Hindawi
Complexity
Volume 2022, Article ID 6742120, 13 pages
https://doi.org/10.1155/2022/6742120

Research Article

Adaptive Mixed-Attribute Data Clustering Method Based on Density Peaks

Shihua Liu [1,2]

[1]*School of Artificial Intelligence, Wenzhou Polytechnic, Wenzhou 325035, China*
[2]*Wenzhou Network Security Detection and Protection Engineering Technology Research Center, Wenzhou 325035, China*

Correspondence should be addressed to Shihua Liu; chaoshua@foxmail.com

Received 20 September 2021; Revised 31 January 2022; Accepted 11 February 2022; Published 17 March 2022

Academic Editor: Qingling Wang

The clustering of mixed-attribute data is a vital and challenging issue. The density peaks clustering algorithm brings us a simple and efficient solution, but it mainly focuses on numerical attribute data clustering and cannot be adaptive. In this paper, we studied the adaptive improvement method of such an algorithm and proposed an adaptive mixed-attribute data clustering method based on density peaks called AMDPC. In this algorithm, we used the unified distance metric of mixed-attribute data to construct the distance matrix, calculated the local density based on K-nearest neighbors, and proposed the automatic determination method of cluster centers based on three inflection points. Experimental results on real University of California-Irvine (UCI) datasets showed that the proposed AMDPC algorithm could realize adaptive clustering of mixed-attribute data, can automatically obtain the correct number of clusters, and improved the clustering accuracy of all datasets by more than 22.58%, by 24.25%, by 28.03%, by 22.5%, and by 10.12% for the Heart, Cleveland, Credit, Acute, and Adult datasets compared to that of the traditional K-prototype algorithm, respectively. It also outperformed a modified density peaks clustering algorithm for mixed-attribute data (DPC_M) algorithms.

1. Introduction

Clustering analysis has been widely used in statistics, machine learning, pattern recognition, image processing, such as image inpainting [1, 2], image super-resolution reconstruction [3], and so on. Mixed-attribute data clustering is one of the research hotspots in data mining. There are many solutions to mixed-attribute data clustering, including attribute conversion methods, clustering ensemble methods, prototype-based methods, hierarchical clustering methods, and density clustering methods [4]. The K-prototypes algorithm proposed by Huang [5] and the iterative clustering learning based on object-cluster similarity metric (OCIL) algorithm proposed by Cheung and Jia [6] are both typical prototype-based methods. The similarity-based agglomerative clustering (SBAC) algorithm proposed by Li and Biswas [7] is a famous aggregation hierarchical clustering method. Density clustering algorithms include the relative density-based clustering algorithm for mixture datasets (RDBC_M) algorithm based on relative density proposed by Huang and Li [8] and the density-based clustering algorithm for mixed data with mixed distance measure methods (MDCDen) algorithm based on density and mixed-distance measurement proposed by Chen and He [9]. But the state-of-the-art methods require user intervention and parameter tuning, so they cannot realize adaptive clustering.

The density peaks clustering (DPC) algorithm proposed by Rodriguez and Laio [10] has attracted a great deal of attention from researchers in recent years [11–14]. In this algorithm, a decision graph is constructed by calculating a local density ρ_i and a relative distance δ_i, and the number of clusters is determined by manually selecting the center points of the clusters in the decision graph. The remaining data points will be assigned to the cluster of the nearest higher-density neighbor. Theoretically, it can cluster data of both arbitrary shape and type and automatically identify outliers. The algorithm is efficient and has only one parameter, d_c (cutoff distance), which determines the local

density calculation. The input of the DPC algorithm is the distance matrix between data points, and as long as the distance measurement problem of mixed-attribute data is solved, the algorithm can be applied directly to cluster the mixed-attribute data. Therefore, Liu et al. [15] defined a distance measurement method for mixed attributes and improved the DPC algorithm to a modified DPC algorithm for mixed-attribute data (DPC_M) algorithm, which was successfully applied to mixed-attribute data clustering. Du et al. [16] defined a distance-measurement method between mixed-attribute data points by referring to the similarity in OCIL algorithm and used the DPC algorithm to perform clustering analysis on the numerical attribute, categorical attribute, and mixed-attribute data. These two algorithms prove the feasibility of the density peaks algorithm in the clustering of mixed-attribute data. But they are not adaptive and need manual intervention in the clustering process.

Adaptive algorithm is one of the most popular research fields [17–19]. There are also many studies on adaptive improvement of DPC algorithm, but they mainly focus on clustering of numerical attribute datasets, which will be detailed in the next section. To realize the adaptive clustering of mixed-attribute data, we proposed an adaptive mixed-attribute data clustering method based on the DPC algorithm, called AMDPC. Experimental results showed that the proposed AMDPC algorithm had a better clustering effect, automatically determined the cluster number, and realized adaptive clustering of mixed-attribute datasets with no parameter.

