense regions of data set, the SNN-based similarity measurement is able to reflect the distribution of data points in a better way.

Fruit Fly Optimization Method

Although using p -Laplacian operator is able to get an approximate solution of Cheeger cut, it is not easy to determine the parameter p of p -Laplacian matrix. If p 's value is improper, the algorithm would be misled and produce poor clustering results. In order to solve this problem and improve the clustering accuracy, this paper introduces fruit fly optimization method to find the optimal parameters p . Fruit fly optimization algorithm [24] is a novel global optimization method deduced from the foraging behavior of drosophila. Drosophila's sensory perception is superior to other species, especially in the sense of smell and vision. The olfactory organ of fruit fly can well collect various smells floating in the air. When flying near to food, fruit flies will lock the food position by their sharp sense of sight and fly to the direction where their companions are gathered.

Compared with other swarm intelligence algorithms, FOA is simple and easy to understand (for example, the objective function of particle swarm optimization is a second-order differential equation [25], while that of FOA is a first-order differential equation). FOA is also an efficient algorithm and may spend less running time to find the optimal solution. Besides, FOA only has four parameters, but other swarm intelligence algorithms need to adjust at least seven or eight parameters. In these algorithms, the complex relationships and interactions between parameters are difficult to study clearly, and improper parameter values will seriously affect the performance of algorithm, resulting in large calculation errors. According to the characteristics of drosophila searching for food, the specific steps of FOA are shown as Algorithm 1.

Algorithm 1 Fruit fly optimization algorithm

Input: population size $Sizepop$, maximum iteration number $Maxgen$

Output: the optimal value of fitness function, the optimal parameter p

Step 1: Randomly initialize fruit flies' population location X_axis , Y_axis .

Step 2: Set a random direction and distance for each fruit fly searching food with its olfactory; $RandomValue$ is the search distance:

$$\begin{cases} X_i = X_axis + RandomValue \\ Y_i = Y_axis + RandomValue \end{cases} \quad (20)$$

Step 3: As not knowing the exact location of food, first estimate the distance $Dist_i$ between the new position and the origin and then calculate the determine value S_i of smell concentration on the new position, which is the reciprocal of the distance:

$$Dist_i = \sqrt{X_i^2 + Y_i^2} \quad (21)$$

$$S_i = 1/Dist_i \quad (22)$$

Step 4: Bring the determine value S_i into smell consistence decision function (or called fitness function) to calculate the smell concentration $Smell_i$ relative to the position of fruit fly:

$$Smell_i = Function(S_i) \quad (23)$$

Step 5: Identify the *Drosophila* that having the best smell concentration (best individual) in fruit fly population:

$$[bestSmell \ bestindex] = \min(Smell_i) \quad (24)$$

Step 6: Record and retain the best smell concentration value $bestSmell$ and its X , Y coordinates, and then the *Drosophila* groups will fly to this position using their visual sense:

$$\begin{cases} Smellbest = bestSmell \\ X_axis = X(bestindex) \\ Y_axis = Y(bestindex) \end{cases} \quad (25)$$

Step 7: Enter the iterative refinement process, repeat from Step 2 to Step 5, and determine whether the best smell concentration is better than the previous value. Meanwhile, make sure the current iteration time is less than the maximum iteration number $Maxgen$, otherwise jump to Step 6.

Self-Tuning p -Spectral Clustering Algorithm Based on Shared Nearest Neighbors

In order to improve the performance of p -spectral clustering algorithm, this paper introduces SNN similarity measure and fruit fly optimization method and proposes a self-

tuning p -spectral clustering algorithm based on shared nearest neighbors (SNN-PSC). Its main idea is as follows: first, measure the similarity of pairwise points according to their shared neighbors' number to construct similarity matrix; recursively divide the sub-graph into two parts and calculate the eigenvalues and eigenvectors of p -Laplacian matrix; use fruit fly optimization method to determine the optimum value of parameter p ; minimize the graph cut criterion; and finally obtain high quality clustering results. The detailed steps of SNN-PSC algorithm are given in Algorithm 2, and its flowchart is shown in Fig. 2.

