Hindawi
Mathematical Problems in Engineering
Volume 2019, Article ID 1713801, 13 pages
https://doi.org/10.1155/2019/1713801

Research Article

A Density Peak Clustering Algorithm Based on the K-Nearest Shannon Entropy and Tissue-Like P System

Zhenni Jiang,[1] Xiyu Liu ,[1] and Minghe Sun [2]

[1]*Business School, Shandong Normal University, Jinan, China*
[2]*Business School, University of Texas at San Antonio, San Antonio, USA*

Correspondence should be addressed to Xiyu Liu; xyliu@sdnu.edu.cn

Received 29 March 2019; Revised 8 June 2019; Accepted 20 June 2019; Published 31 July 2019

Academic Editor: Paolo Spagnolo

This study proposes a novel method to calculate the density of the data points based on K-nearest neighbors and Shannon entropy. A variant of tissue-like P systems with active membranes is introduced to realize the clustering process. The new variant of tissue-like P systems can improve the efficiency of the algorithm and reduce the computation complexity. Finally, experimental results on synthetic and real-world datasets show that the new method is more effective than the other state-of-the-art clustering methods.

1. Introduction

Clustering is an unsupervised learning method, which aims to divide a given population into several groups or classes, called clusters, in such a way that similar objects are put into the same group and dissimilar objects are put into different groups. Clustering methods generally include five categories: partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods [1]. Partitioning and hierarchical methods can find spherical-shaped clusters but do not perform well on arbitrary clusters. Density-based clustering [2] methods can be used to overcome this problem, which can model clusters as dense regions and boundaries as sparse regions. Three representative approaches of the density-based clustering method are DBSCAN (Density-Based Spatial Clustering of Application with Noise), OPTICS (Ordering Points to Identify the Clustering Structure), and DENCLUE (DENsity-based CLUstEring).

Usually, an objective function measuring the clustering quality is optimized by an iterative process in some clustering algorithms. However, this approach may cause low efficiency. Thus, the density peaks clustering (DPC) algorithm was proposed by Rodriguez and Laio [3] in 2014. This method can obtain the clusters in a single step regardless of the shape and dimensionality of the space. DPC is based on the idea that cluster centers are characterized by a higher density than the surrounding regions by a relatively large distance from points with higher densities. For the DPC algorithm, scholars have done a lot of research. However, DPC still has several challenges that need to be addressed. First, the local density of data points can be affected by the cut off distance, which can influence the clustering results. Second, the number of clusters needs to be decided by users, but the manual selection of the cluster centers can influence the clustering result. Cong et al. [4] proposed a clustering model for high dimensional data based on DPC that accomplishes clustering simply and directly for data with more than six-dimensions with arbitrary shapes. The problem in this model is that clustering effect is not ideal for different classes and big differences in order of magnitude. Xu et al. [5] introduced a novel approach, called density peaks clustering algorithm based on grid (DPCG). But this method also needs to rely on the user experiment in the choice of clustering centers. Bie et al. [6] proposed a fuzzy-CFSFDP method for adaptively but effectively selecting the cluster centers. Du et al. [7] proposed a new DPC algorithm using geodesic distances. And Du et al. [8] also proposed a FN-DP (fuzzy neighborhood density peaks) clustering algorithm. But they cannot select clustering center automatic and the FN-DP algorithm can cost much time in the calculation of the similarity matrix. Hou and Cui [9] introduced a density normalization step to make

large-density clusters partitioned into multiple parts and small-density clusters merged with other clusters. Xu et al. proposed a FDPC algorithm based on a novel merging strategy motivated by support vector machines [10]. But it also has the problem of higher complexity and needs to select the clustering center by users. Liu et al. [11] proposed a shared-nearest-neighbor-based clustering by fast search and find of density peaks (SNN-DPC) algorithm. Based on prior assumptions of consistency for semisupervised learning algorithms, some scholars also made assumptions of consistency for density-based clustering. The first assumption is of local consistency, which means nearby points are likely to have similar local density, and the second assumption is of global consistency, which means points in the same high-density area (or the same structure, i.e., the same cluster) are likely to have the same label [12]. This method also cannot find the clustering centers automatically. Although many studies about DPC have been reported, it still has many problems that need to be studied.

Membrane computing proposed by Păun [13], as a new branch of natural computing, abstracts out computational models from the structures and functions of biological cells and from the collaboration between organs and tissues. Membrane computing mainly includes three basic computational models, i.e., the cell-like P system, the tissue-like P system, and the neural-like P system. In the computation process, each cell is treated as an independent unit, each unit operates independently and does not interfere with each other, and the entire membrane system operates in maximally parallel. Over the past years, many variants of membrane systems have been proposed [14–18], including membrane algorithms of solving global optimization problems. In recent years, applications of membrane computing have attracted a lot of attention from researchers [19–22]. There are also some other applications; for example, membrane systems are used to solve multiobjective fuzzy clustering problems [23], solve unsupervised learning algorithms [24], solve automatic fuzzy clustering problems [25], and solve the problems of fault diagnosis of power systems [26]. Liu et al. [27] proposed an improved Apriori algorithm based on an Evolution-Communication tissue-Like P System. Liu and Xue [28] introduced a P system on simplices. Zhao et al. [29] proposed a spiking neural P system with neuron division and dissolution.

Based on previous works, the main motivation of this work is using membrane systems to develop a framework for a density peak clustering algorithm. A new method of calculating the density of the data points is proposed based on the K-nearest neighbors and Shannon entropy. A variant of the tissue-like P system with active membranes is used to realize the clustering process. The new model of the P system can improve efficiency and reduce computation complexity. Experimental results show that this method is more effective and accurate than the state-of-the-art methods.

The rest of this paper is organized as follows. Section 2 describes the basic DPC algorithm and the tissue-like P system. Section 3 introduces the tissue-like P system with active membranes for DPC based on the K-nearest neighbors and Shannon entropy and describes the clustering procedure. Section 4 reports experimental results on synthetic datasets and UCI datasets. Conclusions are drawn and future research directions are outlined in Section 5.

2. Preliminaries

2.1. The Original Density Peak Clustering Algorithm. Rodriguez and Laio [3] proposed the DPC algorithm in 2014. This algorithm is based on the idea that cluster centers have higher densities than the surrounding regions and the distances among cluster centers are relatively large. It has three important parameters. The first one ρ_i is the local density of data point i, the second one δ_i is the minimum distance between data point i and other data points with higher density, and the third one $\gamma_i = \rho_i \times \delta_i$ is the product of the other two. The first two parameters correspond to two assumptions of the DPC algorithm. One assumption is that cluster centers have higher density than the surrounding regions. The other assumption is that this point has larger distance from the points in other clusters than from points in the same cluster. In the following, the computations of ρ_i and δ_i are discussed in detail.

Let $\boldsymbol{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ be a dataset with n data points. Each $\mathbf{x}_i$ has M attributes. Therefore, x_{ij} is the jth attribute of data point $\mathbf{x}_i$. The Euclidean distance between the data points $\mathbf{x}_i$ and $\mathbf{x}_j$ can be expressed as follows:

$$d_{ij} = d\left(\mathbf{x}_i, \mathbf{x}_j\right) = \left\|\mathbf{x}_i - \mathbf{x}_j\right\| \tag{1}$$

The local density ρ_i of the data point $\mathbf{x}_i$ is defined as

$$\rho_i = \sum_{j \neq i} \chi\left(d\left(\mathbf{x}_i - \mathbf{x}_j\right) - d_c\right) \tag{2}$$

with

$$\chi(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} < 0 \\ 0, & \mathbf{x} > 0, \end{cases} \tag{3}$$

where d_c is the cutoff distance. In fact, ρ_i is the number of data points adjacent to data point $\mathbf{x}_i$. The minimal distance δ_i between data point $\mathbf{x}_i$ and any other data points $\mathbf{x}_{i'}$ with a higher density $\rho_{i'}$ is given by

$$\delta_i = \begin{cases} \min\limits_{j:\rho_{i'} > \rho_i} \left(d_{ij}\right), & if\ \exists i^i, s.t.\ \rho_{i'} > \rho_i \\ \max\limits_{i'} \left(d_{ii'}\right), & \text{otherwise} \end{cases} \tag{4}$$

After ρ_i and δ_i are calculated for each data point $\mathbf{x}_i$, a decision graph with δ_i on the vertical axis and ρ_i on the horizontal axis can be plotted. This graph can be used to find the cluster centers and then to assign each remaining data point to the cluster with the shorted distance.