For this paper, the main contributions are as follows:

(1) The distance-measurement method of mixed-attribute data is studied, and a unified distance-measurement method is used to construct the distance matrix between data points of mixed-attribute data, to solve the problem of using the DPC algorithm to cluster mixed-attribute data.

(2) The adaptive improvement of the DPC algorithm is studied and a new method to determine the center of the cluster is proposed. Because the cluster center is usually a data point with large local density and distance, after calculating $\gamma_i = \rho_i \times \delta_i$ the cluster center is determined by calculating the inflection points of the sorted γ_i, ρ_i, and δ_i sets.

(3) An adaptive local density calculation method based on K-nearest neighbor (KNN) is used to improve the robustness of the algorithm, without manually determining the cutoff distance d_c and other parameters.

2. Related Work

2.1. Density Peaks Clustering. The DPC algorithm is based on the following two assumptions: the cluster center point has a higher local density, which is surrounded by neighbor points with lower local density, and the cluster center point is relatively far from other denser data points. Therefore, the DPC algorithm constructs a decision graph by calculating a local density ρ_i and a relative distance δ_i to find the cluster center of a dataset. The remaining data points in the dataset will be assigned to the cluster with the nearest local density that is higher than its own.

Suppose that $X = \{X_1, X_2, \ldots, X_n\}$ is the dataset to be clustered consisting of n data points; we define the distance between the data points X_i and X_j as $d_{ij} = \mathrm{dist}(X_i, X_j)$. A cutoff distance d_c is defined in the DPC algorithm, and the local density ρ_i and the distance δ_i of each data point are defined, as shown in equations (1) and (2), where $\chi(d_{ij} - d_c) = 1$ when $d_{ij} - d_c < 0$ and 0 otherwise.

$$\rho_i = \sum_j \chi\left(d_{ij} - d_c\right), \tag{1}$$

$$\delta_i = \begin{cases} \min\limits_{j:\ \rho_j > \rho_i} \left(d_{ij}\right) & \rho_i < \max_k\left(\rho_k\right), \\ \max_j\left(d_{ij}\right) & \rho_i = \max_k\left(\rho_k\right). \end{cases} \tag{2}$$

When the local density of X_i is not the maximum, the relative distance is the minimum value of the distance from this point to all points with higher density; otherwise, the relative distance is the maximum distance from this point to all other points.

When the dataset has few data points, the local density is generally calculated using a Gaussian kernel, as shown in

$$\rho_i = \sum_{j \neq i} \exp\left(-\left(\frac{d_{ij}}{d_c}\right)^2\right). \tag{3}$$

Based on the local density ρ_i and relative distance δ_i for each data point, users can explicitly choose the cluster centroids on the decision graph. Once the center point is determined, each remaining data point can be classified into the same cluster as its nearest neighbor with a higher density.

2.2. Adaptive Improvement of the DPC Algorithm. The original DPC algorithm has some drawbacks: the cutoff distance d_c and cluster centroids selected manually have great influence on the clustering results, and the original method of local density calculation is not effective for data with different density clusters or different shapes. At the same time, the original sample allocation strategy will create a domino effect. Once a sample is misallocated, it will lead to a series of sample allocation errors, resulting in incorrect clustering results and a reduction in the reliability of the clustering results [20, 21]. Therefore, many adaptive improvement methods of the original DPC clustering algorithm have been proposed. The research on the adaptive improvement of the density peaks algorithm mainly focuses on the automatic determination of the cutoff distance d_c, the calculation of adaptive local density, the design of adaptive distance measurement, and the automatic determination of cluster number (the selection of cluster centroids). Most of these studies are focused on the clustering of numerical attribute data.

2.2.1. Adaptive Improvement for Selection of Cutoff Distance and Local Density Calculation. Wang et al. [22] proposed a

method to automatically extract the optimal value of the threshold in different kernel functions and different datasets from the original dataset by using the potential entropy of the data field. According to the characteristics of the dataset, Jiang et al. [23] used the change of the nearest-neighbor distance curve to automatically determine the density threshold d_c and used this method to guide the clusters merging after the first clustering using DPC. This was done to solve the problem that the DPC algorithm will divide a cluster into multiple clusters when there are two or more density peaks in a cluster. Sun et al. [24] proposed an ADPC method with a Fisher linear discriminant. The Pearson correlation coefficient is first introduced as the weight, and then the kernel-density-estimation function based on the weighted Euclidean distance is used to calculate the local density between the samples. Lotfi et al. [25] proposed a novel dynamic density peaks clustering method based on density backbone and fuzzy neighborhood called DPC-DBFN in which a fuzzy kernel is proposed to compute the local densities of the data points. Parmar et al. adopted the residual error computation to measure the local density within a neighbourhood region and proposed residual error-based density peak clustering algorithm named REDPC [26, 27] and FREDPC [28].

