IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 26, 2020, accepted March 29, 2020, date of publication April 13, 2020, date of current version April 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2987345

Clustering by Using the Way of Atomic Fission

SHIZHAN LU[1], LONGSHENG CHENG[1], AND RASHID MEHMOOD[2,3]

[1]School of Economics and Management, Nanjing University of Science and Technology, Nanjing 210094, China
[2]Department of Computer Sciences and Information Technology, University of Kotli Azad Jammu and Kashmir, Kotli 14400, Pakistan
[3]School of Oncology and Pathology, Karolinska Institute, 17177 Stockholm, Sweden

Corresponding author: Longsheng Cheng (chenglongshengnj@163.com; cheng_longsheng@njust.edu.cn)

ABSTRACT Cluster analysis, which focuses on the grouping and categorization of similar elements, is widely used in various fields of research. Inspired by the phenomenon of atomic fission, this paper proposes a novel density-based clustering algorithm, called fission clustering (FC). It focuses on mining the dense families of clusters in the dataset and utilizes the information of the distance matrix to fissure the dataset into subsets. A K-nearest neighbor (KNN) local density indicator is applied to identify and remove the points of sparse areas so as to obtain a dense subset that consists of the dense families of clusters. The algorithm, denoted as FC-KNN, is achieved by merging FC and KNN local density indicator. Several frequently-used datasets were applied to test the performance of the proposed clustering approach and to compare the results with those of other algorithms. The comprehensive comparisons indicate that the proposed method has advantages over other common methods.

INDEX TERMS Clustering, density-based, K-nearest neighbor, fission clustering algorithm.

I. INTRODUCTION

The data clustering processes used in numerous current clustering methods are similar to those of atomic fusion. In contrast, we propose a method to cluster data by the pattern of atomic fission. The proposed method can cluster data category by category without assuming that the number of categories is known before clustering occurs.

In data clustering, the basic task is to divide data into distinct groups on the basis of their similarity. Initial methods of clustering tended to focus on finding the center point of every category and then assigning the other points to the nearest center. To make computer cluster data faster, some researchers, such as Schikuta [1], Ma and Chow [2] et al., have applied the grid-based clustering method to divide objects part by part. The grid-based clustering method does not need to cluster data point by point; however, this method is influenced by the size of grid cells and cannot easily determine the number of categories.

A fundamental and challenging task of clustering analysis is to determine the number of clusters. This number is however assumed known in the earlier research on clustering. A clustering approach with few known conditions is expected when we face increasing numbers of poor information datasets (scant or incomplete data). The similarity matrix of objects is the unique known condition in our method.

The associate editor coordinating the review of this manuscript and approving it for publication was Jenny Mahoney.

Inspired by the phenomenon and rapid process of atomic fission, a fast and effective clustering method is proposed, which we call fission clustering (FC). If the distances between every pair of clusters are large enough, two maximal values are applied to determine the number of categories, i.e., ''the maximal crack of the distance matrix and the maximal value of all the distances between objects and their nearest neighbors''. Otherwise, the K-nearest neighbor method can be applied to obtain a local density indicator for every object in the clustering dataset. Then, the objects that have a small indicator value will be removed, while a dense subset with large distances between every two clusters is obtained.

Border points are distributed in two cases: (*i*) the border points of the *i*th cluster are far away from the border points of the *j*th cluster ($i \neq j$), and (*ii*) the border points of different clusters are close together. The main works in this article can be described as follows: (a) propose the FC algorithm for case (*i*); (b) combine the FC algorithm and the K-nearest neighbor local density indicator to propose the FC-KNN algorithm for case (*ii*); and (c) demonstrate our algorithms by some numerical experiments of both simulated and real datasets.

II. RELATED WORK

Clustering, a classical issue in data mining, is widely used in a number of different areas, such as climate research [3], computational biology, biophysics and

bioinformatics [4], [5], economics and finance [6], [7], and neuroscience [8], [9].

In general, different clustering methods can be basically categorized as follows: density-based (DBSCAN [10], NQ-DBSCAN [11], OPTICS [12], DP [13], DP-HD [14] and CSSub [15]); grid-based (DGB [16], STING [17], CLIQUE [18] and WaveCluster [19]); model-based (Gaussian mixture models [20], COBWEB [21] and Latent tree models [22]); partitioning (K-means [23], CLARANS [24], and TLBO [25]); graph-based (GRAPHCLUS [26], ProClust [27] and MCSSGC [28]); and hierarchical (DIANA [29], BIRCH [30] and CHAMELEON [31]) approaches.

Of the earlier methods in the literature, a most representative clustering method may be K-means [23], which focuses on determining K centers and dividing data points into K clusters. However, K-means and its variants (see [32], [33]) need to know the number of categories before clustering occurs. More recently, a fast algorithm by finding density peaks (DP) was proposed [13] and widely used. DP can scan its decision graph to determine the number of clusters automatically, and the experimenter can also select some core points of the decision graph as centers when the number of clusters is known. DP combines the advantages of both density-based and centroid-based clustering methods. Many variants have been developed by using DP, such as ADPC [34], GDPC [35], FastDPeak [36], REDPC [37], FREDPC [38], DPC-KNN-PCA [39] and SNN-DPC [40], to list a few. As a local density-based method, DP can obtain good results in most instances. However, as a centroid-based method, DP and its variants cannot cluster points correctly when a category has more than one center.

Schikuta [1] designed a grid structure in the data distribution area to partition data into blocks, and then applied the block information via the index structure of the grid cell and clustered the objects according to their surrounding blocks. Typical examples of this type of algorithm include STING [17], CLIQUE [18] and WaveCluster [19]. The grid-based clustering approach does not need to input the number of clusters, and it considers cells rather than data points, so it can deal with cases in which a category has more than one center. However, these grid-based methods are hard to be applied to high-dimensional datasets as the number of the cells in the grid grows exponentially with the dimensionality of the data.

DBSCAN [10] is a representative density-based algorithm which does not need to input the number of clusters. It determines clusters by defining the density criterion with two parameters, Eps-distance and MinPts. NQ-DBSCAN [11], ReCon-DBSCAN [41], AA-DBSCAN [42] and RNN-DBSCAN [43] are some up-to-date developments of DBSCAN. In particular, NQ-DBSCAN is a high-efficiency algorithm for high-dimensional data. However, DBSCAN and its extensions are difficult for their parameters to be set, which are ruleless (as shown in TABLE 5) on account of the different densities for variant datasets.

TABLE 1. The full names of the algorithm abbreviations.

DBSCAN	Density based spatial clustering of applications with noise
NQ-DBSCAN	Neighbor query-DBSCAB
OPTICS	Ordering points to identify the clustering structure
DP	Fast search and find of density peaks
DP-HD	Fast search and find of density peaks via heat diffusion
CSSub	Clustering by shared subspaces
DGB	Density and grid based clustering method
STING	Statistical information grid-based method
CLIQUE	Clustering in quest
CLARANS	Clustering large applications based on randomized search
TLBO	Teaching learning based optimization
ProClust	Clustering of protein sequences
MCSSGC	Must-link and cannot-link constraints semi-supervised graph based clustering
DIANA	Divisive analysis
BIRCH	Balanced iterative reducing and clustering using hierarchies
ADPC	Search in descending order and automatic find of density peaks
GDPC	Gravitation-based density peaks clustering algorithm
FastDPeak	Fast density peak clustering for large scale data based on kNN
REDPC	Residual error-based density peak clustering algorithm
FREDPC	Feasible residual error-based density peak clustering algorithm
DPC-KNN-PCA	Density peaks clustering based on k nearest neighbors and principal component analysis
SNN-DPC	Shared-nearest-neighbor-based clustering by fast search and find of density peaks algorithm
ReCon-DBSCAN	Recondition DBSCAN
AA-DBSCAN	Approximate adaptive DBSCAN
RNN-DBSCAN	DBSCAN with reverse nearest neighbor density estimates
AP	Affinity propagation
NKGA	NK hybrid genetic algorithm

In this article, we are proposing a method which will not need to set the number of clusters as an input. Neither will it be impacted by the data dimensionality like grid-based methods. Our method focuses on mining the dense families, rather than the center points, of the dataset, so it can also overcome the inadequacy of centroid-based methods, that is, it can cluster data points correctly when a category has more than one center. For this reason, our method is more robust than the related methods as a comparison, with its parameters easier to be set than those in DGB and NQ-DBSCAN.

III. PROPOSED METHODS

In general, a cluster center is surrounded by neighbors with lower local densities, and is at a relatively large distance from other cluster centers [13]. Based on this feature, we can make an algorithm assumption that there are $k-1$ neighbourhoods $U(x_i, r_i)$ composed of higher local density points in the dataset of k categories. This assumption is satisfied in many existing simulated and real datasets.