Algorithm 2 Self-tuning p -spectral clustering algorithm based on shared nearest neighbors

Input: data set $X = \{x_1, x_2, \dots, x_n\}$, the number of clusters k

Output: k divided clusters

Step 1: According to formula (19), calculate the similarities between data points based on "shared nearest-neighbors" and create the similarity matrix $W \in R^{n \times n}$.

Step 2: Initialize the first cluster $A_1 = V$ and set the cluster number $s = 1$.

Step 3: Repeat from Step 3 to Step 7.

Step 4: Construct p -Laplacian matrix according to formula (7) or (8); then treat the objective function $F_p(f)$ as FOA's fitness function and use FOA algorithm to find the optimal parameter p to minimize the fitness function.

Step 5: Calculate the second eigenvector $v_p^{(2)}$ of un-normalized p -Laplacian matrix $\Delta_p^{(u)}$ (or normalized p -Laplacian matrix $\Delta_p^{(n)}$) and search an appropriate threshold value that satisfies formula (17) (or formula (18)).

Step 6: Use $v_p^{(2)}$ to split each cluster $A_i (i = 1, 2, \dots, s)$ and minimize the overall cut criterion [ratio cut (2) corresponds to $\Delta_p^{(u)}$, normalized cut (3) corresponds to $\Delta_p^{(n)}$].

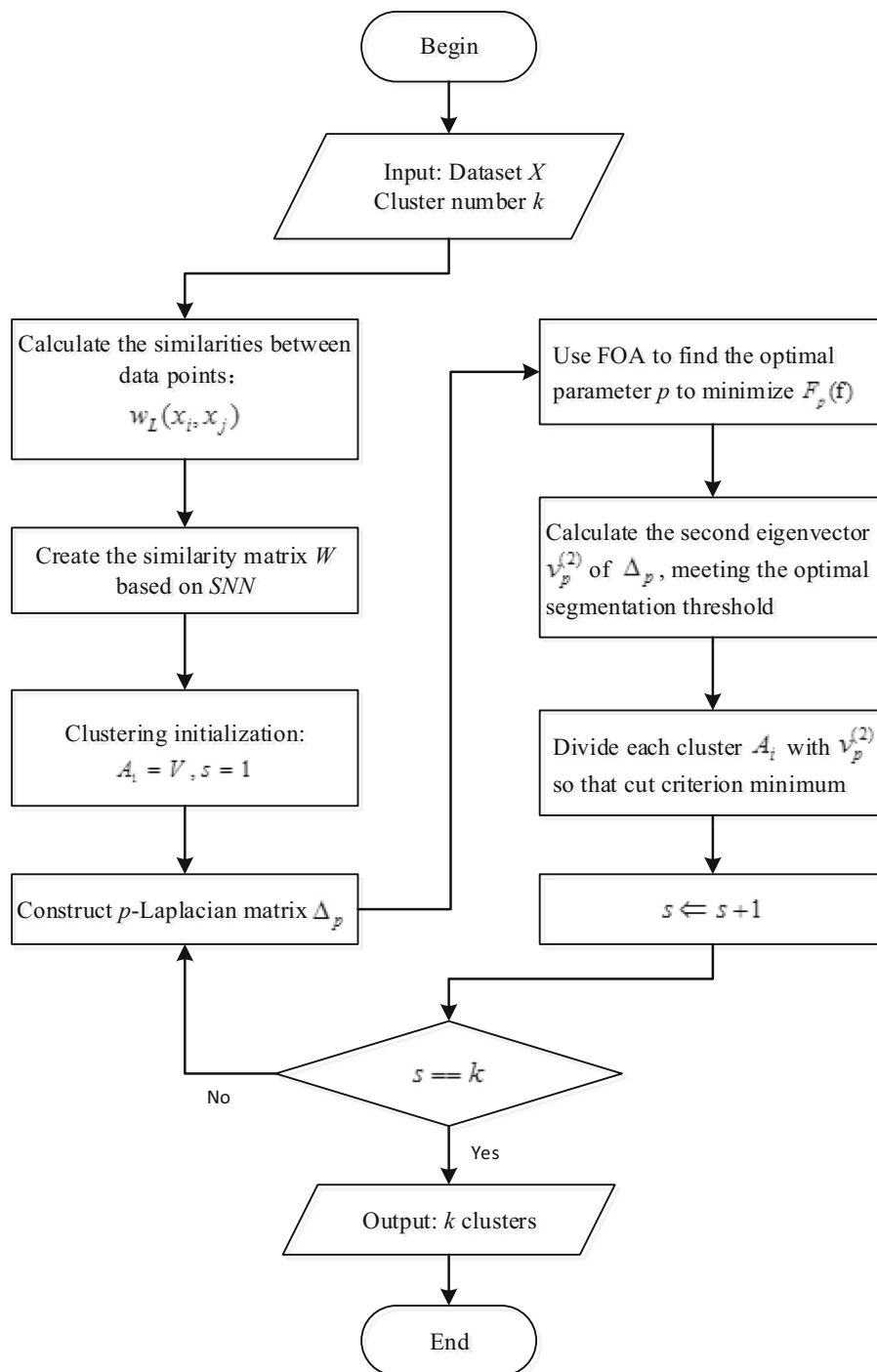
Step 7: $s \leftarrow s + 1$.