The computation of the local densities of the data points is a key factor for the effectiveness and efficiency of the DPC. There are many other ways to calculate the local densities. For example, the local density of $\mathbf{x}_i$ can be computed using (5) in the following [3]:

$$\rho_i = \sum_j \exp\left(-\frac{d_{ij}^2}{{d_c}^2}\right) \tag{5}$$

The way in (5) is suitable for "small" datasets. In fact, it is difficult to judge if the dataset is small or large. When (5) is used to calculate the local density, the results can be greatly affected by the cutoff distance d_c.

Each component on the right side of (5) is a Gaussian function. Figure 1 visualizes the function $\exp(-t)$ and two Gaussian functions $\exp(-t^2/\sigma^2)$ with different values of σ. The blue and red curves are the curves of $\exp(-t^2/\sigma^2)$ with $\sigma = 1$ and $\sigma = \sqrt{2}$, respectively. The curve with a smaller value of σ declines more quickly than the curve with a larger value of σ. Comparing the curve of $\exp(-t)$, the yellow dash dotted curve, with the curves of $\exp(-t^2/\sigma^2)$, it can be found that values of $\exp(-t^2/\sigma^2)$ are greater than those of $\exp(-t)$ when $t < \sigma^2$ but decay faster when $t > \sigma^2$. This means that if the value of the parameter σ needs to be decided manually in the density calculation, the result of the calculated densities will be influenced by the selected value. This analysis shows that the parameter σ has a big effect on the calculated results. Furthermore, the density in (5) can be influenced by the cutoff distance d_c. To eliminate the influence from the cutoff distance d_c and give a uniform metric for datasets with any size, Du et al. [30] proposed the K-nearest neighbor method. The local density in Du et al. [30] is given by

$$\rho_i = \exp\left(-\frac{1}{K}\sum_{j\in KNN_i} d_{ij}^2\right), \tag{6}$$

where K is an input parameter and KNN_i is the set of the K nearest neighbors of data point $\mathbf{x}_i$. However, this method did not consider the influence of the position of the data point on its own density. Therefore, this current study proposes a novel method to calculate the densities.

2.2. The Tissue-Like P System with Active Membrane. A tissue-like P system has a graphical structure. The nodes of the graph correspond to the cells and the environment in the tissue-like P system, whereas the edges of the graph represent the channels for communication between the cells. The tissue-like P system is slightly more complicated than the cell-like P system. Each cell has a different state. Only the state that meets the requirement specified by the rules can be changed. The basic framework of the tissue-like P system used in this study is shown in Figure 2.

A P system with active membranes is a construct:

$$\prod = \left(O, Z, H, \omega_1, \ldots, \omega_m, E, ch, \left(s_{(i,j)}\right)_{(i,j)\in ch}, \left(R_{(i,j)}\right)_{(i,j)\in ch}, i_0\right) \tag{7}$$

where

(1) O is the set of alphabets of all objects which appear in the system;

(2) Z represents the states of the alphabets;

(3) H is the set of labels of the membranes;

(4) $\omega_1, \ldots, \omega_m$ are the initial multiple objects in cells 1 to m;

(5) $E \subseteq O$ is the set of objects present in an arbitrary number of copies in the environment;

(6) $ch \subseteq \{(i, j) \mid i, j \in \{0, 1, \ldots, m\}, i \neq j\}$ is the set of channels between cells and between cells and the environment;

(7) $s_{(i,j)}$ is the initial state of the channel (i, j);

(8) $R_{(i,j)}$ is a finite set of symport/antiport rules of the form $(s, x/y, s')$ with $s, s' \in Z$ and x, y $\in O$:

(i) $[\mathrm{a} \longrightarrow \mathrm{b}]_h$, where $h \in H$, $a \in O$, and $b \in O$. (Object evolution rules: an object is evolved into another in a membrane).

(ii) $\mathrm{a}[]_h \longrightarrow [\mathrm{b}]_h$, where $h \in H$ and $a, b \in O$. (Send-in communication rules: an object is introduced into a membrane and may be modified during the process).

(iii) $[\mathrm{a}]_h \longrightarrow []_h b$, where $h \in H, a, b \in O$. (Send-out communication rules: an object is sent out of the membrane and may be modified during the process).

(iv) $[\mathrm{a}]_h \longrightarrow [\mathrm{b}]_{h_1}[\mathrm{c}]_{h_2}$, where $h, h_1, h_2 \in H$ and $a, b \in O$. (Division rules for elementary membranes: the membrane is divided into two membranes with possibly different labels; the object specified in the rule is replaced by possibly new objects in the two new membranes; and the remaining objects are duplicated in the process).

(9) $i_0 \in \{1, \ldots, m\}$ is the output cell.

The biggest difference between a cell-like P system and a tissue-like P system is that each cell can communicate with the environment in the tissue-like P system, but only the skin membrane can communicate with the environment in the cell-like P system. This does not mean that any two cells in the tissue-like P system can communicate with each other. If there is no direct communication channel between the two cells, they can communicate through the environment indirectly.

3. The Proposed Method

3.1. Density Metric Based on the K-Nearest Neighbors and Shannon Entropy. DPC still has some defects. The current DPC algorithm has the obvious shortcoming that it needs to set the value of the cutoff distance d_c manually in advance. This value will largely affect the final clustering results. In order to overcome this shortcoming, a new method is proposed to calculate the density metric based on the K-nearest neighbors and Shannon entropy.

K-nearest neighbors (KNN) is usually used to measure a local neighborhood of an instance in the fields of classification, clustering, local outlier detection, etc. The aim of this approach is to find the K-nearest neighbors of a sample among N samples. In general, the distances between points are achieved by calculating the Euclidean distance. Let

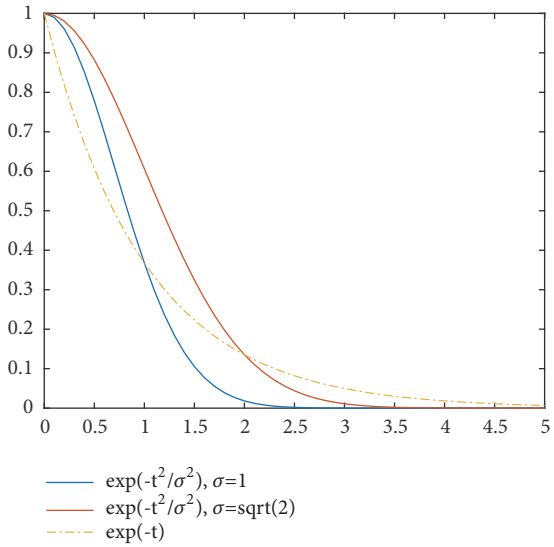

Figure 1: Three different function curves.

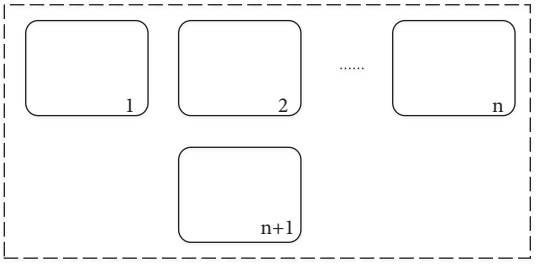

Figure 2: Membrane structure of a tissue-like P system.

KNN(i) be a set of nearest neighbors of a point i and it can be expressed as

$$KNN\,(i) = \left\{ j \mid d_{ij} \le d_{i,kth(i)} \right\}, \tag{8}$$

where $d(\mathbf{x}_i, \mathbf{x}_j)$ is the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ and $kth(i)$ is the k-th nearest neighbor of i. Local regions measured by KNN are often termed K-nearest neighborhood, which, in fact, is a circular or spherical area or radius $R = d_{i,kth(i)}$. Therefore, KNN-based method cannot apply to handle datasets with clusters nonspherical distributions. Therefore, these methods usually have poor clustering results when handling datasets with clusters of different shapes.