Section A and B below are dealing with the datasets of the case (*i*) and (*ii*), respectively. Section A proposes the FC algorithm and describes two key steps of that algorithm: (a) splitting the dataset into subsets and (b) stopping splitting sets. Section B includes two parts: (a) using equation (1) to obtain an indicator for every object in the dataset X, presenting Algorithm 2 and then applying Algorithm 2 to obtain a subset $C \subset X$, which has the feature of case (*i*); and (b) proposing the FC-KNN algorithm.

A. FISSION CLUSTERING ALGORITHM (FC)

In this section, we address case (*i*) first. To develop the algorithm, we first give a definition needed below.

Definition: $f : X \times X \to R$ is a distance (similarity) function, where X is a sample set, and R is the real number set. For all $x_k \in X$, if $f(x_0, x_k) \notin (f(x_0, x_i), f(x_0, x_j))$ (or $f(x_0, x_k) \notin (f(x_0, x_j), f(x_0, x_i))$), we call $|f(x_0, x_i) - f(x_0, x_j)|$ a crack of (X, f), where $x_0, x_i, x_j \in X$.

Obviously, the maximal crack (MC) of (X, f) exists for a finite dataset.

The key steps of the FC algorithm are to fissure a dataset into two subsets and to stop fissuring subsets when all the clusters are obtained. These two key steps are detailed as follows.

1) DIVIDE DATASETS

For a distance (similarity) function $f(x_i, x_j)$ between x_i and x_j, we define $f(x_i, x_j) < f(x_i, x_k)$ if the relationship between x_i and x_j is closer than that between x_i and x_k. Then the distance (similarity) matrix of (X, f) can be easily obtained, denoted by $S(X)$. The matrix $S_1(X)$ is obtained by sorting every row of the distance matrix $S(X)$. The ith column of $S_1(X)$ is subtracted from the $(i+1)$th column of $S_1(X)$ to obtain the ith column of the matrix $S_2(X)$, $MC = max\{s_{ij} : s_{ij} \in S_2(X)\}$. If $MC = |f(x_i, x_j) - f(x_i, x_k)|$ and $f(x_i, x_t) \leqslant min\{f(x_i, x_j), f(x_i, x_k)\}$, then $x_t \in X_1$; otherwise, $x_t \in X_2$, and the set X is fissured into two subsets.

If there are k categories of objects in X, then the k categories can be obtained step by step by application of the above fissuring method.

A toy example to show how to compute the MC is presented as follows. Let $X = \{x_1(0, 0), x_2(0.1, 0), x_3(0, 0.2), x_4(5, 0), x_5(5.2, 0.1), x_6(5.1, 0.3)\}$ and Euclidean distance function be the similarity function.

$$S_1(X) = \begin{bmatrix} 0 & 0.10 & 0.20 & 5.00 & 5.11 & 5.20 \\ 0 & 0.10 & 0.22 & 4.90 & 5.01 & 5.10 \\ 0 & 0.20 & 0.22 & 5.00 & 5.10 & 5.20 \\ 0 & 0.22 & 0.32 & 4.90 & 5.00 & 5.00 \\ 0 & 0.22 & 0.22 & 5.10 & 5.20 & 5.20 \\ 0 & 0.22 & 0.32 & 5.01 & 5.10 & 5.11 \end{bmatrix}$$

$$S_2(X) = \begin{bmatrix} 0.10 & 0.10 & 4.80 & 0.11 & 0.09 \\ 0.10 & 0.12 & 4.68 & 0.11 & 0.09 \\ 0.20 & 0.02 & 4.78 & 0.10 & 0.10 \\ 0.22 & 0.10 & 4.58 & 0.10 & 0 \\ 0.22 & 0 & \underline{4.88} & 0.10 & 0 \\ 0.22 & 0.10 & 4.69 & 0.09 & 0.01 \end{bmatrix}$$

$MC(X) = S_2(X)(5, 3) = 4.88 = |f(x_5, x_3) - f(x_5, x_4)|$, if $f(x_5, x_i) \leqslant min\{f(x_5, x_3), f(x_5, x_4)\}$, then $x_i \in X_1$; otherwise, $x_i \in X_2$. X is fissured into two subsets $X_1 = \{x_4, x_5, x_6\}$ and $X_2 = \{x_1, x_2, x_3\}$.

The mappings $\bigcup_{x_i \in X} \{f(x_i, x_j) : x_j \in X - \{x_i\}\}$ generate many cracks (as the Definition described above). If a set contains two clusters, it must be most reasonable to divide the dataset into two subsets using the maximal crack of all the cracks.

2) STOP DIVIDING DATASETS

In this section, we turn to investigate the characteristics of the distance matrix, and then apply the useful information in the matrix to determine the number of categories.

We use the following formulae as an illustration: let $d_0(C) = max\{f(x_i, \hat{x}_i) : x_i \in C \subset X\}$ and $d_0 = max\{f(x_i, \hat{x}_i) : x_i \in X\}$, where $\hat{x}_i$ is the nearest neighbor of x_i. The object $x_i \in X$ can be considered as a village and $f(x_i, x_j)$ can be considered as the distance between two villages x_i and x_j. Suppose there is a road such that x_i and x_j are connected for all $x_i, x_j \in C$, and the distance of every pair of adjacent connection villages on the road is less than or equal to $d_0(C)$. This road is denoted as $d_0(C)$-road. The theorem below is an effective indicator to determine the number of categories.

Theorem: If the distance function f satisfies triangle inequality and $C \subset X$ has a $d_0(C)$-road, then $MC(C) \leqslant d_0(C)$, where $MC(C)$ is the MC of (C, f).

Proof: Shown as APPENDIX A.

If the distance of every pair of clusters is much greater than d_0, and every cluster has a d_0-road, the inequation $MC(C) \leqslant d_0$ can be considered as the condition under which stop fissuring a subset C. If all the subsets that fissured from X are satisfied by the inequation $MC(C) \leqslant d_0$, then the process of fissuring subsets will stop. The number of clusters will be determined at the same time.

Numerous common distance functions satisfy the triangle inequality, such as the Manhattan distance, Euclidean distance, and Minkowski distance. If the densities of clusters are not extremely different in the same dataset, the inequation is effective.

The details of the FC algorithm are as shown in follows, where $S_k(C)(:, i)$ is the ith column of $S_k(C)$.

B. THE FISSION CLUSTERING ALGORITHM WITH K-NEAREST NEIGHBOR LOCAL DENSITY INDICATOR (FC-KNN)

In this section the main purpose is to obtain a dense subset $C \subset X$ in Case (ii) such that the distances between every pair of clusters in C are large enough but the distances between every pair of nearest neighbors are sufficiently small, and then apply the Algorithm 1 to split the subset C.

1) OBTAIN THE LOCAL DENSITY INDICATOR FOR DENOISING

This subsection aims to obtain a local density indicator ρ_i for every object x_i and then distinguish the dense area objects from the sparse area objects.

KNN-density is a frequently-used indicator to describe the local density indicator ρ_i [36], [39]. Our method focuses on mining the dense families of the dataset. It is more robust than other methods which focus on mining the center points. Then, we select a relatively straightforward and useful

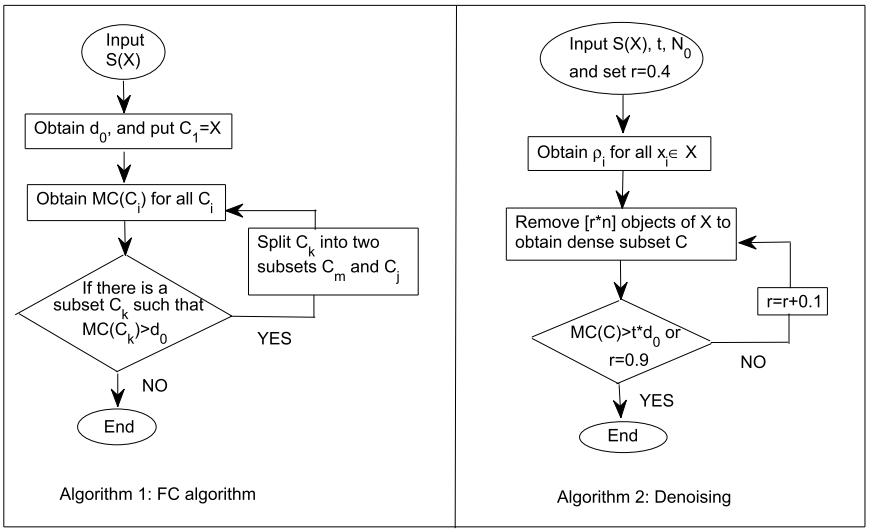

FIGURE 1. The flowcharts of Algorithm 1 and 2.