Du et al. [29] proposed a DPC-KNN algorithm that introduced K-nearest-neighbor data to participate in the local density calculation. In addition, they also proposed an improved principal component analysis- (PCA-) based algorithm named the DPC-KNN-PCA algorithm for high-dimensional data clustering. Juanying et al. proposed KNN-DPC [20] and FKNN-DPC [21] algorithms, in which a uniform local density metric based on KNNs, fuzzy KNNs, and two new strategies for assigning the remaining points to their most likely clusters are proposed for both. Yaohui et al. [30] proposed an adaptive DPC algorithm (named ADPC-KNN), which introduced the idea of KNNs to calculate the global parameter d_c and the local density ρ_i of each point, applied a new approach to automatically select the initial cluster centers, and finally aggregated the clusters if they were reachable in density. Shi et al. [31] presented an algorithm called the adaptive clustering algorithm based on KNN and density (ACND) that first determines the KNN of every data point and then redefines the similarity between pairs of points with shared nearest neighbors. It does not force the user to define parameter values, recognizes the core point and constructs the cluster around it, and then attempts to detect the clustering boundary. It makes full use of the effect of KNN, and it has low computational complexity and can deal with different shapes as well as different data sizes with noise and outliers. Xu et al. [32] proposed extended adaptive density peaks clustering (EDAP) for overlapping community detection in which the local density is calculated based on KNN. Jiang et al. [33] proposed a method called G-KNN-DPC to calculate the cutoff distance based on the Gini coefficient and KNN. Sun and Liu [34] proposed a new density formula combined with the idea of gravitation and KNN that can make the local densities of sample points in dense and sparse areas have more obvious separability. Fan et al. [35] proposed a new DPC algorithm by incorporating an improved mutual K-nearest-neighbor graph (Mk-NNG) into DPC.

In general, KNN is used for local density calculations in most of these improved algorithms. Letting $d(X_i, X_j)$ be the Euclidean distance between the ith and jth data points in the dataset $X = \{X_1, X_2, \ldots, X_n\}$, the local density calculation formula defined by DPC-KNN is expressed by

$$\rho_i = \exp\left(-\frac{1}{K}\sum_{X_j \in KNN(X_i)} d^2\left(X_i, X_j\right)\right), \tag{4}$$

and the local density calculation formula defined in the KNN-DPC and FKNN-DPC algorithms is expressed as follows:

$$\rho_i = \exp\left(-\sum_{X_j \in KNN(X_i)} d^2\left(X_i, X_j\right)\right), \tag{5}$$

where $\mathrm{KNN}(X_i)$ represents the KNN set of data point X_i. In general, k takes a fixed value of 5 or 6, or is calculated according to the percentage of data points in the dataset. In most cases, $k = p * N$, where the percentage $p = 2$, N is the total number of data points in the dataset, and $*$ is a ceiling function. Most of these algorithms are based on equations (4) and (5) or variants.

2.2.2. Automatic Determination of Cluster Number. To solve the problem that the density peaks algorithm must manually select the cluster center, Ma et al. [36] introduced the weight of the cluster center. First, the products γ ($\gamma_i = \rho_i \times \delta_i$) of the normalized adjacent distance δ_i and the local density ρ_i were calculated. Then, the inflection point of γ was used to determine the cluster center of the dataset to avoid the subjective difference of users' selection of the cluster center. Zhao [37] proposed an improved LDPC algorithm combined with the linear fitting method. In this algorithm, the sparse and dense points are separated by the linear fitting method, and then the residual sequence C is obtained by making a difference between the original γ_s and the fitting value γ_r. The average residual value of the first 20 points is selected as the threshold value, and the data points with residual values greater than the threshold are the central points. Du et al. [38] proposed a parameter-adaptive clustering algorithm named DDPA-DP. The data-driven thought goes through the design of DDPA-DP: at first, a series of fitted curves are established to automatically detect points' roles by points' density attributes instead of any artificial thresholds; meanwhile, a new point's role "pending point" is defined, and then by the change of pending points' number, the local field's radius can be adaptively optimized. García-García and García-Ródenas [39] proposed an optimization-based methodology for automatic parameter/center selection that uses the internal/external cluster validity index as the objective function.

Wang et al. [40] proposed an efficient hierarchical clustering algorithm based on density peaks, used the step characteristics of the parameter γ to distinguish different