Step 8: When the number of clusters $s = k$, stop the loop and output the clustering results.

Discussion

In order to analyze the clustering performance of proposed SNN-PSC algorithm, we conduct the experiments on both artificial data sets and UCI real-world data sets, and compare SNN-PSC algorithm with the standard spectral clustering algorithm (denoted as SSC) [26], p -spectral clustering algorithm (denoted as PSC) [14]. In the experiments, SNN-PSC algorithm's population size $Sizepop = 50$, maximum iterations $Maxgen = 100$; SSC algorithm's scale parameter $\sigma = 1.2$. On artificial data sets, PSC algorithm's Laplace parameter $p = 1.5$; on UCI data sets, the value of p is 1.8, 1.6, 1.4, 1.2, respectively. The experimental environments of computer are as follows:

Fig. 2 Flowchart of SNN-PSC algorithm



Pentium dual-core 2.60 GHz CPU, 2 GB RAM, MATLAB 2013a programming platform.

Experiments on Artificial Data Sets

There are two kinds of artificial data sets used in the experiments, namely Two-Moon data set and Two-Gaussian data set. Each data set has two types “Balanced” and “Unbalanced”. Every type of data set contains 1000 data

points embedded in a 10-dimensional space. And Gaussian noise $N(0, \sigma^2)$ is added in these data sets; the noise variance $\sigma^2 = 0.01$. The characteristics of different data sets are given in Table 1.

Figure 3 shows the clustering results of three algorithms on Two-Gaussian data sets. The first row is SSC algorithm, the second row is PSC algorithm, and the third row is SNN-PSC algorithm. It can be seen from the figure that the clustering performance of these three algorithms is quite

Table 1 Characteristics of artificial data sets

Data set	Data set type	Number of data points	
		Class 1	Class 2
Two-Gaussian	Balanced	483	517
	Unbalanced	199	801
Two-Moon	Balanced	512	488
	Unbalanced	195	805

well on balanced Two-Gaussian data set. When the data set is unbalanced, PSC algorithms and SNN-PSC algorithm can still effectively classify the data points into two groups, but for SSC algorithm, partial points of sparse cluster are erroneously assigned into the dense cluster and the clustering result is not satisfactory.

Figure 4 shows the clustering results of three algorithms on Two-Moon data sets. The first row is SSC algorithm, the second row is PSC algorithm, and the third row is SNN-PSC algorithm. It can be seen from the figure that SSC algorithm performs badly on both balanced data set and unbalanced data set; PSC algorithm can effectively handle balanced Two-Moon data set, but may produce error partitions for unbalanced data set; only SNN-PSC algorithm has strong adaptive capacity and can obtain better clustering results on the two types of data sets.