Algorithm 1 FC algorithm.

Input: Distance matrix $S(X)$.
Output: Clusters of X.
1. $d_0 \leftarrow max\{f(x_i, \hat{x_i}) : x_i \in X\}$.
2. $C_1 \leftarrow X$ (initial value).
3. **While** There is a subset C_i such that $MC(C_i) > d_0$ **do**
4. **repeat**
5. Pick the subset C_i if $MC(C_i) > d_0$.
6. Sort every row of $S(C_i)$ to obtain $S_1(C_i)$.
7. $S_2(C_i)(:, k) \leftarrow S_1(C_i)(:, k+1) - S_1(C_i)(:, k)$, $k = 1, 2, \cdots, n-1$.
8. $MC \leftarrow max\{S_2(C_i)(k, j) : k = 1, 2, \cdots, n, j = 1, 2, \cdots, n-1\}$.
9. Find out x_i, x_j and x_k which generate the MC ($|f(x_i, x_j) - f(x_i, x_k)| = MC$).
10. If $f(x_i, x_t) \leqslant min\{f(x_i, x_j), f(x_i, x_k)\}$ then $x_t \in C_{count}$; otherwise, $x_t \in C_{count+1}$.
11. **until** $max\{MC(C_i)\} \leqslant d_0$.
12. **end while**

KNN-density indicator, as shown in follows:

$$\rho_i = 1/\sum_{x_j \in KNN(x_i)} f(x_i, x_j), \tag{1}$$

where $KNN(x_i)$ is the K-nearest neighbor set of x_i.

Differently from the objects of the sparse area, the objects in a dense area have a spherical neighborhood with a smaller radius which contains the same number of neighbors. The object in the dense areas has a larger local density indicator ρ_i by using equation (1). The sample is considered to belong to the dense subset C if it has a larger ρ_i.

A denoising method is designed as Algorithm 2 after obtaining ρ_i for every object.

Algorithm 2 Denoising.

Input: Distance matrix $S(X)$, t and $N_0 = |KNN(x_i)|$.
Output: The dense subset C.
Initialize: r=0.4.
1. Apply equation (1) to obtain ρ_i for every object.
2. Remove $[0.4 * n]$ objects of X that have smaller ρ_i, retain the other objects in C.
3. $d_0 \leftarrow max\{f(x_i, \hat{x_i}) : x_i, \hat{x_i} \in C\}$.
4. **While** $MC(C) \leqslant t \times d_0$ **do**
5. **repeat**
6. $r \leftarrow r + 0.1$.
7. Remove $[r * n]$ objects of the entire dataset X that have smaller ρ_i, retain the other points in C.
8. Update d_0 and $MC(C)$ of the new subset C.
9. **until** $MC(C) > t \times d_0$ or $r = 0.9$.
10. **end while**.

In general, |C|>[50%n] ($n = |X|$), $r = 0.4$ is considered as an initial value, i.e. [60%n] points are considered as the initial members of the dense subset. A smaller initial value of r may let Algorithm 2 add the times of iteration. $r = 0.9$ (line 9. of Algorithm 2) means that $|C| \geqslant 10\%|X|$. The flowcharts of Algorithm 1 and 2 are as shown in FIGURE 1.

2) FC-KNN ALGORITHM

The main steps of the FC-KNN are as shown in follows.

When the fission of dense subset C is complete after Step 2 of the FC-KNN processes, the remaining objects in the set $X - C$ need to be assigned to their correct category. A simple method is applied to assign the objects of $X - C$: let $A \subset X$ be the subset that contains the already classified points and $U \subset X$ be the subset of unclassified points. If $f(x_i', x_j') = min\{f(x_i, x_j) : x_i \in A, x_j \in U\}$, then x_j' is assigned to the category that contains x_i'.

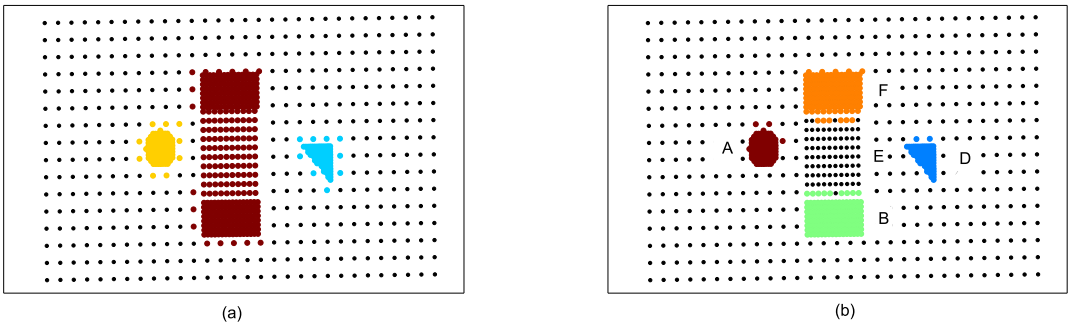

FIGURE 2. Results for tuning courses with parameter t.

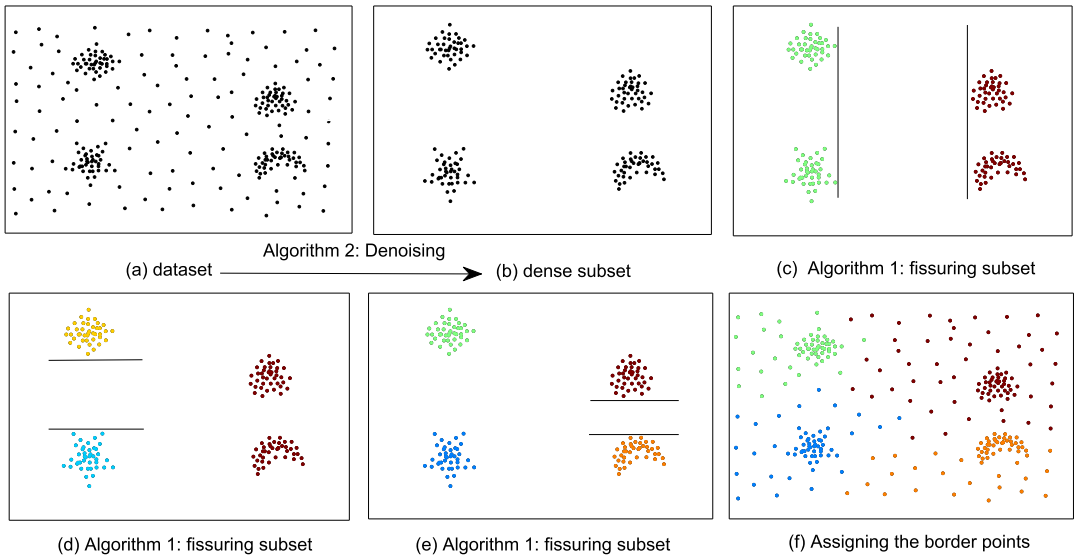

FIGURE 3. A simple example for describing the processes of Algorithm 3.

Algorithm 3 FC-KNN algorithm.

Input: Distance matrix $S(X)$, t and $N_0 = |KNN(x_i)|$.
Output: The clustering result.
Initialize: r=0.4.
1. Use Algorithm 2 to obtain a dense subset C.
2. Cluster the subset C by using Algorithm 1.
3. Assign the objects of $X - C$ to their nearest cluster.

The parameter N_0 is set according to the number of objects in X (such as $N_0 = [1\%|X|]$). We suggest $N_0 < min\{|C_i| : i = 1, 2, \cdots, t\}$, where C_i is the dense family of ith cluster in X and $C = C_1 \cup C_2 \cup \cdots \cup C_t$. Note that $t > 1$ can be considered as a tuning parameter. As shown in FIGURE 2 (b), the densities of different areas can be approximatively ranked as: $density(A) \approx density(D) > density(B) \approx density(F) > density(E)$. Algorithm 2 increases the value of t to remove more border points (sparse area points). For $N_0 = [2\%n]$, when $t \in (1, 3.2]$ the dense subset $C = A \cup D \cup B \cup E \cup F$, the families B and F are connected by some points of E, so the dense families of categories are A, D and $B \cup E \cup F$. When $t \in [3.3, 7.3]$ the dense subset $C = A \cup D \cup B \cup F$, the points of E are considered as the sparse area points and removed, so the dense families of categories are A, D, B and F. When $t \in [7.4, 55]$ the dense subset $C = A \cup D$, the points of $B \cup E \cup F$ are considered as the sparse area points and removed, the dense families of categories are A and D.