levels of clustering, and then constructed a hierarchical clustering tree based on the intermediate result of DPC (NNeigh, a DPC array) to complete efficient hierarchical clustering and determined the cluster number automatically. Zhang and Li [41] extended the traditional DPC algorithm by using the CHAMELEON hierarchical clustering algorithm. The DPC algorithm was used for the initial clustering in the extended algorithm, and then the hierarchical clustering algorithm was used to merge the subclasses for the clustering results, and the effect was improved. Bie et al. [42] proposed a fuzzy DPC algorithm called Fuzzy-CFSFDP that uses fuzzy rules to find all density peaks and treats each peak as a local cluster, and then merges the close local clusters into a global cluster to achieve the final cluster. Ding et al. [43] proposed an improved density peaks clustering based on a natural neighbor expanded group (DPC-NNEG). They first define a natural neighbor expanded (NNE) and a natural neighbor expanded group (NNEG) and then divide all NNEGs into a target number of sets as the final clustering result according to the degree of closeness of the NNEGs. To describe the clustering center more comprehensively, Diao et al. [44] redefined the local density and relative distance and distance attributes of the two neighbor relationships (KNN and SNN) as fused. This method can detect the low-density clustering center. Mehrmohammadi et al. [45] proposed a better method for selecting centers based on the mutual kNN graph and the shortest path. Fang et al. [46] proposed adaptive core fusion-based density peak clustering (CFDPC) to detect clusters in any shape and density adaptively. An initial clustering based on automatic finding of density peaks is proposed first. An adaptive search approach is then proposed to find the core points and a core fusion strategy based on similarity within the cluster is proposed to obtain the final clustering results.

In summary, the main improvement ideas of automatic determination of cluster number can be categorized as following two directions. One is to determine the cluster center by taking a larger value of γ, ρ, and δ, such as finding the inflection point of γ or using curve fitting and residual analysis. The second is to adopt the idea of hierarchical clustering, initially selecting more clustering centers and then merging the close local clusters.

Notably, γ, ρ, and δ are discrete sequences, and for the calculation of the inflection point of the discrete sequence, Ma et al. [36] used the slope of the line segment at two points to represent it. The calculation formula is expressed as follows: S_i^m represents the average change rate of the discrete sequence in the interval $[i, i+m]$, namely, the slope change of y in the interval $[i, i+m]$.

$$S_i^m = \frac{y_{i+m} - y_i}{m}. \tag{6}$$

Based on the slope calculation, the inflection point is defined as follows:

$$i = \operatorname{argmax}\left(\frac{S_i^1}{S_1^{i-1}}\right), \tag{7}$$

where S_i^1 is the slope from the ith point to the ith +1 point, S_1^{i-1} is the slope from the first point to the ith point, and S_i^1/S_1^{i-1} represents the average change rate of the discrete sequence y in the interval $[1, i]$. In this case, the inflection point is the critical point with the fastest slope change.

3. Adaptive Mixed-Attribute-Data Density Peaks Clustering

3.1. Definition of Unified Distance Metric of Mixed-Attribute Data. Suppose that $DS = \{X_1, X_2, \ldots, X_n\}$ is a mixed dataset with d dimensions and n instances, which contains d_r dimensional numerical attributes and d_c ($d_c = d - d_r$) dimensional categorical attributes, for two instances X_i and X_j in the dataset; their distance is defined as $D(X_i, X_j)$ as follows:

$$D\left(X_i, X_j\right) = D_r\left(X_i, X_j\right) + D_c\left(X_i, X_j\right). \tag{8}$$

Equations (9) and (10) illustrate the distance computation of the numerical attribute $D_r(X_i, X_j)$ and that of the categorical attribute $D_c(X_i, X_j)$, respectively:

$$D_r\left(X_i, X_j\right) = 1 - e^{-\operatorname{dist}\left(X_i^{d_r}, X_j^{d_r}\right)}, \tag{9}$$

$$D_c\left(X_i, X_j\right) = \sum_{t=1}^{d_c} \omega_t * \delta\left(x_{it}, x_{jt}\right), \tag{10}$$

where $\operatorname{dist}(X_i^{d_r}, X_j^{d_r})$ denotes the normalized Euclidean distances of the numerical attribute of the data points X_i, X_j. Because the Euclidean distance is non-negative, it is ensured that the distance value of the numerical attribute is in the interval [0,1]. Regarding the distance of the categorical attribute, the matching method with the entropy weight is used. The matching distance of the data point X_i, X_j in the tth categorical attribute is calculated by

$$\delta\left(x_{it}, x_{jt}\right) = \begin{cases} 0, & \text{if } \left(x_{it} = x_{jt}\right) \\ 1, & \text{if } \left(x_{it} \neq x_{jt}\right) \end{cases}. \tag{11}$$

The importance of a categorical attribute is quantified by its average entropy on each attribute value. The weight of each attribute ω_t is then computed by

$$\omega_t = \frac{H_{A_t}}{\sum_{s=1}^{d_c} H_{A_s}}. \tag{12}$$

Assume that the total number of categorical values on the tth categorical attribute is m_t, where the probability of occurrence of the sth ($s = 1,2, \ldots, m_t$) values is $p(a_{ts})$. The entropy weight H_{A_t} can be calculated using equation (13); it represents the average entropy of m_t values of tth classification attribute:

$$H_{A_t} = -\frac{1}{m_t} \sum_{s=1}^{m_t} p\left(a_{ts}\right) \log\, p\left(a_{ts}\right). \tag{13}$$

Assuming a mixed-attribute dataset about the weather record is shown in Table 1, the dataset DS = $\{X_1,X_2,X_3,X_4,X_5\}$ has five records X_1–X_5 and four attributes A_1–A_4. The four

Table 1: A sample mixed-attribute dataset.