Shannon entropy measures the degree of molecular activity. The more unstable the system is, the larger the value of the Shannon entropy is, and vice versa. The Shannon entropy, represented by $H(X)$, is given by

$$H\,(X) = -\sum_{i=0}^{N} p_i \log_2 (p_i), \tag{9}$$

where X is the set of objects and p_i is the probability of object i appearing in X. When $H(X)$ is used to measure the distance between the clusters, the smaller the value of $H(X)$ is, the better the clustering result is. Therefore, the Shannon entropy is introduced to calculate the data point density in the K-nearest neighbor method, so that the final density calculation not only considers the distance metric, but also adds the influence of the data point position to the density of the data point.

However, the decision graph is calculated by the product of ρ_i and δ_i. A larger value of ρ_i makes it easier to choose the best clustering centers. Therefore, the reciprocal form of the Shannon entropy is adopted. The metrics for ρ_i and δ_i may be inconsistent, which directly leads to ρ_i and δ_i playing different roles in the calculation of the decision graph. Hence, it is necessary to normalize ρ_i and δ_i.

The specific calculation method is as follows. First, the local density of data point $\mathbf{x}_i$ is calculated:

$$w_i' = \sum_{j=1, j\neq i}^{n} \frac{1}{\left\| \mathbf{x}_j - \mathbf{x}_i \right\|}, \tag{10}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are data points and w_i' is the density of data point $\mathbf{x}_i$. Next, the density of data point $\mathbf{x}_i$ is normalized and the normalized density is denoted as w_i:

$$w_i = \frac{w_i'}{\sum_{i=1}^{n} w_i'}. \tag{11}$$

Finally, the density metric which uses the idea of the K-nearest neighbor method is defined as

$$\rho_i = -\frac{1}{(1/K)\sum_{i\in KNN} w_i \log(w_i)}. \quad (12)$$

To guarantee the consistence of the metrics of ρ_i and δ_i, δ_i also needs to be normalized.

3.2. Tissue-Like P System with Active Membranes for Improved Density Peak Clustering. In the following, a tissue-like P system with active membranes for density peak clustering, called KST-DPC, is proposed. As mentioned before, assume the dataset with n data points is represented by $\boldsymbol{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$. Before performing any specific calculation of the DPC algorithm, the Euclidean distance between each pair of data points in the dataset is calculated and the result is stored in the form of a matrix. The initial configuration of this P system is shown in Figure 3.

When the system is initialized, the objects $\mathbf{x}_i, \mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ are in membrane i for $1 \le i \le n$ and object λ is in membrane $n+1$, where λ means there is no object. First, the Euclidean distance w_{ij} between the data points $\mathbf{x}_i$ and $\mathbf{x}_j$ (represented by $\mathbf{b}_j$ for $1 \le j \le n$) is calculated with the rule $r_1 = \{[\mathbf{x}_i\mathbf{b}_1\mathbf{b}_2\cdots\mathbf{b}_n]_i \longrightarrow [d_{i,1}^{w_{i,1}} d_{i,2}^{w_{i,2}} \cdots d_{i,\mathrm{n}}^{w_{i,\mathrm{n}}}]_i \mid 1 \le i \le n\}$. Note that $\mathbf{x}_j$ for $1 \le j \le n$ are expressed as $\mathbf{b}_1\mathbf{b}_2\cdots\mathbf{b}_n$. The results are stored as the distance matrix, also called the dissimilarity matrix, D_{nn},

$$D_{nn} = \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ & \cdots & & \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{pmatrix}. \quad (13)$$

At the beginning, there are $n+1$ membranes in the P system. After the distances are calculated, objects $x_i, d_{\mathrm{i,1}}^{w_{i1}}, d_{\mathrm{i,2}}^{w_{i2}} \cdots d_{\mathrm{i,n}}^{w_{\mathrm{i},n}}$ are placed in membrane i for $1 \le i \le n$. In the next step, the densities of the data points are calculated by the rule $r_2 = \{[d_{i,1}^{w_{i,1}} d_{i,2}^{w_{i,2}} \cdots d_{i,\mathrm{n}}^{w_{i,\mathrm{n}}}]_i \longrightarrow [w_i']_i \mid 1 \le i \le n\}$. Then the send-in and send-out communication rules are used to calculate the values of ρ_i, δ_i, and γ_i and to put them in membrane i for $1 \le i \le n$. Next, according to the sorted results of γ_i for $1 \le i \le n$, the number of clusters k can be determined. The rule of the active membranes is used to split membrane $n+1$ into k membranes as shown in Figure 4. The k cluster centers are put in membranes $n+1$ to $n+k$, respectively. Finally, the remaining data points are divided and each is put into a membrane with a cluster center that is closest to the data point. Up to this point, the clusters are obtained.

The main steps of KST-DPC is summarized as in Algorithm 1.

3.3. Time Complexity Analysis of KST-DPC. As usual, computations in the cells in the tissue-like P system can be implemented in parallel. Because of the parallel implementation, the generation of the dissimilarity matrix uses n computation steps. The generation of the data points densities needs 1 computation step. The calculation of the final density ρ_i uses k computation steps. The calculation of δ_i needs n steps. The calculation of γ_i uses 1 step. $n\log n$ steps are used to sort γ_i for $1 \le i \le n$. Finally, the final clustering needs 1 more computation step. Therefore, the total time complexity of KST-DPC is $n + 1 + k + n + 1 + n\log n + 1 = O(n\log n)$. The time complexity of the DPC-KNN is $O(n^2)$. As compared to DPC-KNN, KST-DPC reduces the time complexity by transferring time complexity to space complexity. The above analysis demonstrates that the overall time complexity of KST-DPC is superior to that of DPC-KNN.

Table 1: Synthetic datasets.

Dataset	Instances	Dimensions	Clusters
Spiral	312	2	3
Compound	7266	2	6
Jain	373	2	2
Aggregation	788	2	7
R15	600	2	15
D31	3100	2	31

4. Test and Analysis

4.1. Data Sources. Experiments on six synthetic datasets and four real-world datasets are carried out to test the performance of KST-DPC. The synthetic datasets are from http://cs.uef.fi/sipu/datasets/. These datasets are commonly used as benchmarks to test the performance of clustering algorithms. The real-world datasets used in the experiments are from the UCI Machine Learning Repository [31]. These datasets are chosen to test the ability of KST-DPC in identifying clusters having arbitrary shapes without being affected by noise, size, or dimensions of the datasets. The numbers of features (dimensions), data points (instances), and clusters vary in each of the datasets. The details of the synthetic and real-world datasets are listed in Tables 1 and 2, respectively.

The performance of KST-DPC was compared with those of the well-known clustering algorithms SC [32], DBSCAN [33], and DPC-KNN [28, 34]. The codes for SC and DBSCAN are provided by their authors. The code of DPC is optimized by using the matrix operation instead of iteration cycle based on the original code provided by Rodriguez and Laio [3] to reduce running time.

The performances of the above clustering algorithms are measured in clustering quality or Accuracy (Acc) and Normalized Mutual Information (NMI). They are very popular measures for testing the performance of clustering algorithms. The larger the values are, the better the results are. The upper bound of these measures is 1.

4.2. Experimental Results on the Synthetic Datasets. In this subsection, the performances of KST-DPC, DPC-KNN, DBSCAN, and SC are reported on the six synthetic datasets. The clustering results by the four clustering algorithms for the six synthetic datasets are color coded and displayed in two-dimensional spaces as shown in Figures 5–10. The results of

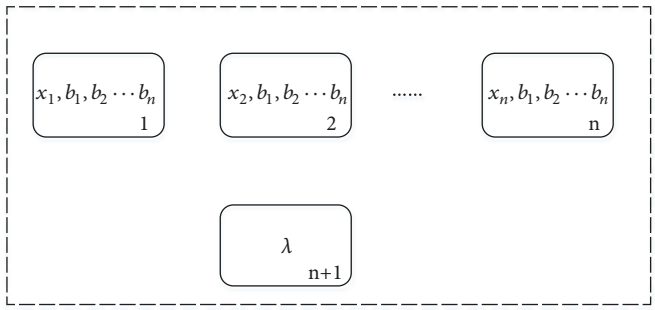

Figure 3: The initial configuration of the tissue-like P system.