FIGURE 3 shows the processes of Algorithm 3. Algorithm 2 is used to obtain a dense subset C, as shown in FIGURE 3 (a) and (b). Algorithm 1 is applied to split the dense subset C into several subsets, as shown in FIGURE 3 (c), (d) and (e). When Algorithm 1 stops splitting subsets, the border point is assigned to its nearest cluster, as shown in FIGURE 3 (f).

IV. EXPERIMENTS

In this section, we evaluate the performance of the proposed method on both simulation data and real data, and then compare it with some state-of-the-art methods that do not need the number of clusters to be input. All the experiments

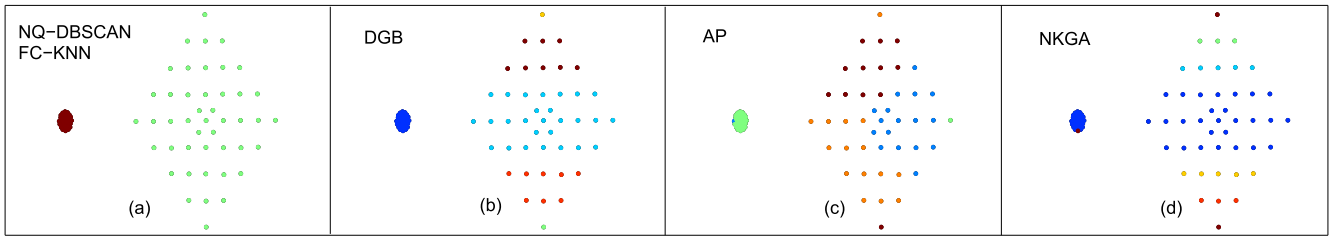

FIGURE 4. The clustering results for different methods used on the Imbalance dataset. ((b), (c) and (d) Were clustered by DGB, AP and NKGA, respectively. FC-KNN and NQ-DBSCAN obtained the same result as in (a)).

TABLE 2. The simple description of datasets.

Dataset	Instances	Features	Clusters	Dataset	Instances	Features	Clusters	Detail
D31	3100	2	31	Iris	150	4	3	three kinds of irises: Setosa, Versicolour and Virginica. Each kind has 50 samples
S1	5000	2	15	Seeds	210	7	3	seeds from Kama, Rosa and Canadian, 70 seeds from each place
A1	3000	2	20	Soybean	47	35	4	47 soybean samples with different diseases, sample distribution: 10 D1, 10 D2, 10 D3 and 17 D4
R15	600	2	15	Vertebral	310	6	2	310 orthopaedic samples, 210 abnormal samples and 100 normal samples
Dimond	2999	2	9	Wifi	2000	7	4	2000 times of signal records in 4 rooms, 500 records in each room
Dim2	1350	2	9	WebKB	1051	4840	2	2 kinds of Web pages, 230 pages and 821 pages, respectively
Imbalance	101	2	2	Adenoma	6	12488	2	6 genes: 3 ADE samples and 3 N1 samples
Aggregation	788	2	7	Leukemia	38	999	3	11 AML samples, 8 T-lineage ALL samples and 19 B-lineage ALL samples
SynthesisO	10000	2	4	AML	15	22283	3	9 AML samples, 3 poly samples and 3 mono samples
SynthesisT	20000	2	3	HL60	12	22283	2	6 HL60-DMSO samples and 6 HL60-Iressa samples

TABLE 3. Number of clusters estimated by various methods.

Dataset	The estimated number of clusters						Dataset	The estimated number of clusters					
	AP	ADPC	NKGA	DGB	NQ-DBSCAN	FC-KNN		AP	ADPC	NKGA	DGB	NQ-DBSCAN	FC-KNN
D31	8	**31**	19	16	**31**	**31**	Iris	2	2	11	4	**3**	**3**
S1	**15**	**15**	14	**15**	**15**	**15**	Seeds	2	**3**	2	7	4	**3**
A1	4	**20**	16	**20**	**20**	**20**	Soybean	2	**4**	**4**	2	**4**	**4**
R15	5	**15**	**15**	**15**	**15**	**15**	Vertebral	1	1	**2**	**2**	**2**	**2**
Dimond	15	**9**	5	**9**	**9**	**9**	Wifi	5	**4**	1	3	**4**	**4**
Dim2	7	**9**	**9**	**9**	**9**	**9**	WebKB	6	1	1	3	3	3
Imbalance	4	1	6	6	**2**	**2**	Adenoma	1	**2**	**2**	3	**2**	**2**
Aggregation	5	**7**	6	**7**	**7**	**7**	Leukemia	**3**	**3**	2	**3**	**3**	**3**
SynthesisO	31	20	12	2	**4**	**4**	AML	1	2	2	**3**	4	**3**
SynthesisT	37	8	9	**3**	**3**	**3**	HL60	**2**	**2**	3	3	3	**2**

are implemented based on the same software and hardware: MATLAB R2014a in the Win7 operating system with Intel Core I5-3230 M 2.6 GHz and 32 G Memory.

The Euclidean function was applied to obtain the distance matrix in all experiments. We selected the following methods for our comparisons with the proposed method: the affinity propagation algorithm (AP) [44], automatic find of density peaks (ADPC) [34], Neighbor Query DBSCAB (NQ-DBSCAN) [11], NK hybrid genetic algorithm (NKGA) [45] and a density and grid based (DGB) clustering method [16].

A. DESCRIPTIONS OF EXPERIMENT DATA

1) SIMULATION DATA

First, some frequently-used datasets obtained from different references are applied to test the algorithms, such as R15 [46], D31 [46], Aggregation [47], A1 [48], S1 [49], Dim2 [50] and Dimond [51] etc. And then three datasets, Imbalance (FIGURE 4), SynthesisO (FIGURE 5) and SynthesisT (FIGURE 5), are constructed for the supplementary tests. All the simulation data are points of two-dimensional Euclidean space.

2) REAL DATA

Several real-world datasets are applied to test the performance of the proposed method, including three plant datasets: Iris[1] [52], [53], Seeds[1] [54] and Soybean[1] [55]; a wireless signal dataset: Wifi[1] [56]; a human vertebral column dataset: Vertebral[1] [57]; a web page dataset: WebKB[2] [58]; and four high-dimensional gene datasets: Adenoma[3] [59],

[1] http://archive.ics.uci.edu/ml/datasets.php
[2] http://www.cs.umd.edu/sen/lbc-proj/LBC.html
[3] http://portals.broadinstitute.org/cgi-bin/cancer/datasets.cgi

TABLE 4. The results' comparison for different methods.