Records	Weather (A_1)	Windy (A_2)	Temperature (A_3)	Humidity (A_4)
X_1	Sunny	No	36	0.9
X_2	Sunny	Yes	35	0.8
X_3	Cloudy	No	34	0.9
X_4	Rainy	No	26	0.9
X_5	Rainy	No	10	0.7

attributes represent weather, windy, temperature, and humidity: the first two attributes weather and windy are categorical attributes and the last two are numerical ones. Here $d_r = 2$, $d_c = 2$. Let us look at the calculation process of the unified distance metric.

Firstly, it is necessary to normalize the numerical attributes A_3 and A_4; the results are $A_3 = [1.0000, 0.9615, 0.9231, 0.6154, 0]^T$, $A_4 = [1, 0.5, 1, 1, 0]^T$, Then formulas (12) and (13) are used to calculate the weights of the first and second dimensions as $\omega_1 = 0.5708$ and $\omega_2 = 0.4292$, respectively. Finally, the distance between the first record and other records can be calculated according to formula (8) as follows:

$$\begin{aligned} D(X_1, X_2) &= 1 - \exp\left(-\sqrt{(1-0.9615)^2 + (1-0.5)^2}\right) + 0.4292 = 0.8235, \\ D(X_1, X_3) &= 1 - \exp\left(-\sqrt{(1-0.9231)^2 + (1-1)^2}\right) + 0.5708 = 0.6449, \\ D(X_1, X_4) &= 1 - \exp\left(-\sqrt{(1-0.6154)^2 + (1-1)^2}\right) + 0.5708 = 0.8901, \\ D(X_1, X_5) &= 1 - \exp\left(-\sqrt{(1-0)^2 + (1-0)^2}\right) + 0.5708 = 1.3277. \end{aligned} \tag{14}$$

3.2. Local Density Calculation Based on KNN. In a small dataset, the Gaussian kernel function is usually used to calculate the local density, which requires manually setting the density threshold parameter d_c. As mentioned above, to adaptively calculate the local density, many studies have adopted KNN information to improve the calculation of the local density. We adopt the idea of the DPC-KNN algorithm and use the improved Gaussian kernel function of KNN information to calculate the local density of each data point. Using the KNN set of data points, we can calculate the average of the sum of squares of distance between each data point and KNNs. Thus, equation (4) can be used to calculate the local density of data points. In this calculation method, it is not necessary to set the cutoff distance parameter d_c, but rather to determine the nearest-neighbor number K. Through subsequent experiments, the nearest-neighbor number K can be automatically determined based on the data points in the dataset.

3.3. Automatically Determining the Cluster Number. To realize the automatic determination of the cluster number, the method of calculation of the inflection point of γ proposed by Ma et al. [36] is simple but not sufficiently accurate. In theory, the center points should be points with large local density ρ_i and large relative distance δ_i, and the product of the two γ_i does not fully guarantee that the local density and the relative distance are both large.

From the sample dataset of the DPC algorithm and its decision graph [10] in Figure 1, points 1 and 10 in the upper right corner of the decision graph are cluster centers for which the local density and relative distance are both large. Points 26, 27, and 28, however, are treated as outliers for which the relative distance is large, but the local density is small. Therefore, we presented a three-inflection-point improvement method, which is based on equations (2) and (4) to calculate the local density and distance of the data points. We then sorted the γ, ρ, and δ values of each data point by descending order and used equation (7) to calculate the inflection point of γ, ρ, and δ, and to obtain three candidate sets S_g, S_p, and S_d according to three inflection points of γ, ρ, and δ, respectively. The candidate set S_g contains the points with γ values that are larger than those of the inflection point of γ. Similarly, the candidate set S_p contains the points for which the value of ρ is larger than that of the inflection point of ρ, and the candidate set S_d contains the points with δ values larger than those of the inflection point of δ. Then we calculated the intersections $S_c = S_g \cap S_p \cap S_d$, and S_c is the cluster center set. Points for which the relative distances are larger, but not the cluster center, can be judged as the outliers, which can be obtained by calculating $S_o = S_d - S_c$. Therefore, the improved method proposed in this paper could automatically identify the cluster center and outlier point.

For example, the local density ρ_i, relative distance δ_i, and γ_i of partial data points of a sample dataset are shown in Table 2. According to equation (7), we can calculate $S_d = \{1,2,3,4,5,6,8\}$, $S_p = \{1,2,3,4,5,6,7,10\}$, $S_d = \{1,2,3,4,5,6,7\}$, and then $S_c = S_g \cap S_p \cap S_d = \{1,2,3,4,5,6\}$; it contains the cluster centers. At the same time, we can get the outlier $S_o = S_d - S_c = \{8\}$.