Experiments on Real-World Data Sets

To further verify the effectiveness of SNN-PSC algorithm, we select five real data sets from UCI machine learning database. Their data characteristics are shown in Table 2.

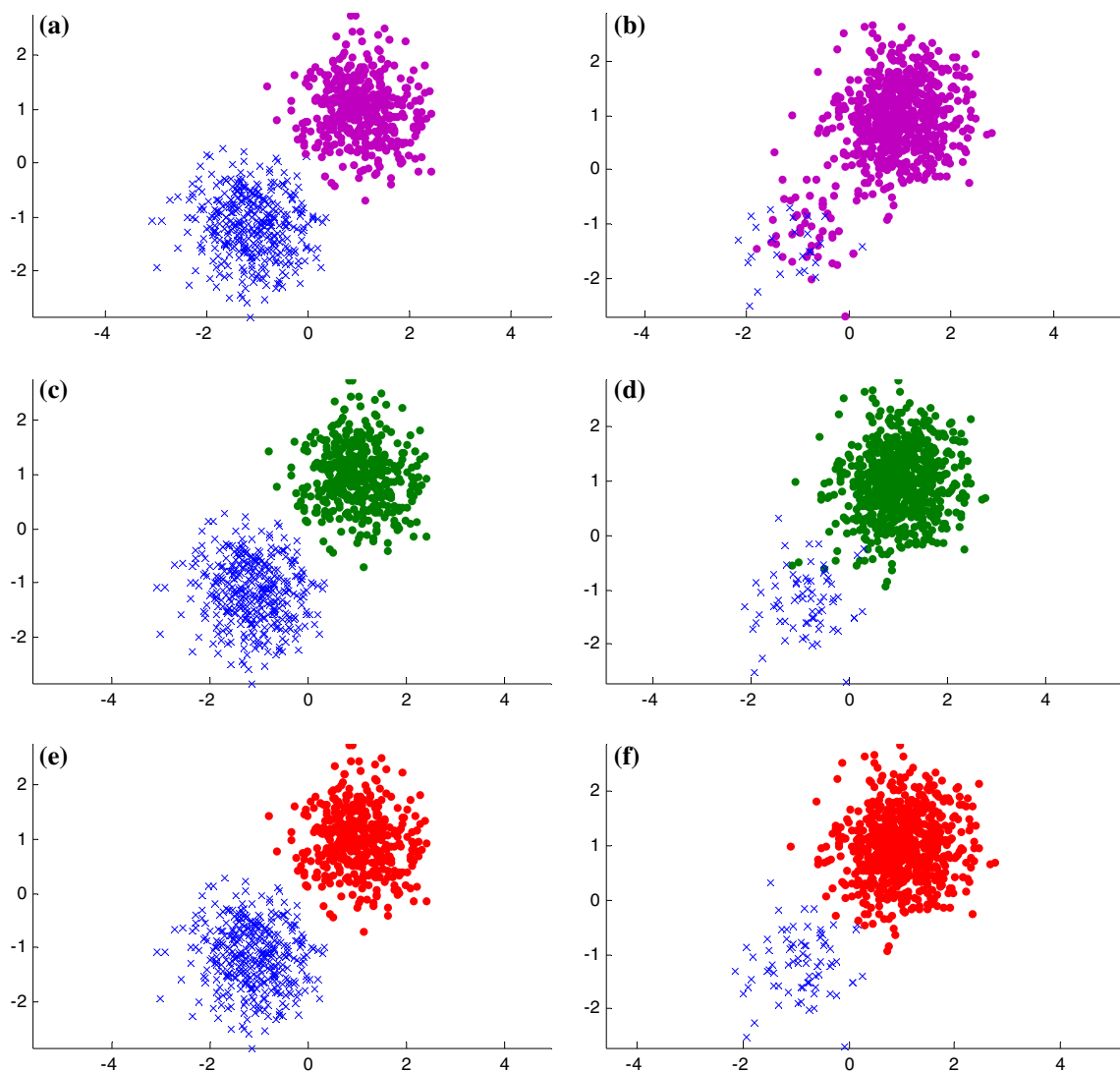


Fig. 3 Clustering results on Two-Gaussian data sets. **a** SSC algorithm (Balanced dataset), **b** SSC algorithm (Unbalanced dataset), **c** PSC algorithm (Balanced dataset), **d** PSC algorithm (Unbalanced