Dataset	Measures	AP	ADPC	NKGA	DGB	NQ-DB SCAN	FC-KNN	Dataset	Measures	AP	ADPC	NKGA	DGB	NQ-DB SCAN	FC-KNN
D31	Accuracy	0.2210	**0.9677**	0.3539	0.4010	0.5416	**0.9677**	Iris	Accuracy	0.5333	0.6667	0.4533	0.3000	0.7867	**0.9067**
	F-Score	0.3466	**0.9679**	0.4537	0.5394	0.6937	**0.9679**		F-Score	0.4329	0.5714	0.5883	0.4615	0.8697	**0.9168**
	ARI	0.1704	**0.9352**	0.3290	0.2267	0.1240	**0.9352**		ARI	0.4120	0.5681	0.2681	0.0990	0.6789	**0.7592**
	NMI	0.4929	**0.9573**	0.6498	0.4242	0.2994	**0.9573**		NMI	0.4509	0.7337	0.0138	0.2170	0.7603	**0.8057**
S1	Accuracy	0.7642	0.9262	0.6992	0.9250	0.9614	**0.9932**	Seeds	Accuracy	0.6048	0.8157	0.3286	0.6524	0.6000	**0.8857**
	F-Score	0.7907	0.9332	0.7315	0.9333	0.9647	**0.9934**		F-Score	0.5102	0.8307	0.1649	0.7549	0.7396	**0.8900**
	ARI	0.6518	0.8915	0.5685	0.8851	0.9378	**0.9858**		ARI	0.4413	**0.8012**	0.0000	0.4386	0.3980	0.7027
	NMI	0.8382	0.9450	0.7878	0.9470	0.9695	**0.9895**		NMI	0.4890	**0.7896**	0.0093	0.3914	0.3636	0.6982
A1	Accuracy	0.1550	0.9433	0.4597	0.9023	0.9450	**0.9717**	Soybean	Accuracy	0.5532	**0.8723**	0.6596	0.5745	0.3617	**0.8723**
	F-Score	0.2615	0.9328	0.5691	0.9077	0.9462	**0.9721**		F-Score	0.3505	0.8872	0.7132	0.3920	0.4592	**0.8904**
	ARI	0.1159	0.9205	0.1954	0.8358	0.8937	**0.9435**		ARI	0.2927	0.6859	0.4977	0.4434	0.1181	**0.7169**
	NMI	0.4174	0.9311	0.6193	0.9135	0.9352	**0.9621**		NMI	0.3603	0.7831	0.6348	0.5901	0.3882	**0.8352**
R15	Accuracy	0.2217	0.9917	0.8983	0.7267	0.8200	**0.9933**	Vertebral	Accuracy	0.6774	0.6774	0.6645	0.3806	0.1516	**0.7710**
	F-Score	0.3416	0.9918	0.9035	0.8363	0.9011	**0.9935**		F-Score	0.4038	0.4038	0.3992	0.3106	0.1437	**0.7976**
	ARI	0.2574	0.9817	0.7968	0.4306	0.7667	**0.9857**		ARI	0.0335	0.0304	0.0166	0.0072	0.2381	**0.2916**
	NMI	0.5460	0.9864	0.8705	0.7435	0.3609	**0.9893**		NMI	0.0000	0.0000	0.0145	0.0448	0.1635	**0.3129**
Dimond	Accuracy	0.3211	**1.0000**	0.5585	0.8303	0.9967	**1.0000**	Wifi	Accuracy	0.1405	0.8625	0.2500	0.5025	0.7545	**0.9355**
	F-Score	0.3799	**1.0000**	0.6632	0.9055	0.9969	**1.0000**		F-Score	0.1671	0.8859	0.1000	0.5416	0.8561	**0.9402**
	ARI	0.2182	**1.0000**	0.5775	0.7158	0.9929	**1.0000**		ARI	0.1948	0.8103	0.0000	0.3497	0.6868	**0.8470**
	NMI	0.4066	**1.0000**	0.8153	0.8036	0.9922	**1.0000**		NMI	0.2646	0.8309	0.0078	0.4260	0.6531	**0.8635**
Dim2	Accuracy	0.8259	**1.0000**	0.9289	**1.0000**	**1.0000**	**1.0000**	WebKB	Accuracy	0.1922	0.7812	0.7812	0.7174	0.7878	**0.8773**
	F-Score	0.8482	**1.0000**	0.9384	**1.0000**	**1.0000**	**1.0000**		F-Score	0.3007	0.4386	0.4386	0.7141	0.7057	**0.8173**
	ARI	0.7549	**1.0000**	0.8714	**1.0000**	**1.0000**	**1.0000**		ARI	0.0011	0.0146	0.0146	0.3094	0.2714	**0.4960**
	NMI	0.8782	**1.0000**	0.9337	**1.0000**	**1.0000**	**1.0000**		NMI	0.0013	0.0000	0.0000	0.1374	0.1449	**0.3647**
Imbalance	Accuracy	0.6931	0.5743	0.5941	0.8218	**1.0000**	**1.0000**	Adenoma	Accuracy	0.5000	**1.0000**	0.6667	0.8333	0.6667	**1.0000**
	F-Score	0.7806	0.6265	0.6608	0.8889	**1.0000**	**1.0000**		F-Score	0.3333	**1.0000**	0.7273	0.9091	0.8000	**1.0000**
	ARI	0.6778	0.0105	0.1464	0.7692	**1.0000**	**1.0000**		ARI	0.0000	**1.0000**	0.0000	0.7059	0.2424	**1.0000**
	NMI	0.4812	0.0415	0.1135	0.2768	**1.0000**	**1.0000**		NMI	0.0000	**1.0000**	0.2314	0.9286	0.4787	**1.0000**
Aggregation	Accuracy	0.7183	0.9987	0.7919	**1.0000**	**1.0000**	**1.0000**	Leukemia	Accuracy	0.4474	0.9737	0.6579	**1.0000**	**1.0000**	**1.0000**
	F-Score	0.8048	0.9980	0.8749	**1.0000**	**1.0000**	**1.0000**		F-Score	0.4196	0.9726	0.5383	**1.0000**	**1.0000**	**1.0000**
	ARI	0.7497	0.9978	0.9231	**1.0000**	**1.0000**	**1.0000**		ARI	0.2332	0.9192	0.3679	**1.0000**	**1.0000**	**1.0000**
	NMI	0.8672	0.9959	0.9479	**1.0000**	**1.0000**	**1.0000**		NMI	0.3081	0.9110	0.5070	**1.0000**	**1.0000**	**1.0000**
SynthesisO	Accuracy	0.2532	0.4871	0.6733	0.7567	0.9533	**1.0000**	AML	Accuracy	0.6000	0.8000	0.5333	0.8667	0.7333	**1.0000**
	F-Score	0.2756	0.4913	0.7051	0.7132	0.9311	**1.0000**		F-Score	0.2500	0.6222	0.3810	0.8510	0.8083	**1.0000**
	ARI	0.1867	0.4113	0.6314	0.7087	0.9636	**1.0000**		ARI	0.0075	0.4788	0.0241	0.6269	0.3697	**1.0000**
	NMI	0.2213	0.4218	0.6533	0.7162	0.9212	**1.0000**		NMI	0.0000	0.6899	0.1605	0.7173	0.5361	**1.0000**
SynthesisT	Accuracy	0.1883	0.6255	0.5819	**1.0000**	**1.0000**	**1.0000**	HL60	Accuracy	0.8333	0.7500	0.6667	0.7500	0.8333	**1.0000**
	F-Score	0.2539	0.6517	0.6103	**1.0000**	**1.0000**	**1.0000**		F-Score	0.8333	0.7895	0.7568	0.8571	0.9091	**1.0000**
	ARI	0.0113	0.5631	0.5528	**1.0000**	**1.0000**	**1.0000**		ARI	0.3889	0.1951	0.4642	0.7179	0.7500	**1.0000**
	NMI	0.0136	0.5882	0.5736	**1.0000**	**1.0000**	**1.0000**		NMI	0.3500	0.3437	0.0973	0.3437	0.4787	**1.0000**

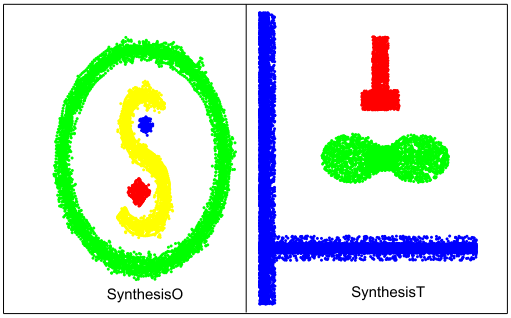

FIGURE 5. SynthesisO and SynthesisT.

Leukemia[3] [60], AML[3] [61] and HL60[3] [61]. Simple descriptions of these real datasets are provided in TABLE 2.

B. RESULTS AND COMPARISONS

1) PRESENTATION OF RESULTS

A simple description of datasets is presented in TABLE 2. TABLE 3 presents the number of clusters estimated by various methods. TABLE 4 shows the clustering results when compared with other various methods.

To evaluate and compare the performance of the clustering methods, we apply the evaluation metrics: Accuracy, F-Score, Adjusted Rand Index (ARI) [62] and Normalized Mutual Information (NMI) [63] in our experiments to do a comprehensive evaluation. The higher the value, the better the clustering performance for all these measures. Compared with the best results of other algorithms, our method has relative advantages of 0.1333, 0.149, 0.3731 and 0.2827 (TABLE 4) with respect to Accuracy, F-Score, ARI and NMI for the AML dataset, respectively.

We conduct the Friedman test with the post-hoc Nemenyi test (α= 0.10) [15] to examine whether the difference between any two clustering algorithms is significant in terms of their average ranks. The difference between two algorithms is significant if the gap between their ranks is larger than CD. There is a line between two algorithms if the rank gap between them is smaller than CD. This test shows that FC-KNN is significantly better than other methods. ADPC and NQ-DBSCAN are significantly better than NKGA and AP.

In summary, our method achieves better results with respect to the estimation of cluster number, Accuracy,

TABLE 5. The parameter settings of FC-KNN, NQ-DBSCAN, DGB and ADPC for experimental datasets.