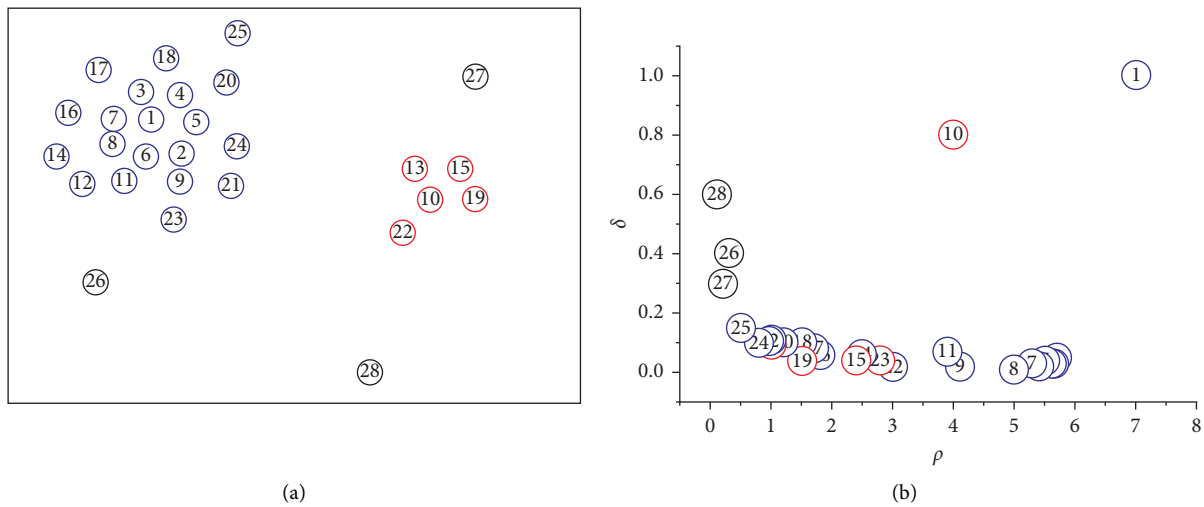

Figure 1: Sample dataset of the DPC algorithm and its decision graph [10].

Table 2: Partial data points of a sample dataset.

i	1	2	3	4	5	6	7	8	9	10	11
δ_i	0.9	0.89	0.9	0.88	0.88	0.88	0.75	0.88	0.7	0.5	0.62
ρ_i	0.88	0.87	0.85	0.83	0.8	0.78	0.88	0.42	0.65	0.88	0.5
γ_i	0.792	0.7743	0.765	0.7304	0.704	0.6864	0.66	0.3696	0.455	0.44	0.31

The three-inflection-point algorithm to determine the center of the cluster is described as Algorithm 1.

3.4. AMDPC Implementation. First, we used the unified distance measurement of the mixed-attribute data to calculate the distance matrix of the mixed-attribute dataset according to equation (8). Then, we calculated the local density ρ_i of each data point using the KNN equation (4) and calculated the distance δ_i using the method of equation (2); thus, $\gamma_i = \rho_i \times \delta_i$ is calculated and the cluster centers are found using Algorithm 1. Finally, the remaining points could be clustered by finding the nearest local point with densities higher than it and setting the clustering label to be consistent with its nearest-neighbor point with high density. The overall flow diagram of the AMDPC algorithm is shown in Figure 2.

The input of the algorithm is the mixed-attribute dataset (DS) and the output is the cluster label vector (CL). The detailed process of the AMDPC algorithm is as Algorithm 2.

3.5. Complexity Analysis. For datasets with n data points, the space complexity of the algorithm is mainly from the storage of distance matrix. According to the input demand of DPC algorithm, $3 * n * (n-1)/2$ storage space is needed. Columns 1 and 2 are the data point numbers and column 3 is the distance between the two data points. In addition, the algorithm requires three arrays of length n to store the local density ρ, distance δ, and its product γ, so the space complexity is $O(n^2)$.

The time complexity of the AMDPC algorithm is mainly derived from distance calculation in Step 2 and the local density computation in Step 3. The time complexity of distance computation and its product calculation is $O(n^2)$. The sort time complexity in Step 4 (Algorithm 1) depends on the sorting algorithm, the minimum $O(n \log(n))$, and the largest $O(n^2)$, so the total complexity is no more than $O(n^2)$. The time complexity of the data point allocation in Step 5 is $O(n)$. Therefore, the overall complexity of the algorithm is $O(n^2)$, and it is the same as the DPC algorithm.

4. Experimental Analysis

To verify the effectiveness of the AMDPC algorithm in this paper, we used several mixed datasets from the University of California-Irvine (UCI) for experimental study. We compared the clustering results of the AMDPC algorithm with those of the K-prototype and DPC_M algorithms.