Inputs: dataset X, parameter K
Output: Clusters
Step 1: The objects $\mathbf{x}_i, \mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ are in membrane i for $1 \le i \le n$ and object λ is in membrane $n + 1$;
Step 2: Compute the Euclidean distance matrix w_{ij} by the rule1;
Step 3: Compute the local densities of the data points by the rule2 and normalize them using (10) and (11);
Step 4: Calculate ρ_i and δ_i for data point i using (12) and (4) in every membrane i, respectively;
Step 5: Calculate $\gamma_i = \rho_i \times \delta_i$ for all $1 \le i \le n$ in membrane i and sort them by descend, and select the top K values as the initial cluster center. So as to determine the centers of the clusters;
Step 6: Split the membrane $n + 1$ to K membranes by the division rules, which membranes can be number from $n + 1$ to $n + k$;
Step 7: The K clustering centers are put in membranes $n + 1$ to $n + k$, respectively.
Step 8: Assign each remaining point to the membrane with the nearest cluster center;
Step 9: Return the clustering result.

Algorithm 1:

Table 2: Real-world datasets.

Dataset	Instances	Dimensions	Clusters
Vertebral	310	7	2
Seeds	210	7	3
Breast cancer	699	10	2
Banknotes	1372	5	2

the four clustering algorithms on a dataset are shown as four parts in a single figure. The cluster centers of the KST-DPC and DPC-KNN algorithms are marked in the figures with different colors. For DBSCAN, it is not meaningful to mark the cluster centers because they are chosen randomly. Each clustering algorithm ran multiple times on each dataset and the best result of each clustering algorithm is displayed.

The performance measures of the four clustering algorithms on the six synthetic datasets are reported in Table 3. In Table 3, the column "Par" for each algorithm is the number of parameters the users need to set. KST-DPC and DPC-KNN have only one parameter K, which is the number of nearest neighbors to be prespecified. In this paper, the value of K is determined by the percentage of the data points. It references the method in [34]. For each dataset, we adjust the percentage of data points in the KNN for the multiple times and find the optimal percentage that can make the final clustering reach the best. Because we perform more experiments, we only list the best result in Tables 3 and 4. And in order to be consistent wi other parameters in the table, we directly convert the percentage of data points into specific K values. DBSCAN has two input parameters, the maximum radius *Eps* and the minimum point *MinPts*. The SC algorithm needs the true number of clusters. C1 in Table 3 refers to the number of cluster centers found by the algorithms. The performance measures including Acc and NMI are presented in Table 3 for the four clustering algorithms on the six synthetic datasets.

The Spiral dataset has 3 clusters with 312 data points embracing each other. Table 3 and Figure 5 show that KST-DPC, DPC-KNN, DBSCAN, and SC can all find the correct number of clusters and get the correct clustering results. All the benchmark values are 1.00 reflecting the four algorithms all performing perfectly well on the Spiral dataset.

The Compound dataset has 6 clusters with 399 data points. From Table 3 and Figure 6, it is obvious that KST-DPC can find the ideal clustering result; DBSCAN cannot find the right clusters whereas DPC-KNN and SC cannot find the clustering centers. Because DPC has a special assignment strategy [3], it may assign data points erroneously to clusters

Table 3: Results on the synthetic datasets.

Algorithm	Par	C1	Acc	NMI	Algorithm	Par	C1	Acc	NMI
Spiral					*Compound*				
KST-DPC	16	3	1.00	1.00	KST-DPC	217	6	0.98	0.95
DPC-KNN	20	3	1.00	1.00	DPC-KNN	360	6	0.6466	0.7663
DBSCAN	1.2/3	3	1.00	1.00	DBSCAN	1.5/3	5	0.8596	0.9429
SC	3	3	1.00	1.00	SC	6	6	0.6015	0.7622
Jain					*Aggregation*				
KST-DPC	4	2	1.00	1.00	KST-DPC	40	7	1.00	1.00
DPC-KNN	8	2	0.9035	0.5972	DPC-KNN	40	7	0.9987	0.9957
DBSCAN	2.62/4	2	1.00	1.00	DBSCAN	1.59/3	5	0.8274	0.8894
SC	2	2	1.00	1.00	SC	7	7	0.9937	0.9824
R15					*D31*				
KST-DPC	20	15	1.00	0.99	KST-DPC	25	31	1.0000	1.0000
DPC-KNN	20	15	1.00	0.99	DPC-KNN	25	31	0.9700	0.9500
DBSCAN	0.4/5	13	0.78	0.9155	DBSCAN	0.46/3	27	0.6516	0.8444
SC	15	15	0.9967	0.9942	*SC*	31	31	0.9765	0.9670

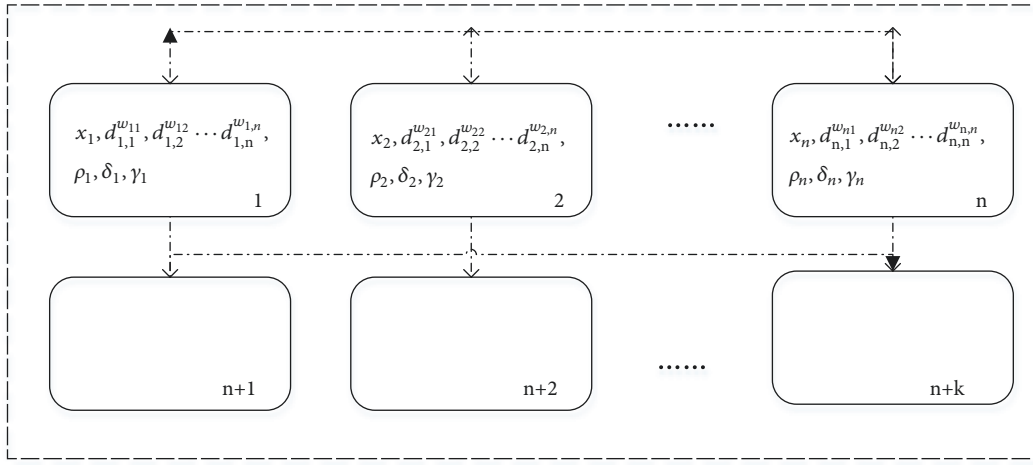

Figure 4: The tissue-like membrane system in the calculation process.

once a data point with a higher density is assigned to an incorrect cluster. For this reason, some data points belonging to cluster 1 are incorrectly assigned to cluster 2 or 3 as shown in Figures 6(b)–6(d). DBSCAN has some prespecified parameters that can have heavy effects on the clustering results. As shown in Figure 6(c), two clusters are merged into one cluster in two occasions. KST-DPC obtained Acc and NMI values higher than those obtained by the other algorithms.

The Jain dataset has two clusters with 373 data points in a 2 dimensional space. The clustering results show that KST-DPC, DBSCAN, and SC can get correct results and both of the benchmark values are 1.00. The experimental results of the 4 algorithms are shown in Table 3 and the clustering results are displayed in Figure 7. DPC-KNN devides some points that should belong to the bottom cluster into the upper cluster. Although all the four clustering algorithms can find the correct number of clusters, KST-DPC, DBSCAN, and SC are more effective because they can put all the data points into the correct clusters.

The Aggregation dataset has 7 clusters with different sizes and shapes and two pairs of clusters connected to each other. Figure 8 shows that both the KST-DPC and DPC-KNN algorithms can effectively find the cluster centers and correct clusters, except that an individual data point is put into an incorrect cluster by DPC-KNN. Table 3 shows that the benchmark values of KST-DPC are all 1.00 and those of DPC-KNN are close to 1.00. SC also can recognize all clusters, but the values of Acc and NMI are lower than those of DPC-KNN. DBSCAN did not find all clusters and could not partition the clusters connected to each other.