Dataset	FC-KNN	NQ-DBSCAN	DGB	ADPC	Dataset	FC-KNN	NQ-DBSCAN	DGB	ADPC
D31	N_0 =ceil(2%n) t=4	Eps=0.6 MinPts=23	GN=20^2, CF=0.43, NT=0.2	d_c=0.01	Iris	N_0=ceil(3%n) t=5	Eps=0.42 MinPts=5	GN=22^2, CF=0.4, NT=0	d_c=0.02
S1	N_0=ceil(2%n) t=4	Eps=5000 MinPts=19	GN=30^2, CF=0.2, NT=1	d_c=0.02	Seeds	N_0=ceil(3%n) t=5	Eps=0.8 MinPts=8	GN=30^2, CF=0.3, NT=0	d_c=0.02
A1	N_0=ceil(2%n) t=4	Eps=1350 MinPts=36	GN=35^2, CF=0.5, NT=1	d_c=0.02	Soybean	N_0=ceil(2%n) t=2	Eps=1.8 MinPts=3	GN=3^2, CF=0.05, NT=0	d_c=0.03
R15	N_0=ceil(3%n) t=4	Eps=0.3 MinPts=6	GN=20^2, CF=0.5, NT=1	d_c=0.03	Vertebral	N_0=ceil(3%n) t=2	Eps=16 MinPts=7	GN=20^2, CF=0.3, NT=0	d_c=0.02
Dimond	N_0=ceil(3%n) t=4	Eps=0.3 MinPts=35	GN=20^2, CF=0.5, NT=0.2	d_c=0.02	Wifi	N_0=ceil(3%n) t=2	Eps=6 MinPts=20	GN=30^2, CF=0.3, NT=0	d_c=0.02
Dim2	N_0=ceil(2%n) t=4	Eps=5000 MinPts=10	GN=25^2, CF=0.3, NT=0	d_c=0.02	WebKB	N_0=ceil(2%n) t=2	Eps=0.11 MinPts=8	GN=50^2, CF=0.1, NT=1	d_c=0.02
Imbalance	N_0=ceil(3%n) t=4	Eps=1 MinPts=5	GN=20^2, CF=0.01, NT=0	d_c=0.02	Adenoma	N_0=ceil(3%n) t=2	Eps=10000 MinPts=2	GN=5^2, CF=0.01, NT=0	d_c=0.5
Aggregation	N_0=ceil(3%n) t=4	Eps=1.5 MinPts=10	GN=25^2, CF=0.3, NT=0	d_c=0.02	Leukemia	N_0=ceil(2%n) t=2	Eps=0.2 MinPts=3	GN=5^2, CF=0.3, NT=0	d_c=0.05
SynthesisO	N_0=ceil(1%n) t=4	Eps=0.2 MinPts=50	GN=20^2, CF=0.2, NT=1	d_c=0.01	AML	N_0=ceil(2%n) t=4	Eps=8000 MinPts=2	GN=5^2, CF=0.2, NT=0	d_c=0.3
SynthesisT	N_0=ceil(1%n) t=4	Eps=0.38 MinPts=50	GN=35^2, CF=0.08, NT=0	d_c=0.005	HL60	N_0=ceil(2%n) t=2	Eps=5000 MinPts=2	GN=6^2, CF=0.1, NT=1	d_c=0.3

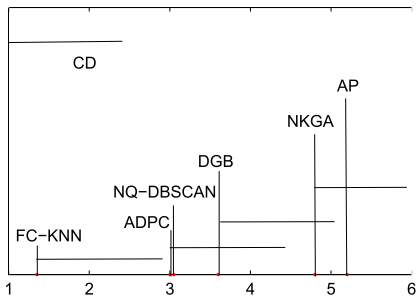

FIGURE 6. Critical difference (CD) diagram of the post-hoc Nemenyi test.

F-Score, ARI and NMI, compared comprehensively with other methods.

2) PARAMETER ANALYSIS

As shown in TABLE 5, the parameter settings of NQ-DBSCAN and DGB are random and ruleless for the datasets. It is thus difficult to guess the right parameters for NQ-DBSCAN and DGB if the results are unknown before clustering occurs. Because the parameters of the FC-KNN have their own regulation, N_0 can be set by the indicator of the objects' number, such as $N_0 = ceil(2\%n)$, where $ceil(\cdot)$ is a rounding function. The selection of parameter t has a direction: the larger the value of t, the fewer the points are retained in the dense families. The parameters of the FC-KNN are therefore easy to set.

The proposed method is robust. It can obtain the same clustering results even when we choose values for parameters t and N_0 in wide intervals $[t^-, t^+]$ and $[N_0^-, N_0^+]$, respectively. For the Iris dataset, the FC-KNN can obtain the same results with $N_0 \in [ceil(3\%n), ceil(5\%n)]$ and $t \in [1.1, 6]$. However, NQ-DBSCAN cannot obtain the same results for three slightly different cases of Eps=0.41, Eps=0.42 and Eps=0.43, when MinPts=5. In fact, three of the most influential parameters for DGB are cutoff factor (CF), grid number (GN) and noise threshold (NT). The DGB cannot get the same results for three cases of CF=0.19, CF=0.20 and CF=0.21, when GN=25^2 and NT=0. Clearly, the NQ-DBSCAN and the DGB are not robust in their parameters.

In the description of the above algorithms, the distance matrix is a significant input. This matrix depends on the correct selection of the attributes, the correct value of the selected attributes and a good distance (recognition) function. We say the recognition function f_1 is better (stronger) than f_2 if $|f_1(x_i, x_i') - f_1(x_i, x_j)| \geqslant |f_2(x_i, x_i') - f_2(x_i, x_j)|$ for all $x_i, x_i' \in C_i$ and $x_j \in C_j$, where C_i and C_j are two clusters of X.

For the simulation data that are Euclidean space points, the Euclidean function is a strong recognition function for them; then, the parameter t can be set to a large value. However, if the Euclidean function is a weak recognition function for some real datasets, such as the dataset Vertebral, the AP algorithm classifies the data Vertebral as one cluster. Differently, our FC-KNN algorithm can determine the correct clusters after tuning the parameter t with a smaller value when the recognition function is weak.

3) RUNTIME

Equation (1) takes $\mathcal{O}(n*N_0)$ operations. Algorithm 1 splits the set X (or dense subset C) into subsets $C_1, C_2, \cdots, C_k$. Since $|C_i| \ll |X|$, data processing will become faster and faster, accompanied by the dividing courses of subsets. k clusters are obtained after $k-1$ times of dividing subsets, then, its time complexity is $\mathcal{O}(1)$. Moreover, if $|C| = m$, then $|X - C| = n - m$, and the time complexity of assigning border points is

TABLE 6. The runtime (second) of various methods.

Dataset	AP	ADPC	NKGA	DGB	NQ-DBSCAN	FC-KNN
SynthesisT	1192.818	803.211	6158.251	36.127	509.853	285.787
D31	76.940	26.712	613.943	4.922	8.144	5.589
AML	1.553	1.826	23.082	3.762	0.826	0.577
HL60	1.048	1.138	19.513	1.908	0.752	0.424

hence $\mathcal{O}(n-m)$. The time complexity of the whole FC-KNN algorithm is $\mathcal{O}(n^2)$ in the worst case.

The runtime of various methods for four datasets are presented in TABLE 6. Here the SynthesisT dataset has the most objects, the D31 dataset has the most clusters, and the HL60 and AML datasets have the most dimensionality. The runtime of our method is not influenced by the data dimensionality.

4) COMPARISONS AND DISCUSSIONS

In FIGURE 4, we plotted the densities of the two clusters in the Imbalance dataset, which have a significant difference. It is difficult to determine the number of categories with the AP, ADPC, NKGA and DGB algorithms. The ADPC can not find the second center point of the Imbalance dataset, that is this algorithm considers the dataset as one cluster. In FIGURE 5, clearly no single point can be considered as the geometrical centroid of the annulus in the SynthesisO dataset, and no single point can be considered as the geometrical centroid of the T shape cluster in the SynthesisT dataset. The ADPC, as a state-of-the-art centroid-based clustering method, cannot correctly estimate the number of clusters for them. Our proposed method aims at mining the dense family of every category, not the center points. It can correctly determine the number of clusters for the Imbalance, SynthesisO and SynthesisT datasets.

The NQ-DBSCAN and the DGB produce good results for two-dimensional data after they tune the parameters many times with reference to two-dimensional figures. However, they do not work well for high-dimensional data, because these data cannot show well in two-dimensional figures. The DGB and the NQ-DBSCAN are much superior to their prototypes with respect to accuracy and runtime. However, they are also hampered by the ruleless parameters when they deal with multidimensional data.

AP [44] is an unsupervised algorithm without any parameters. The parameters of NKGA [45] are recommended by the publication [45]. The algorithms of parameter-free or fixed parameter value may not be adaptive to various kinds of datasets. The parameter of ADPC [34] is shown in TABLE 5, $d_c = 0.02$ means that the parameter of ADPC takes the value at the position of first 2% of all distances [34].

Unlike the methods that need the number of clusters to be input, such as K-means, our method need no prior conditions. In contrast to the centroid-based methods, such as ADPC, our method focuses on seeking dense families for every category, not just the center point, so it can deal with more kinds of datasets. To compare it with the grid-based method DGB, our method is not influenced by the size or number of grid cells and the dimensionality of data. Moreover, unlike the NQ-DBSCAN, our method it is easier to set parameters. Our FC-KNN obtains satisfying results more easily than the DGB and the NQ-DBSCAN do, when faced with a new high-dimensional dataset that has no references to known clustering results.