dataset), **e** SNN-PSC algorithm (Balanced dataset), **f** SNN-PSC algorithm (Unbalanced dataset)

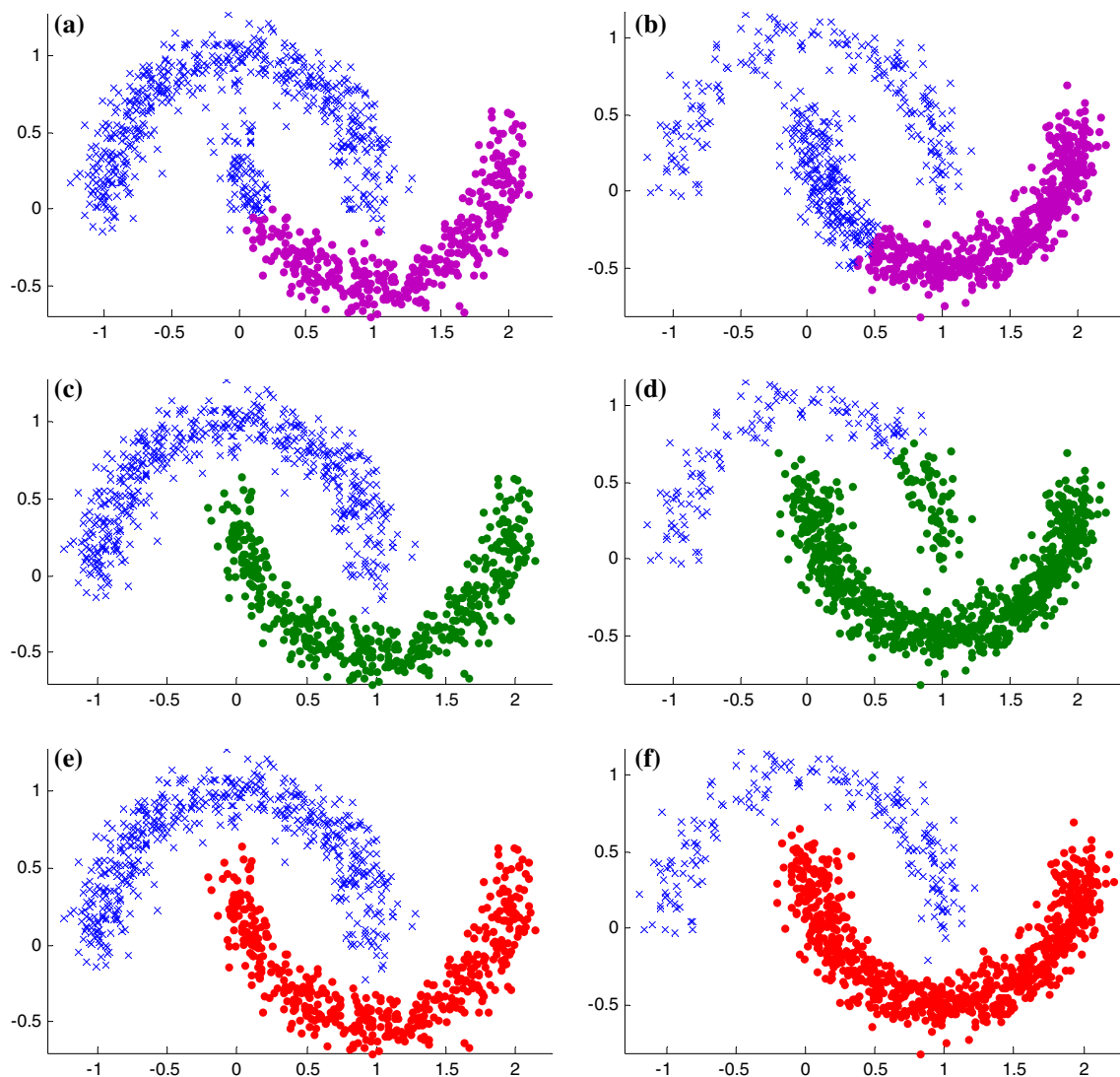


Fig. 4 Clustering results on Two-Moon data sets. **a** SSC algorithm (Balanced dataset), **b** SSC algorithm (Unbalanced dataset), **c** PSC algorithm (Balanced dataset), **d** PSC algorithm (Unbalanced dataset),

e SNN-PSC algorithm (Balanced dataset), **f** SNN-PSC algorithm (Unbalanced dataset)

Table 2 Characteristics of UCI data sets

Data set	Number of samples	Number of attributes	Number of classes
Dermatology	366	34	6
Heart	270	14	2
WDBC	569	30	2
Wine	178	13	3
Zoo	101	16	7

Clustering error rate (CE) is a commonly used method to evaluate the performance of clustering [27]. After obtaining the clustering results, we need to build a permutation mapping function to find corresponding relationship

between the cluster label and the actual class label. Clustering error rate is based on this mapping relationship, and its calculation formula is as follows:

$$CE = 1 - \frac{1}{n} \sum_{i=1}^n \delta(y_i, \text{map}(c_i)) \quad (26)$$

where y_i is the true class label, c_i is the serial number of clusters obtained by clustering algorithm, $\delta(x, y) =$

$$\begin{cases} 1 & x = y \\ 0 & x \neq y \end{cases}$$