The R15 dataset has 15 clusters containing 600 data points. The clusters are slightly overlapping and are distributed randomly in a 2-dimensional space. One cluster lays in the

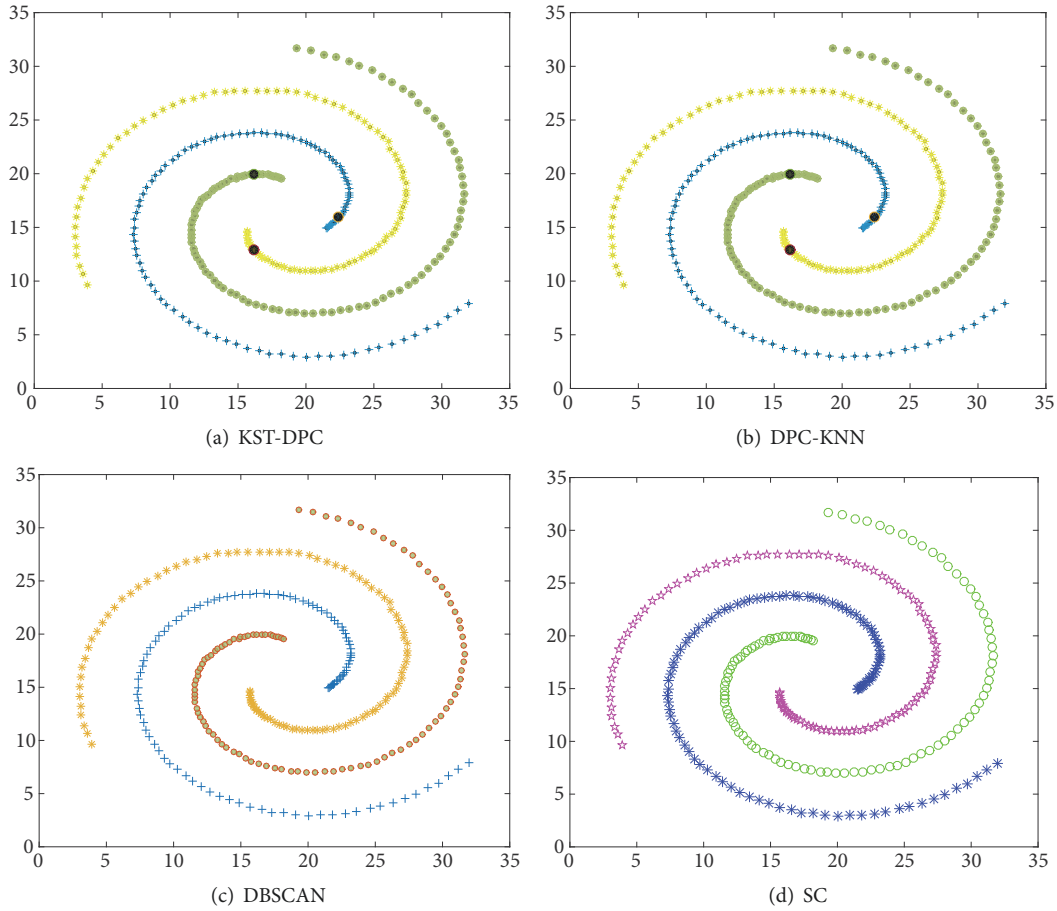

Figure 5: Clustering results of the Spiral dataset by the four clustering algorithms.

center of the 2-dimensional space and is closely surrounded by seven other clusters. The experimental results of the 4 algorithms are shown in Table 3 and the clustering results are displayed in Figure 9. KST-DPC and DPC-KNN can both find the correct cluster centers and assign almost all data points to their corresponding clusters. SC also obtained good experimental result, but DBSCAN did not find all clusters.

The D31 dataset has 31 clusters and contains 3100 data points. These clusters are slightly overlapping and distribute randomly in a 2-dimensional space. The experimental results of the 4 algorithms are shown in Table 3 and the clustering results are displayed in Figure 10. The values of Acc and NMI obtained by KST-DPC are all 1.00. This shows that KST-DPC obtained perfect clustering results on the D31 dataset. DPC and SC obtained similar results to those of KST-DPC on this dataset, but DBSCAN was not able to find all clusters.

4.3. Experimental Results on the Real-World Datasets. This subsection reports the performances of the clustering algorithms on the four real-world datasets. The varying sizes and dimensions of these datasets are useful in testing the performance of the algorithms under different conditions.

The number of clusters, Acc and NMI are also used to measure the performances of the clustering algorithms on these real-world datasets. The experimental results are reported in Table 4 and the best results of the each dataset are shown in italic. The symbol "--" indicates there is no value for that entry.

The Vertebral dataset consists of 2 clusters and 310 data points. As Table 4 shows, the value of Acc got by KST-DPC is equal to that got by DPC-KNN, but the value of NMI got by KST-DPC is lower than that got by DPC-KNN. No values of Acc and NMI were obtained by SC. As Table 4 shows, all algorithms could find the right number of clusters.

The Seeds dataset consists of 210 data points and 3 clusters. Results in Table 4 show that KST-DPC obtained the best, whereas DBSCAN obtained the worst, values of Acc and NMI. It is obvious that all four clustering algorithms could get the right number of clusters.

The Breast Cancer dataset consists of 699 data points and 2 clusters. The results on this dataset in Table 4 show that all four clustering algorithms could find the right number of clusters. KST-DPC obtained the Acc and NMI values of 0.8624 and 0.4106, respectively, which are higher than those obtained by other clustering algorithms. The results also show that DBSCAN has the worst performance on this dataset, except that SC did not get experimental results on these benchmarks.

The Banknotes dataset consists of 1372 data points and 2 clusters. From Table 4, it is obvious that KST-DPC got the best

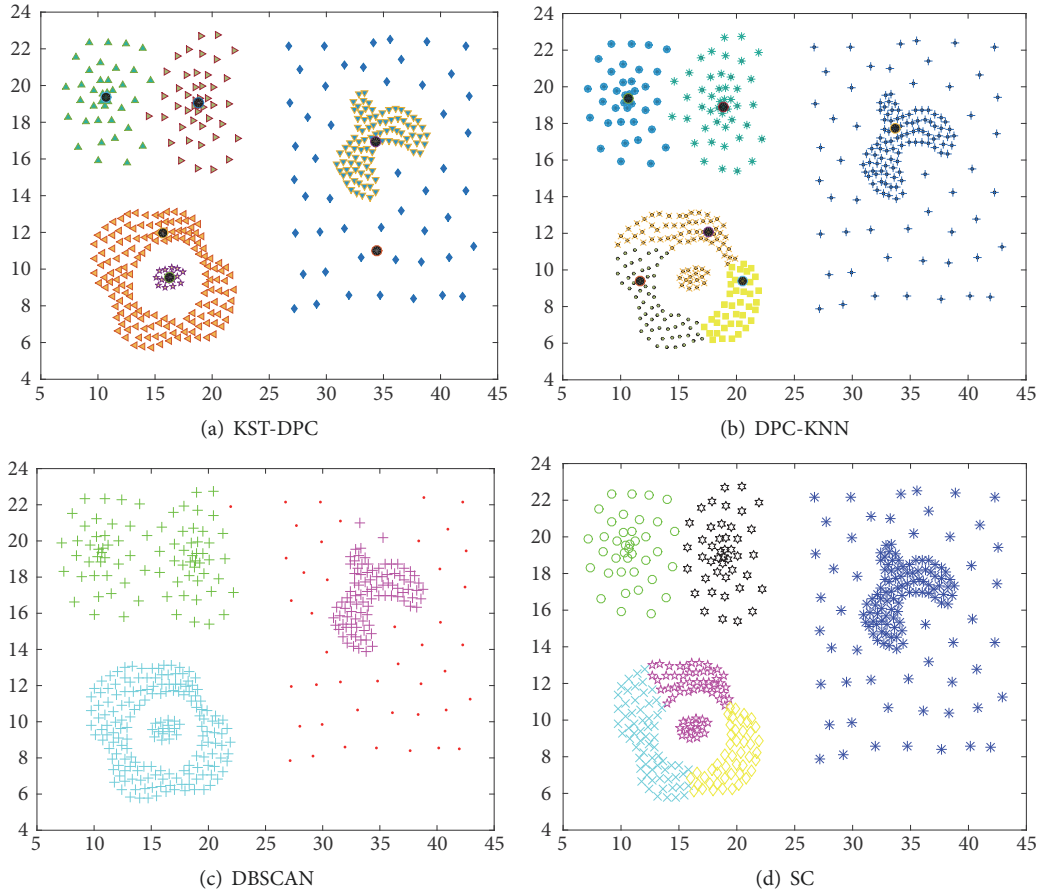

Figure 6: Clustering results of the Compound dataset by the four clustering algorithms.