V. CONCLUSION

In this article, the FC algorithm is proposed, and then it is combined with the K-nearest neighbor local density indicator to propose the FC-KNN algorithm. The fundamental task that is challenging for clustering is how to determine the number of clusters for a dataset. Our proposed method aims at dealing with this task. Interestingly, our method does not need to assume that the number of categories is known before clustering occurs. Both the simulation and real datasets are applied to test the performance and effectiveness of the proposed method. Our proposed algorithm is also compared with several frequently-used clustering algorithms, including the centroid-based algorithm ADPC, the intelligent algorithm NKGA, the grid-based algorithm DGB, the density-based algorithm NQ-DBSCAN and the parameter-free algorithm AP. The experiments indicate that our method achieves better results, in terms of the evaluation metrics (TABLE 4) and the estimated number of clusters (TABLE 3), than the other methods under comparison. Based on this work, it will be interesting to extend our FC-KNN into a fully adaptive method in the future.

APPENDIX A: THE PROOF OF THEOREM

Reduction to absurdity is applied to prove the theorem. If there is a crack $|f(x_0,x_i) - f(x_0,x_j)| > d_0(C)$, then there is an x_s such that $f(x_0,x_s) \in (f(x_0,x_j), f(x_0,x_i))$ holds. On the other hand, $|f(x_0,x_i) - f(x_0,x_j)|$ is a crack, $f(x_0,x_s) \notin (f(x_0,x_j), f(x_0,x_i))$ for all $x_s \in X$. Detailed descriptions are as follows.

Proof: If there is a crack $|f(x_0,x_i) - f(x_0,x_j)| > d_0(C)$, then $f(x_i,x_j) \geqslant f(x_0,x_i) - f(x_0,x_j) > d_0(C)$ (suppose $f(x_0,x_i) > f(x_0,x_j)$). Thus, if x_i and x_j are not adjacent points on the $d_0(C)$-road, there must be a point x_k on the road from x_i to x_j ($x_i \sim x_k \sim x_j$).

If $f(x_0,x_k) \notin (f(x_0,x_j), f(x_0,x_i))$, then one of $|f(x_0,x_i) - f(x_0,x_k)| > |f(x_0,x_i) - f(x_0,x_j)| > d_0(C)$ and $|f(x_0,x_j) - f(x_0,x_k)| > |f(x_0,x_i) - f(x_0,x_j)| > d_0(C)$ holds. Assuming that $|f(x_0,x_i) - f(x_0,x_k)| > d_0(C)$, then there is x_t on the road x_i to x_j ($x_i \sim x_t \sim x_k \sim x_j$). Because the road of x_i to x_j is a part of the $d_0(C)$-road, x_i and x_j can be connected by some points, and the distance between two connection points is less than or equal to $d_0(C)$. If $f(x_0,x_t) \notin (f(x_0,x_j), f(x_0,x_i))$, then there must be a point x_s on the road from x_i to x_j such that $f(x_0,x_s) \in (f(x_0,x_j), f(x_0,x_i))$ holds in the finite set C.

However, $|f(x_0,x_i) - f(x_0,x_j)|$ is a crack, $f(x_0,x) \notin (f(x_0,x_j), f(x_0,x_i))$ for all $x \in C$. It is a contradiction. Hence, all the cracks must be less than or equal to $d_0(C)$.

REFERENCES

[1] E. Schikuta, "Grid-clustering: An efficient hierarchical clustering method for very large data sets," in *Proc. 13th Int. Conf. Pattern Recognit.*, Aug. 1996, pp. 101–105.

[2] E. W. M. Ma and T. W. S. Chow, "A new shifting grid clustering algorithm," *Pattern Recognit.*, vol. 37, no. 3, pp. 503–514, Mar. 2004.

[3] T. Parsons, "Persistent earthquake clusters and gaps from slip on irregular faults," *Nature Geosci.*, vol. 1, no. 1, pp. 59–63, Jan. 2008.

[4] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proc. Nat. Acad. Sci. USA*, vol. 95, no. 25, pp. 14863–14868, Dec. 1998.

[5] W. Huang, X. Cao, F. H. Biase, P. Yu, and S. Zhong, "Time-variant clustering model for understanding cell fate decisions," *Proc. Nat. Acad. Sci. USA*, vol. 111, no. 44, pp. E4797–E4806, Nov. 2014.

[6] J. D. Hamilton, "A new approach to the economic analysis of nonstationary time series and the business cycle," *Econometrica*, vol. 57, no. 2, pp. 357–384, Mar. 1989.

[7] G. Leibon, S. Pauls, D. Rockmore, and R. Savell, "Topological structures in the equities market network," *Proc. Nat. Acad. Sci. USA*, vol. 105, no. 52, pp. 20589–20594, Dec. 2008.

[8] S. Galbraith, J. A. Daniel, and B. Vissel, "A study of clustered data and approaches to its analysis," *J. Neurosci.*, vol. 30, no. 32, pp. 10601–10608, Aug. 2010.

[9] D. Allen and G. Goldstein, *Cluster Analysis in Neuropsychological Research Recent Applications*. New York, NY, USA: Springer, 2013.

[10] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," *Data Min. Knowl. Disc.*, vol. 96, no. 34, pp. 226–231, Aug. 1996.

[11] Y. Chen, S. Tang, N. Bouguila, C. Wang, J. Du, and H. Li, "A fast clustering algorithm based on pruning unnecessary distance computations in DBSCAN for high-dimensional data," *Pattern Recognit.*, vol. 83, pp. 375–387, Nov. 2018.

[12] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: Ordering points to identify the clustering structure," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, Philadelphia, PA, USA, May/Jun. 1999, pp. 49–60.

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

[14] R. Mehmood, G. Zhang, R. Bie, H. Dawood, and H. Ahmad, "Clustering by fast search and find of density peaks via heat diffusion," *Neurocomputing*, vol. 208, pp. 210–217, Oct. 2016.

[15] Y. Zhu, K. M. Ting, and M. J. Carman, "Grouping points by shared subspaces for effective subspace clustering," *Pattern Recognit.*, vol. 83, pp. 230–244, Nov. 2018.

[16] B. Wu and B. M. Wilamowski, "A fast density and grid based clustering method for data with arbitrary shapes and noise," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 1620–1628, Aug. 2017.

[17] W. Wang, J. Yang, and R. R. Muntz, "STING: A statistical information grid approach to spatial data mining," in *Proc. 23rd Int. Conf. Very Large Data Bases*, Athens, Greece, Feb. 1997, pp. 186–195.

[18] R. Agrawal, J. E. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data for data mining applications," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, Seattle, WA, USA, Jun. 1998, pp. 94–105.

[19] G. Sheikholeslami, S. Chatterjee, and A. Zhang, "WaveCluster: A wavelet-based clustering approach for spatial data in very large databases," *VLDB J. Int. J. Very Large Data Bases*, vol. 8, nos. 3–4, pp. 289–304, Feb. 2000.

[20] C. Fraley and A. E. Raftery, "Model-based clustering, discriminant analysis, and density estimation," *J. Amer. Stat. Assoc.*, vol. 97, no. 458, pp. 611–631, Jun. 2002.

[21] D. H. Fisher, "Improving inference through conceptual clustering," in *Proc. 6th Nat. Conf. Artif. Intell. (AAAI)*, Seattle, WA, USA, Jul. 1987, pp. 461–465.

[22] T. Chen, N. L. Zhang, T. Liu, K. M. Poon, and Y. Wang, "Model-based multidimensional clustering of categorical data," *Artif. Intell.*, vol. 176, no. 1, pp. 2246–2269, Jan. 2012.

[23] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, vol. 1, L. M. Le Cam and J. Neyman, Eds. Berkeley, CA, USA: Univ. California Press, Jan. 1967, pp. 281–297.

[24] R. T. Ng and J. Han, "CLARANS: A method for clustering objects for spatial data mining," *IEEE Trans. Knowl. Data Eng.*, vol. 14, no. 5, pp. 1003–1016, Sep. 2002.

[25] K. Lahari, M. R. Murty, and S. C. Satapathy, "Partition based clustering using genetic algorithm and teaching learning based optimization: Performance analysis," *Adv. Intell. Syst. Comput.*, vol. 338, pp. 191–200, Mar. 2015.

[26] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1988.

[27] P. Pipenbacher, A. Schliep, S. Schneckener, A. Schonhuth, D. Schomburg, and R. Schrader, "ProClust: Improved clustering of protein sequences with an extended graph-based approach," *Bioinformatics*, vol. 18, no. 2, pp. S182–S191, Oct. 2002.