is a discriminant function. Among all possible corresponding relations, clustering error rate is determined by the case that the number of misclassified data points is the minimum. The smaller the clustering error rate, the better the performance of clustering algorithm.

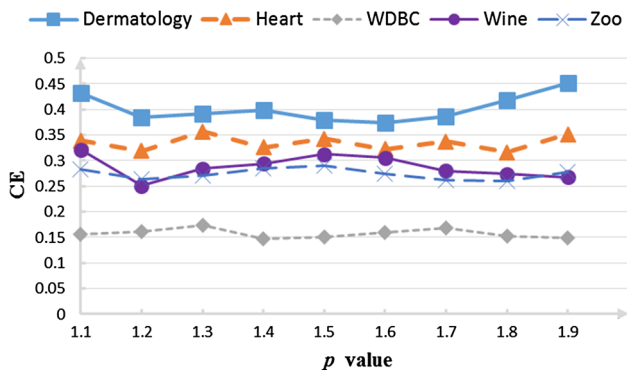


Fig. 5 Clustering performance of PSC with different parameter p

We first test the clustering performance of PSC algorithm on UCI real-world data sets. The clustering results are shown in Fig. 5. In Fig. 5, the horizontal coordinate is the parameter p of p -Laplacian matrix with different values and the vertical coordinate is the CE rates of algorithm on different data sets. From Fig. 5, we can see that with the changes in p 's value, the clustering performance of PSC algorithm also swings up or down. The fluctuation of the curves indicates that PSC algorithm is easily affected by parameter p . For different data sets, in order to give full play to the performance of PSC algorithm, the selection of proper parameters is crucial. So we use FOA algorithm to automatically find the optimal parameter p to improve the clustering performance. Table 3 shows the concrete comparison of the CE rates and clustering time of SSC algorithm, PSC algorithm, FOA-PSC algorithm and SNN-PSC algorithm on UCI data sets. FOA-PSC algorithm served as a control uses FOA directly on the parameter selection for PSC, without introducing the SNN measure.