Table 4: Results on the real-world datasets.

Algorithm	Par	Cl	Acc	NMI	Algorithm	Par	Cl	Acc	NMI
Vertebral					*Seeds*				
KST-DPC	9	2	*0.6806*	0.0313	KST-DPC	4	3	*0.8429*	*0.6574*
DPC-KNN	9	2	*0.6806*	*0.0821*	DPC-KNN	6	3	0.8143	0.6252
DBSCAN	7/48	2	0.6742	--	DBSCAN	0.92/7	3	0.5857	0.4835
SC	2	2	--	--	SC	3	3	0.6071	0.5987
Breast cancer					*Banknotes*				
KST-DPC	70	2	*0.8624*	*0.4106*	KST-DPC	68	2	*0.8434*	*0.7236*
DPC-KNN	76	2	0.7954	0.3154	DPC-KNN	82	2	0.7340	0.3311
DBSCAN	6/20	2	0.6552	0.0872	DBSCAN	6.5/5	2	0.5554	6.7210e-16
SC	2	2	--	--	SC	2	2	0.6152	0.0598

values of Acc and NMI among all four clustering algorithms. The values of Acc and NMI obtained by KST-DPC are 0.8434 and 0.7260, respectively. Larger values of these benchmarks indicate that the experimental results obtained by KST-DPC are closer to the true results than those obtained by the other clustering algorithms.

All these experimental results show that KST-DPC outperform the other clustering algorithms. It obtained larger values of Acc and NMI than the other clustering algorithms.

5. Conclusion

This study proposed a density peak clustering algorithm based on the K-nearest neighbors, Shannon entropy and tissue-like P systems. It uses the K-nearest neighbors and Shannon entropy to calculate the density metric. This algorithm overcomes the shortcoming that DPC has that is to set the value of the cutoff distance d_c in advance. The tissue-like P system is used to realize the clustering process. The analysis demonstrates that the overall time taken by KST-DPC is

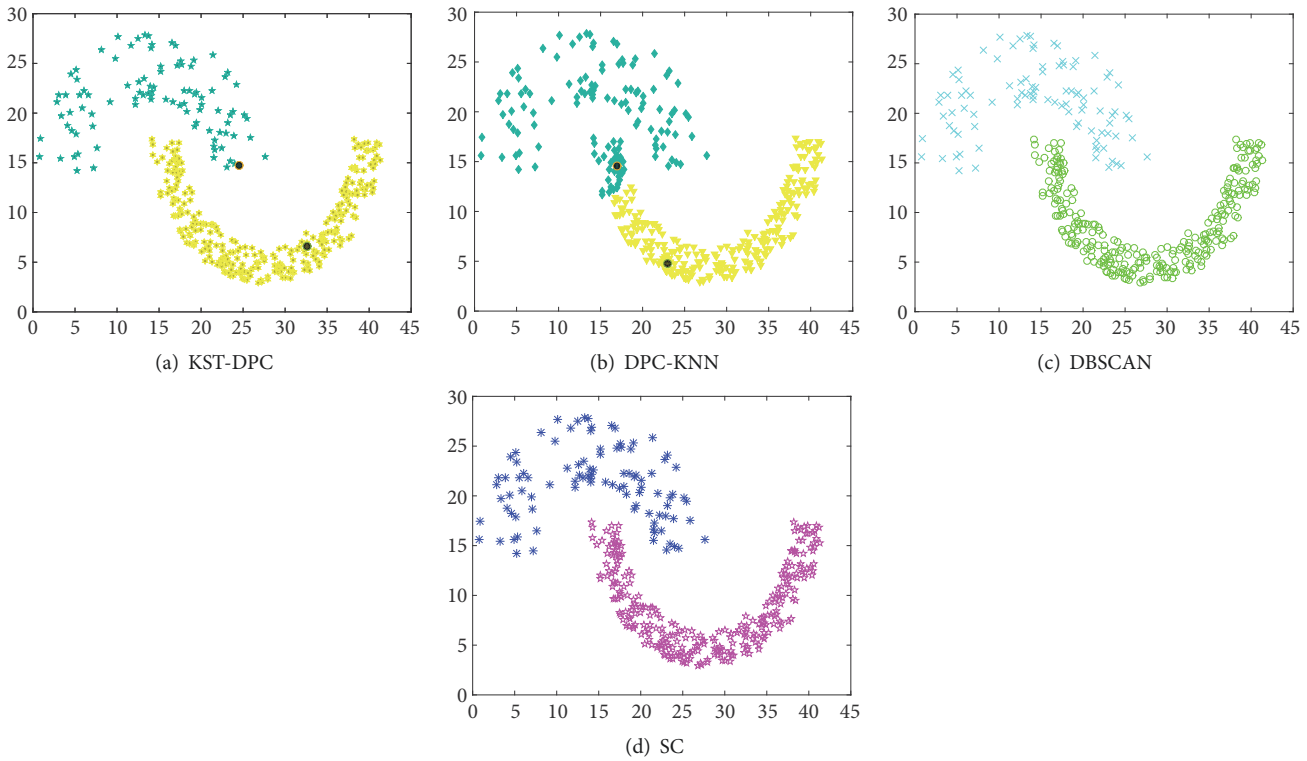

Figure 7: Clustering results of the Jain dataset by the four clustering algorithms.