[28] V.-V. Vu and H.-Q. Do, "Graph-based clustering with background knowledge," in *Proc. 8th Int. Symp. Inf. Commun. Technol. (SoICT)*, New York, NY, USA, Dec. 2017, pp. 167–172.

[29] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. Hoboken, NJ, USA: Wiley, Mar. 1990.

[30] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, Montreal, QC, Canada, Jun. 1996, vol. 25, no. 2, pp. 103–114.

[31] G. Karypis, E. H. Han, and V. Kumar, "CHAMELEON: A hierarchical clustering algorithm using dynamic modeling," *IEEE Comput.*, vol. 32, no. 8, pp. 68–75, Aug. 1999.

[32] R. Scitovski and K. Sabo, "Analysis of the k-means algorithm in the case of data points occurring on the border of two or more clusters," *Knowl.-Based Syst.*, vol. 57, pp. 1–7, Feb. 2014.

[33] G. Tzortzis and A. Likas, "The MinMax k-means clustering algorithm," *Pattern Recognit.*, vol. 47, no. 7, pp. 2505–2516, Jul. 2014.

[34] T. Liu, H. Li, and X. Zhao, "Clustering by search in descending order and automatic find of density peaks," *IEEE Access*, vol. 7, pp. 133772–133780, 2019.

[35] J. Jiang, D. Hao, Y. Chen, M. Parmar, and K. Li, "GDPC: Gravitation-based density peaks clustering algorithm," *Phys. A, Stat. Mech. Appl.*, vol. 502, pp. 345–355, Jul. 2018.

[36] Y. Chen, X. Hu, W. Fan, L. Shen, Z. Zhang, X. Liu, J. Du, H. Li, Y. Chen, and H. Li, "Fast density peak clustering for large scale data based on kNN," *Knowl.-Based Syst.*, vol. 187, Jan. 2020, Art. no. 104824, doi: 10.1016/j.knosys.2019.06.032.

[37] M. Parmar, D. Wang, X. Zhang, A.-H. Tan, C. Miao, J. Jiang, and Y. Zhou, "REDPC: A residual error-based density peak clustering algorithm," *Neurocomputing*, vol. 348, pp. 82–96, Jul. 2019.

[38] M. D. Parmar, W. Pang, D. Hao, J. Jiang, W. Liupu, L. Wang, and Y. Zhou, "FREDPC: A feasible residual error-based density peak clustering algorithm with the fragment merging strategy," *IEEE Access*, vol. 7, pp. 89789–89804, 2019.

[39] M. Du, S. Ding, and H. Jia, "Study on density peaks clustering based on k-nearest neighbors and principal component analysis," *Knowl.-Based Syst.*, vol. 99, pp. 135–145, May 2016.

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

[41] Y. Zhu, K. M. Ting, and M. J. Carman, "Density-ratio based clustering for discovering clusters with varying densities," *Pattern Recognit.*, vol. 60, pp. 983–997, Dec. 2016.

[42] J.-H. Kim, J.-H. Choi, K.-H. Yoo, and A. Nasridinov, "AA-DBSCAN: An approximate adaptive DBSCAN for finding clusters with varying densities," *J. Supercomput.*, vol. 75, no. 1, pp. 142–169, Jan. 2019.

[43] A. Bryant and K. Cios, "RNN-DBSCAN: A density-based clustering algorithm using reverse nearest neighbor density estimates," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 6, pp. 1109–1121, Jun. 2018.

[44] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, Feb. 2007.

[45] R. Tinos, L. Zhao, F. Chicano, and D. Whitley, "NK hybrid genetic algorithm for clustering," *IEEE Trans. Evol. Comput.*, vol. 22, no. 5, pp. 748–761, Oct. 2018.

[46] C. J. Veenman, M. J. T. Reinders, and E. Backer, "A maximum variance cluster algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1273–1280, Sep. 2002.

[47] A. Gionis, H. Mannila, and P. Tsaparas, "Clustering aggregation," *ACM Trans. Knowl. Discovery Data*, vol. 1, no. 1, pp. 1–30, Mar. 2007.

[48] K. Ismo and P. Franti, "Dynamic local search for clustering with unknown number of clusters," in *Proc. Int. Conf. Pattern Recogn.*, Aug. 2002, vol. 2, no. 16, pp. 240–243.

[49] P. Fränti and O. Virmajoki, "Iterative shrinking method for clustering problems," *Pattern Recognit.*, vol. 39, no. 5, pp. 761–775, May 2006.

[50] P. Franti, O. Virmajoki, and V. Hautamaki, "Fast agglomerative clustering using a k-Nearest neighbor graph," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1875–1881, Nov. 2006.
[51] S. Salvador and P. Chan, "Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms," in *Proc. 16th IEEE Int. Conf. Tools with Artif. Intell.*, Nov. 2004, pp. 576–584.
[52] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugenics*, vol. 7, no. 2, pp. 179–188, Sep. 1936.
[53] F. Huang, X. Li, S. Zhang, and J. Zhang, "Harmonious genetic clustering," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 199–214, Jan. 2018.
[54] M. Charytanowicz and J. Niewczas, "Complete gradient clustering algorithm for features analysis of X-ray images," in *Information Technologies in Biomedicine*, E. Pietka and J. Kawa, Eds. Berlin, Germany: Springer-Verlag, Jan. 2010, pp. 15–24.
[55] R. S. Michalski and R. L. Chilausky, "Learning by being told and learning from examples: An experimental comparison of the two methodes of knowledge acquisition in the context of developing an expert system for soybean desease diagnoiss," *Int. J. Policy Anal. Inf. Syst.*, vol. 4, no. 2, pp. 125–161, Jan. 1980.
[56] J. G. Rohra, "User localization in an indoor environment using fuzzy hybrid of particle swarm optimization & gravitational search algorithm with neural networks," in *Proc. 6th Int. Conf. Soft Comput. Problem Solving*, Feb. 2017, pp. 286–295.
[57] E. Berthonnaud, J. Dimnet, P. Roussouly, and H. Labelle, "Analysis of the sagittal balance of the spine and pelvis using shape and orientation parameters," *J. Spinal Disorders Techn.*, vol. 18, no. 1, pp. 40–47, Feb. 2005.
[58] P. Martin, "The WebKB set of tools: A common scheme for shared WWW Annotations, shared knowledge bases and information retrieval," in *Proc. Int. Conf. Conceptual Struct.*, Aug. 1997, pp. 585–588.
[59] A. Sweet-Cordero, "An oncogenic *KRAS*2 expression signature identified by cross-species gene-expression analysis," *Nat. Genet.*, vol. 37, no. 1, pp. 48–55, Dec. 2004.
[60] C. Wiwie, J. Baumbach, and R. Röttger, "Comparing the performance of biomedical clustering methods," *Nature Methods*, vol. 12, no. 11, pp. 1033–1038, Nov. 2015.
[61] K. Stegmaier and S. M. Corsello, "Gefitinib (Iressa) induces myeloid differentiation of acute myeloid leukemia," *Blood*, vol. 106, no. 8, pp. 2841–2848, Oct. 2005.
[62] L. Du, Y. Pan, and X. Luo, "Robust spectral clustering via matrix aggregation," *IEEE Access*, vol. 6, pp. 53661–53670, 2018.
[63] S. Abbasi and S. Nejatian, "Clustering ensemble selection considering quality and diversity," *Artif. Intell. Rev.*, vol. 52, pp. 1311–1340, Jan. 2019.

SHIZHAN LU received the M.S. degree in mathematics from Guangxi University for Nationalities, China, in 2014. He is currently pursuing the Ph.D. degree with the Nanjing University of Science and Technology, China. His current research interests include data mining, clustering analysis, and intelligence algorithm.

LONGSHENG CHENG received the M.S. degree from the East China Institute of Technology, China, in 1988, and the Ph.D. degree in system engineering from the Nanjing University of Science and Technology, China, in 1998. From 1998 to 1999, he was with the City University of Hong Kong as a Research Assistant. Since 2005, he has been a Professor of management sciences and applied statistics with the School of Economics and Management, Nanjing University of Science and Technology. His current research interests include prognostic and health monitoring, machine learning, quality engineering, and data mining.

RASHID MEHMOOD received the M.S. degree from COMSATS University, Pakistan, and the Ph.D. degree from Beijing Normal University, China. He is currently affiliated with the Department of Software Engineering, University of Kotli Azad Jammu and Kashmir, Pakistan. His research interests include clustering, single-cell RNA-seq, analysis of next-generation sequencing data, and circadian rhythm for cancer research.