We implemented the three algorithms in MATLAB 2015a (MathWorks, USA) running on Windows 10 on a laptop with Intel Core i5-5200u model CPU and 4 GB of DDR3 memory.

4.1. Experimental Datasets. In this study, we investigated four datasets of mixed datasets from the UCI machine-learning repository, namely, Statlog Heart, Cleveland Heart Disease, Statlog Credit Approval, and Acute Inflammations. Brief information describing these datasets is shown in Table 3.

The Acute Inflammations dataset contains pathological and physiological indicators for 120 patients with acute inflammation. There is one numerical attribute (body temperature) and five categorical attributes (different symptoms) to determine whether each patient has cystitis and nephritis. There are two class labels to represent the two diseases. We used the first to predict cystitis in our experiments. The

Input: rho, delta (represent local density vector ρ and relative distance vector δ)
Output: Sc (set of cluster *centers* S_c)
(1) //Step 1. Calculate $\gamma_i = \rho_i * \delta_i$.
(2) **for** $i = 1$ to length(rho) **do**
(3) gamma(i) = rho(i) $*$ delta(i)
(4) end
(5) //Step 2. Sort rho, delta, gamma (ρ, δ, γ) in descending order:
(6) Sorted_rho = sort (rho, "descend");
(7) Sorted_delta = sort(delta, "descend");
(8) Sorted_gamma = sort(gamma, "descend");
(9) //Step 3. Calculate the inflection point of rho, delta, gamma using equation (7) separately and construct the three candidate sets S_g, S_p, and S_d.
(10) S_p = calcinflection(Sorted_rho);
(11) S_d = calcinflection(Sorted_delta);
(12) S_g = calcinflection(Sorted_gamma);
(13) //Step 4. Calculate the intersections of the three sets and return the result Sc.
(14) S_c = intersection(S_p,S_d,S_g).

Algorithm 1: Find cluster center.

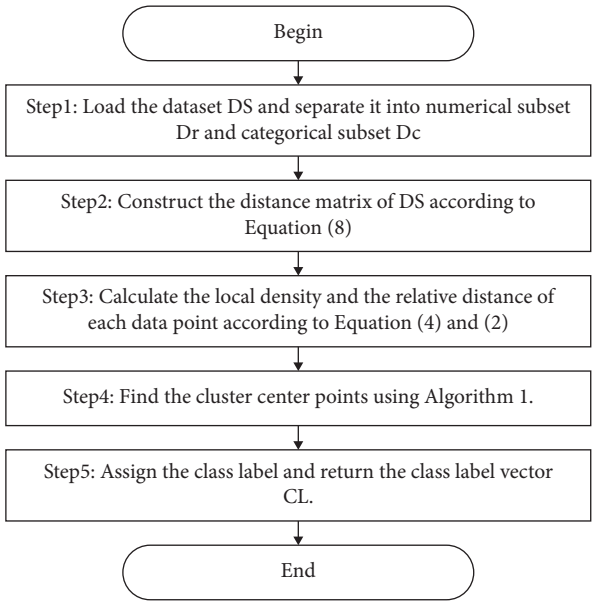

Figure 2: The flow chart of the AMDPC algorithm.

deletion of missing data in the dataset did not affect the result of clustering analysis. Therefore, we eliminated 6 instances with missing values in the Cleveland dataset and 37 instances with missing values in the Credit dataset before the experiment. The Adult dataset was extracted from the census bureau database, which contain 30162 training instances. We selected 3000 of them by random sampling. In addition, we normalized the numerical properties using the maximum-minimum normalization method.

4.2. Effectiveness Analysis. We used the K-prototype algorithm, DPC_M, and the proposed AMDPC algorithm to separately cluster the dataset described in Section 4.1. According to the research in [5], the parameter γ of the K-prototype algorithm was $1/2\sigma$ (σ represents the average standard deviation of the numerical attributes). The K-prototype algorithm ran 100 times and the clustering results were averaged. In the DPC_M algorithm, the percent parameter $p = 2$, as described in [15]. When the AMDPC algorithm calculated the local density, the parameter K was assigned as ceil$(0.1 * N)$; that is, 10% of the data points were taken as the nearest neighbors.

Because the UCI datasets have real class labels, the clustering accuracy rate (ACC) can be used as the validity index. We also used the normalized mutual information (NMI), Rand index (RI), adjusted Rand index (ARI), and F-score as validity indexes. For all indexes, the higher the index values, the better the clustering effect. The optimal results are indicated in bold in Tables 4–8.

Accordingly, we observed that the performance of the AMDPC algorithm was much better than that of the traditional K-prototype algorithm. The AMDPC algorithm improved the clustering accuracy of all datasets by more than 22.58%, by 24.25%, by 28.03%, by 22.5%, and by 10.12% for the Heart, Cleveland, Credit, Acute, and Adult datasets, respectively. It also outperformed the DPC_M algorithm in the first four datasets as shown in Tables 4–7. In the Adult dataset, the clustering accuracy of the AMDPC algorithm was 0.43% worse than that of the DPC_M algorithm, but it is better than the DPC_M algorithm in the NMI index and the ARI index. The F-score takes into account both precision and recall; the value of F-score shows that different algorithms perform differently in different experimental datasets. The proposed AMDPC algorithm got the best performance in Credit dataset.