From the experimental data, we can see that the clustering time of SSC algorithm on each data set is the shortest, but its clustering performance is not as good as PSC algorithm, FOA-PSC algorithm and SNN-PSC algorithm. PSC algorithm is sensitive to the value of parameter p ; when p is set with different values, the clustering time will make a great difference. As shown in Table 3, with p 's value gets close to one, it will cost much more time for PSC algorithm to converge. FOA-PSC algorithm performs much better than PSC algorithm as its parameter p is optimized and its clustering error rate is the lowest on Dermatology data set. But for the rest of data sets, the CE rates of SNN-PSC algorithm are much lower than the others, which demonstrates that SNN-PSC can provide better clustering results. Because in SNN-PSC algorithm, the similarity matrix is constructed by SNN method, which can well describe the intrinsic link between data points, and FOA algorithm is used to adaptively search the optimum value of parameter p , avoiding the blindness of parameter selection, so the clustering accuracy is improved to some extent.

Conclusion

In spectral clustering, an undirected weighted graph made up of data points will be separated into a series of sub-graphs. And each sub-graph represents a cluster in the final clustering results. Cheeger cut criterion can produce balanced clusters, but according to algebraic graph theory, calculating the global optimal solution of Cheeger cut is an NP-hard problem. An approximate solution is relaxing the

Table 3 The clustering performance of different algorithms on UCI data sets

Data set	SSC	PSC				FOA-PSC	SNN-PSC
		$p = 1.8$	$p = 1.6$	$p = 1.4$	$p = 1.2$		
Dermatology							
CE	0.3786	0.4185	0.3741	0.3995	0.3851	0.3672	0.3750
Time (s)	0.59	9.25	23.51	15.06	45.18	17.33	18.62
Heart							
CE	0.3281	0.3161	0.3227	0.3256	0.3185	0.3153	0.3074
Time (s)	0.31	8.87	13.52	32.94	58.11	23.18	21.85
WDBC							
CE	0.1567	0.1523	0.1594	0.1477	0.1612	0.1477	0.1459
Time (s)	1.81	11.62	16.67	53.69	106.23	55.62	57.91
Wine							
CE	0.3206	0.2738	0.3056	0.2941	0.2509	0.2487	0.2326
Time (s)	0.13	20.43	17.03	12.91	74.73	35.46	40.28
Zoo							
CE	0.3162	0.2595	0.2743	0.2845	0.2637	0.2595	0.2574
Time (s)	0.51	4.37	3.99	9.32	25.04	6.57	8.36

discrete optimization problem to the real number field by the eigen-decomposition of p -Laplacian matrix. As the similarity measurement plays an important role in p -spectral clustering, this paper develops a density adaptive similarity measurement based on shared nearest neighbors, which takes into account the neighborhood condition of each data point and may reveal the local density information of data structure. Then we present a SNN-based self-tuning p -spectral clustering algorithm called SNN-PSC. In order to avoid the negative influence of parameter selection, FOA method is used to determine the optimal parameter p of p -Laplacian operator. Comprehensive experiments on both artificial and real-world data sets demonstrate that SNN-PSC algorithm is robust to the noise around data points. The performance of SNN-PSC is much better than that of the standard spectral clustering algorithm and p -spectral clustering algorithm. Next we will study how to reduce the computational complexity of SNN-PSC algorithm and improve its operating efficiency.

Acknowledgments This work is supported by the National Natural Science Foundation of China (No. 61379101), and the National Key Basic Research Program of China (No. 2013CB329502).