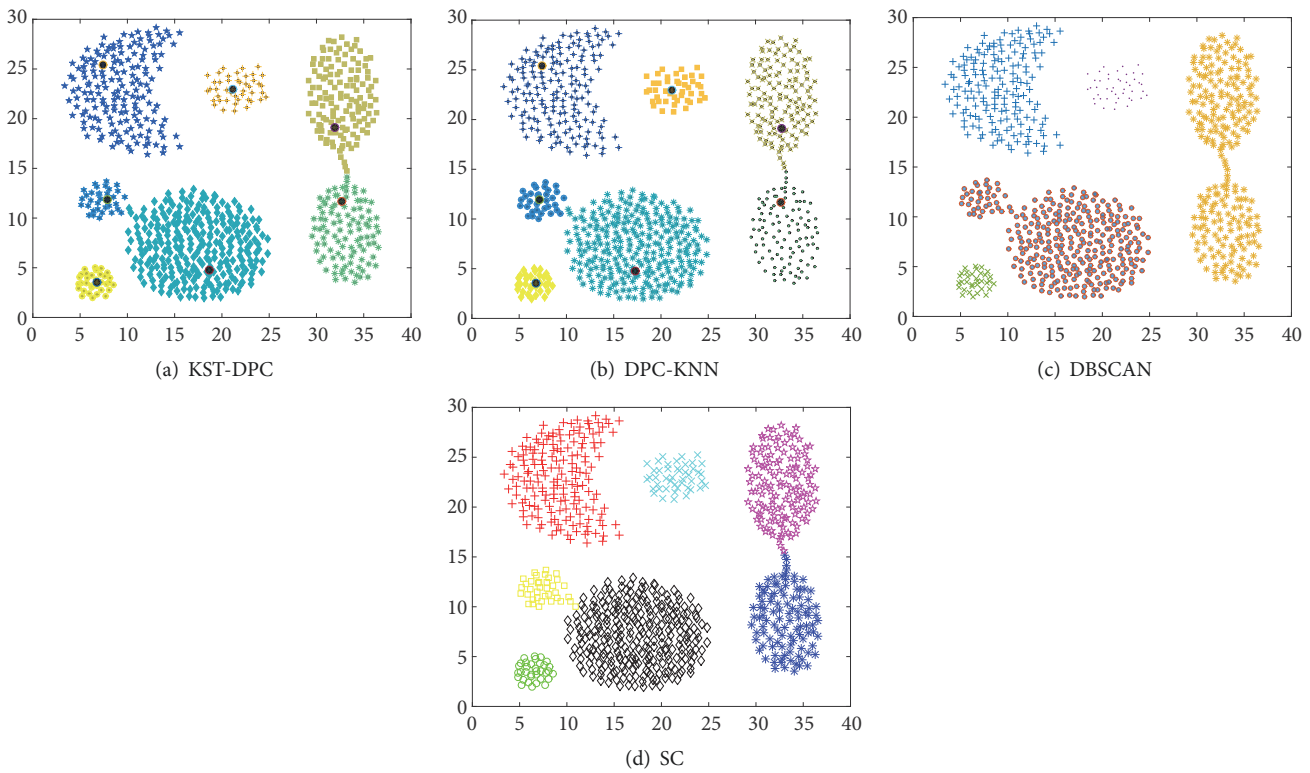

Figure 8: Clustering results of the Aggregation dataset by the four clustering algorithms.

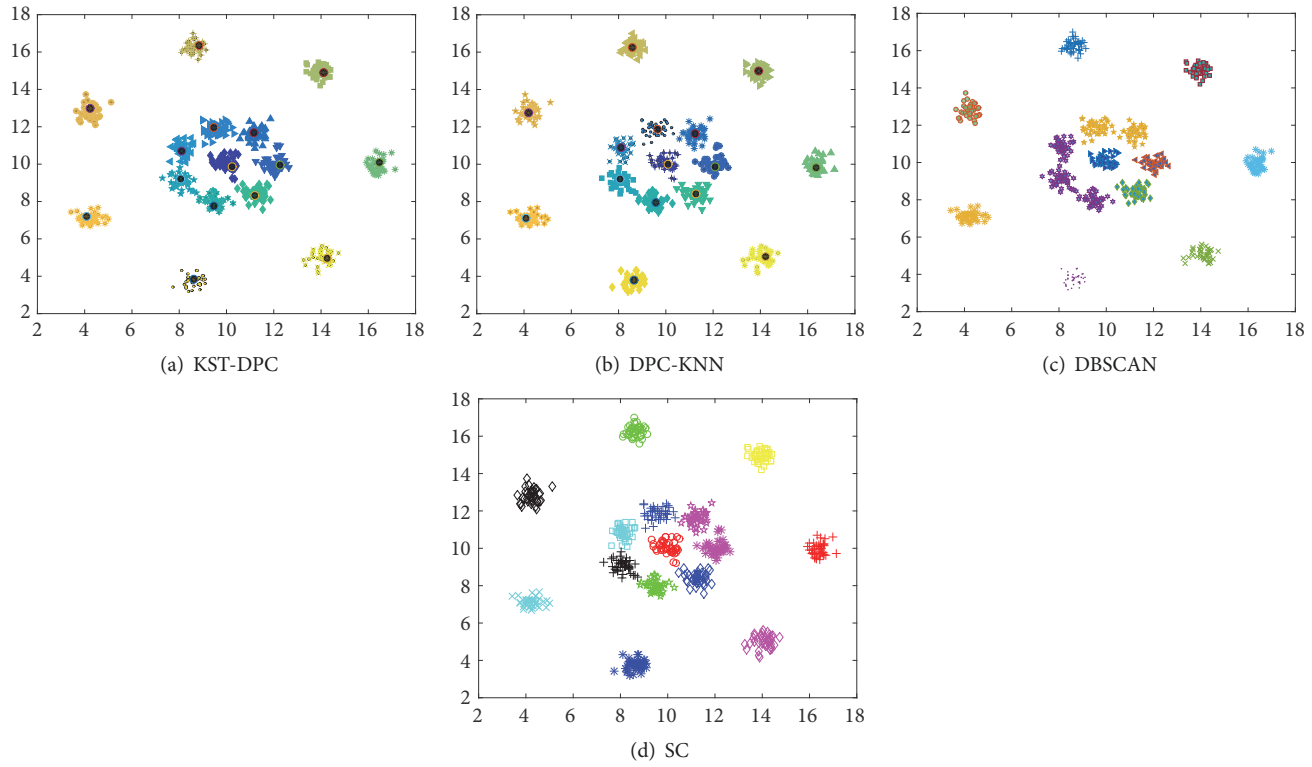

Figure 9: Clustering results of the R15 dataset by the four clustering algorithms.

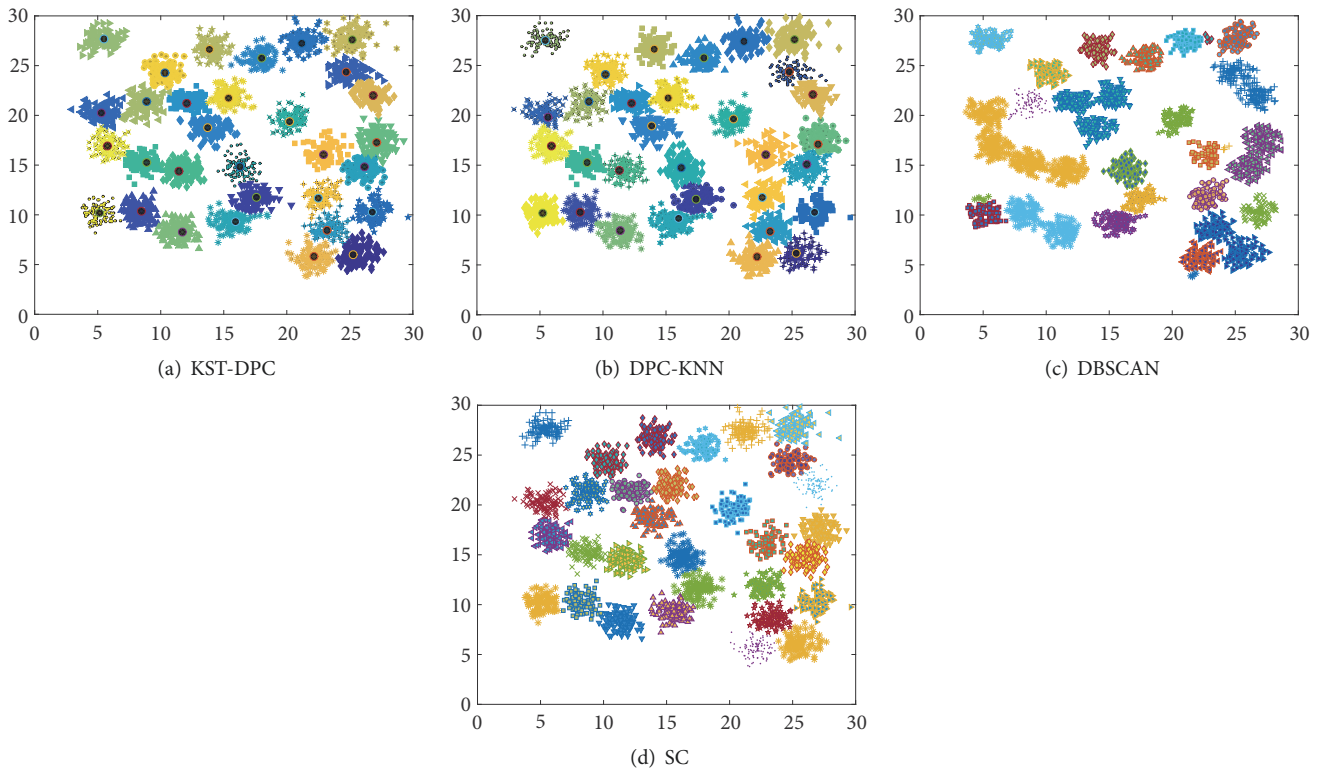

Figure 10: Clustering results of the D31 dataset by the four clustering algorithms.

shorter than those taken by DPC-KNN and the traditional DPC. Synthetic and real-world datasets are used to verify the performance of the KST-DPC algorithm. Experimental results show that the new algorithm can get ideal clustering results on most of the datasets and outperforms the three other clustering algorithms referenced in this study.