References

- Byun SS, Balashingham I, Vasilakos AV, et al. Computation of an equilibrium in spectrum markets for cognitive radio networks. *IEEE Trans Comput*. 2014;63(2):304–16.
- Jiu MY, Wolf C, Garcia C, et al. Supervised learning and codebook optimization for bag-of-words models. *Cogn Comput*. 2012;4(4):409–19.
- Huang XX, Huang HX, Liao BS, et al. An ontology-based approach to metaphor cognitive computation. *Mind Mach*. 2013;23(1):105–21.
- Bian XY, Zhang TX, Zhang XL, et al. Clustering-based extraction of near border data samples for remote sensing image classification. *Cogn Comput*. 2013;5(1):19–31.
- Zeng S, Huang R, Kang Z, et al. Image segmentation using spectral clustering of Gaussian mixture models. *Neurocomputing*. 2014;144:346–56.
- Mital PK, Smith TJ, Hill RL, et al. Clustering of gaze during dynamic scene viewing is predicted by motion. *Cogn Comput*. 2011;3(1):5–24.
- Gao XB, Deng C, Li XL, Tao DC. Local feature based geometric-resistant image information hiding. *Cogn Comput*. 2010;2(2):68–77.
- Li XW. A new text clustering algorithm based on improved K-means. *Journal of Software*. 2012;7(1):95–101.
- Jia HJ, Ding SF, Xu XZ, Nie R. The latest research progress on spectral clustering. *Neural Comput Appl*. 2014;24(7–8):1477–86.
- Blekas K, Lagaris IE. A spectral clustering approach based on Newton's equations of motion. *Int J Intell Syst*. 2013;28(4):394–410.
- Liu AH, Poon LKM, Liu TF, et al. Latent tree models for rounding in spectral clustering. *Neurocomputing*. 2014;144:448–62.
- Jia HJ, Ding SF, Meng LH, et al. A density-adaptive affinity propagation clustering algorithm based on spectral dimension reduction. *Neural Comput Appl*. 2014;25(7–8):1557–67.
- Amghibech S. Eigenvalues of the discrete p -Laplacian for graphs. *Ars Comb*. 2003;67:283–302.
- Bühler T, Hein M. Spectral clustering based on the graph p -Laplacian. *Proceedings of the 26th international conference on machine learning (ICML 2009)*, 2009; p. 81–88.
- Fiedler M. Algebraic connectivity of graphs. *Czechoslov Math J*. 1973;23(98):298–305.
- MacDonald JK. Successive approximations by the Rayleigh-Ritz variation method. *Phys Rev*. 1933;43(10):830–3.
- Hagen L, Kahng AB. New spectral methods for radio cut partitioning and clustering. *IEEE Trans Comput Aided Des Integr Circuits Syst*. 1992;11(9):1074–85.
- Shi J, Malik J. Normalized cuts and image segmentation. *IEEE Trans Pattern Anal Mach Intell*. 2000;22(8):888–905.
- Amghibech S. Bounds for the largest p -Laplacian eigenvalue for graphs. *Discrete Math*. 2006;306(21):2762–71.
- Hein M, Audibert JY, Von Luxburg U. Graph Laplacians and their convergence on random neighborhood graphs. *J Mach Learn Res*. 2007;8(12):1325–68.
- Arthur Szlam, Xavier Bresson. Total variation and Cheeger cuts. *Proceedings of the 27th international conference on machine learning*, 2010; p. 233–240.
- Nguyen T, Khosravi A, Creighton D, et al. Spike sorting using locality preserving projection with gap statistics and landmark-based spectral clustering. *J Neurosci Methods*. 2014;238:43–53.
- Ding SF, Jia HJ, Zhang LW, Jin FX. Research of semi-supervised spectral clustering algorithm based on pairwise constraints. *Neural Comput Appl*. 2014;24(1):211–9.
- Pan WT. A new fruit fly optimization algorithm: taking the financial distress model as an example. *Knowl Based Syst*. 2012;26:69–74.
- Gosciniak I. A new approach to particle swarm optimization algorithm. *Expert Syst Appl*. 2015;42(2):844–54.
- Ng AY, Jordan MI, Weiss Y. On spectral clustering: analysis and an algorithm. *Adv Neural Inf Process Syst*. 2002;14:849–56.
- Ding SF, Jia HJ, Shi ZZ. Spectral clustering algorithm based on adaptive Nyström sampling for big data analysis. *J Softw*. 2014;25(9):2037–49.