However, the parameter K in the K-nearest neighbors is prespecified. Currently there is no technique available to set this value. Choosing a suitable value for K is a future research direction. Moreover, some other methods can be used to calculate the densities of the data points. In order to improve the effectiveness of DPC, some optimization techniques can also be employed.

Data Availability

The synthetic datasets are available at http://cs.uef.fi/sipu/datasets/ and the real-world datasets are available at http://archive.ics.uci.edu/ml/index.php.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (nos. 61876101, 61802234, and 61806114), the Social Science Fund Project of Shandong (16BGLJ06, 11CGLJ22), China Postdoctoral Science Foundation Funded Project (2017M612339, 2018M642695), Natural Science Foundation of the Shandong Provincial (ZR2019QF007), China Postdoctoral Special Funding Project (2019T120607), and Youth Fund for Humanities and Social Sciences, Ministry of Education (19YJCZH244).

References

[1] J. Han, J. Pei, and M. Kamber, *Data Mining: Concepts and Techniques*, San Francisco, CA, USA, 3rd edition, 2011.

[2] R. J. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," in *Advances in Knowledge Discovery and Data Mining*, vol. 7819 of *Lecture Notes in Computer Science*, pp. 160–172, Springer, Berlin, Germany, 2013.

[3] A. Laio and A. Rodriguez, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[4] J. Cong, X. Xie, and F. Hu, "A density peak cluster model of high-dimensional data," in *Proceedings of the Asia-Pacific Services Computing Conference*, pp. 220–227, Zhangjiajie, China, 2016.

[5] X. Xu, S. Ding, M. Du, and Y. Xue, "DPCG: an efficient density peaks clustering algorithm based on grid," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 5, pp. 743–754, 2016.

[6] R. Bie, R. Mehmood, S. Ruan, Y. Sun, and H. Dawood, "Adaptive fuzzy clustering by fast search and find of density peaks," *Personal and Ubiquitous Computing*, vol. 20, no. 5, pp. 785–793, 2016.

[7] M. Du, S. Ding, X. Xu, and X. Xue, "Density peaks clustering using geodesic distances," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 8, pp. 1–15, 2018.

[8] M. Du, S. Ding, and Y. Xue, "A robust density peaks clustering algorithm using fuzzy neighborhood," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 7, pp. 1131–1140, 2018.

[9] J. Hou and H. Cui, "Density normalization in density peak based clustering," in *Proceedings of the International Workshop on Graph-Based Representations in Pattern Recognition*, pp. 187–196, Anacapri, Italy, 2017.

[10] X. Xu, S. Ding, H. Xu, H. Liao, and Y. Xue, "A feasible density peaks clustering algorithm with a merging strategy," *Soft Computing*, vol. 2018, pp. 1–13, 2018.

[11] R. Liu, H. Wang, and X. Yu, "Shared-nearest-neighbor-based clustering by fast search and find of density peaks," *Information Sciences*, vol. 450, pp. 200–226, 2018.

[12] M. Du, S. Ding, Y. Xue, and Z. Shi, "A novel density peaks clustering with sensitivity of local density and density-adaptive metric," *Knowledge and Information Systems*, vol. 59, no. 2, pp. 285–309, 2019.

[13] G. Paun, "A quick introduction to membrane computing," *Journal of Logic Algebraic Programming*, vol. 79, no. 6, pp. 291–294, 2010.

[14] H. Peng, J. Wang, and P. Shi, "A novel image thresholding method based on membrane computing and fuzzy entropy," *Journal of Intelligent & Fuzzy Systems Applications in Engineering & Technology*, vol. 24, no. 2, pp. 229–237, 2013.

[15] M. Tu, J. Wang, H. Peng, and P. Shi, "Application of adaptive fuzzy spiking neural P systems in fault diagnosis of power systems," *Journal of Electronics*, vol. 23, no. 1, pp. 87–92, 2014.

[16] J. Wang, P. Shi, H. Peng, M. J. Perez-Jimenez, and T. Wang, "Weighted fuzzy spiking neural P systems," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 2, pp. 209–220, 2013.

[17] B. Song, C. Zhang, and L. Pan, "Tissue-like P systems with evolutional symport/antiport rules," *Information Sciences*, vol. 378, pp. 177–193, 2017.

[18] H. Peng, J. Wang, M. J. Pérez-Jiménez, and A. Riscos-Núñez, "Dynamic threshold neural P systems," *Knowledge-Based Systems*, vol. 163, pp. 875–884, 2019.

[19] L. Huang, I. H. Suh, and A. Abraham, "Dynamic multi-objective optimization based on membrane computing for control of time-varying unstable plants," *Information Sciences*, vol. 181, no. 11, pp. 2370–2391, 2011.

[20] H. Peng, Y. Jiang, J. Wang, and M. J. Pérez-Jiménez, "Membrane clustering algorithm with hybrid evolutionary mechanisms," *Journal of Software. Ruanjian Xuebao*, vol. 26, no. 5, pp. 1001–1012, 2015.

[21] H. Peng, J. Wang, M. J. Pérez-Jiménez, and A. Riscos-Núñez, "The framework of P systems applied to solve optimal watermarking problem," *Signal Processing*, vol. 101, pp. 256–265, 2014.

[22] G. Zhang, J. Cheng, M. Gheorghe, and Q. Meng, "A hybrid approach based on different evolution and tissue membrane systems for solving constrained manufacturing parameter optimization problems," *Applied Soft Computing*, vol. 13, no. 3, pp. 1528–1542, 2013.

[23] H. Peng, P. Shi, J. Wang, A. Riscos-Núñez, and M. J. Pérez-Jiménez, "Multiobjective fuzzy clustering approach based on tissue-like membrane systems," *Knowledge-Based Systems*, vol. 125, pp. 74–82, 2017.

[24] H. Peng, J. Wang, M. J. Pérez-Jiménez, and A. Riscos-Núñez, "An unsupervised learning algorithm for membrane computing," *Information Sciences*, vol. 304, pp. 80–91, 2015.

[25] H. Peng, J. Wang, P. Shi, M. J. Pérez-Jiménez, and A. Riscos-Núñez, "An extended membrane system with active membranes to solve automatic fuzzy clustering problems," *International Journal of Neural Systems*, vol. 26, no. 3, pp. 1–17, 2016.

[26] H. Peng, J. Wang, J. Ming et al., "Fault diagnosis of power systems using intuitionistic fuzzy spiking neural P systems," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 4777–4784, 2018.

[27] X. Liu, Y. Zhao, and M. Sun, "An improved apriori algorithm based on an evolution-communication tissue-like P System with promoters and inhibitors," *Discrete Dynamics in Nature and Society*, vol. 2017, pp. 1–11, 2017.

[28] X. Liu and J. Xue, "A cluster splitting technique by hopfield networks and P systems on simplices," *Neural Processing Letters*, vol. 46, no. 1, pp. 171–194, 2017.

[29] Y. Zhao, X. Liu, and W. Wang, "Spiking neural P systems with neuron division and dissolution," *PLoS ONE*, vol. 11, no. 9, Article ID e0162882, 2016.

[30] M. Du, S. Ding, and H. Jia, "Study on density peaks clustering based on k-nearest neighbors and principal component analysis," *Knowledge-Based Systems*, vol. 99, no. 1, pp. 135–145, 2016.

[31] K. Bache and M. Lichman, UCI machine learning repository, 2013. http:// archive.ics.uci.edu/ml.

[32] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: analysis and an algorithm," in *Advances in Neural Information Processing Systems*, pp. 849–856, Vancouver, British Columbia, Canada, 2001.

[33] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pp. 226–231, Menlo Park, Portland, USA, 1996.

[34] L. Yaohui, M. Zhengming, and Y. Fang, "Adaptive density peak clustering based on K-nearest neighbors with aggregating strategy," *Knowledge-Based Systems*, vol. 133, pp. 208–220, 2017.