As shown in Table 7, for the first four indexes, the DPC_M algorithm had two different results because of the different selection of center points. The DPC_M2 was worse than the AMDPC algorithm in the clustering effect, whereas the DPC_M1 was better. This showed that the selection of the center point in the DPC algorithm had a significant

Input: DS (the mixed-attribute dataset)
Output: CL (cluster label vector)
(1) //Step 1. Load the dataset DS and separate it into numerical subset Dr, categorical subset Dc and the ture label subset.
(2) [Dr,Dc,truelabel] = loadseparate(DS);
(3) //Step 2. Calculate the distance and construct the distance matrix of the mixed-attribute dataset DS according to equation (8).
distmatrix = distamdpc(Dr,Dc);
(4) //Step 3. Calculate the local KNN density ρ_i of each data point according to equation (4) and calculate the relative distance δ_i according to equation (2).
(5) rho = kNNrho(distmatrix);
(6) delta = calcdelta(distmatrix);
(7) //Step 4. Run Algorithm 1 to obtain the cluster center points and set each point a different label.
(8) Sc = findClusterCenter(rho,delta);
(9) //Step 5. Assign the class label for center and non-center points using original DPC method according to the Sc.
(10) //Step 5.1. Initialize the class label vector CL.
(11) NCLUST = 0;
(12) for $i = 1$ to number of datapoints
(13) CL(i) = −1;
(14) End
(15) //Step 5.2. Assign the class label for center points
(16) for $j = 1$ to sizeof(Sc)
(17) NCLUST = NCLUST + 1;
(18) CL(Sc(j)) = NCLUST;
(19) End
(20) //Step 5.3. Assign the class label for non-center points
(21) for $k = 1$ to number of datapoints
(22) if (CL(ordrho(k)) = = −1)
(23) CL(ordrho(k)) = CL(nneigh(ordrho(k))); //assign the non-center data points to the cluster with the nearest local density that is higher than its own.
(24) end

Algorithm 2: AMDPC_clustering.

Table 3: Brief information of UCI mixed datasets.

Datasets	Instances	Attributes ($d_c + d_r$)	Class	Description
Heart	270	7 + 6	2	
Cleveland	297	7 + 6	2	Original class labels 1–4 are treated as 1
Credit	653	9 + 6	2	Thirty-seven incomplete instances are deleted
Acute	120	5 + 1	2	One class label is used
Adult	3000	8 + 6	2	Use 3000 samples from 30162 instances

Table 4: Clustering results of Heart dataset.

Algorithm	ARI	RI	NMI	ACC	F-score
K-prototype	0.0303	0.5154	0.0202	0.5927	0.5541
DPC_M	0.3751	0.6878	0.3107	0.8074	0.5717
AMDPC	0.4032	0.7018	0.3251	0.8185	0.5622

Table 5: Clustering results of Cleveland dataset.

Algorithm	ARI	RI	NMI	ACC	F-score
K-prototype	0.0195	0.5098	0.0139	0.5758	0.5565
DPC_M	0.3609	0.6805	0.3135	0.8013	0.5730
AMDPC	0.4028	0.7015	0.3362	0.8183	0.5620

influence on the clustering effect, whereas the AMDPC algorithm automatically determined the center point of the clustering, and the algorithm ran more stably. The decision graphs for center selection by the two aforementioned clustering types using the DPC_M algorithm (that is, DPC_M1 and DPC_M2) are shown in Figure 3.

Table 6: Clustering results of Credit dataset.

Algorithm	ARI	RI	NMI	ACC	F-score
K-prototype	0.0026	0.5048	0.0303	0.5528	0.6680
DPC_M	0.3875	0.6938	0.2988	0.8116	0.7006
AMDPC	0.4428	0.7215	0.3475	0.8331	0.7157

Table 7: Clustering results of Acute dataset.

Algorithm	ARI	RI	NMI	ACC	F-score
K-prototype	0.039	0.5195	0.0358	0.6083	0.9414
DPC_M1	0.4629	0.7312	0.4932	0.8417	0.8200
DPC_M2	0.1151	0.5576	0.0903	0.6750	0.9917
AMDPC	0.2609	0.6304	0.2091	0.7583	0.7399

Table 8: Clustering results of the Adult dataset.

Algorithm	ARI	RI	NMI	ACC	F-score
K-prototype	−0.0042	0.5225	0.0001	0.6365	0.6476
DPC_M	0.1956	0.6170	0.0931	0.7420	0.6442
AMDPC	0.2062	0.6128	0.1179	0.7377	0.6133

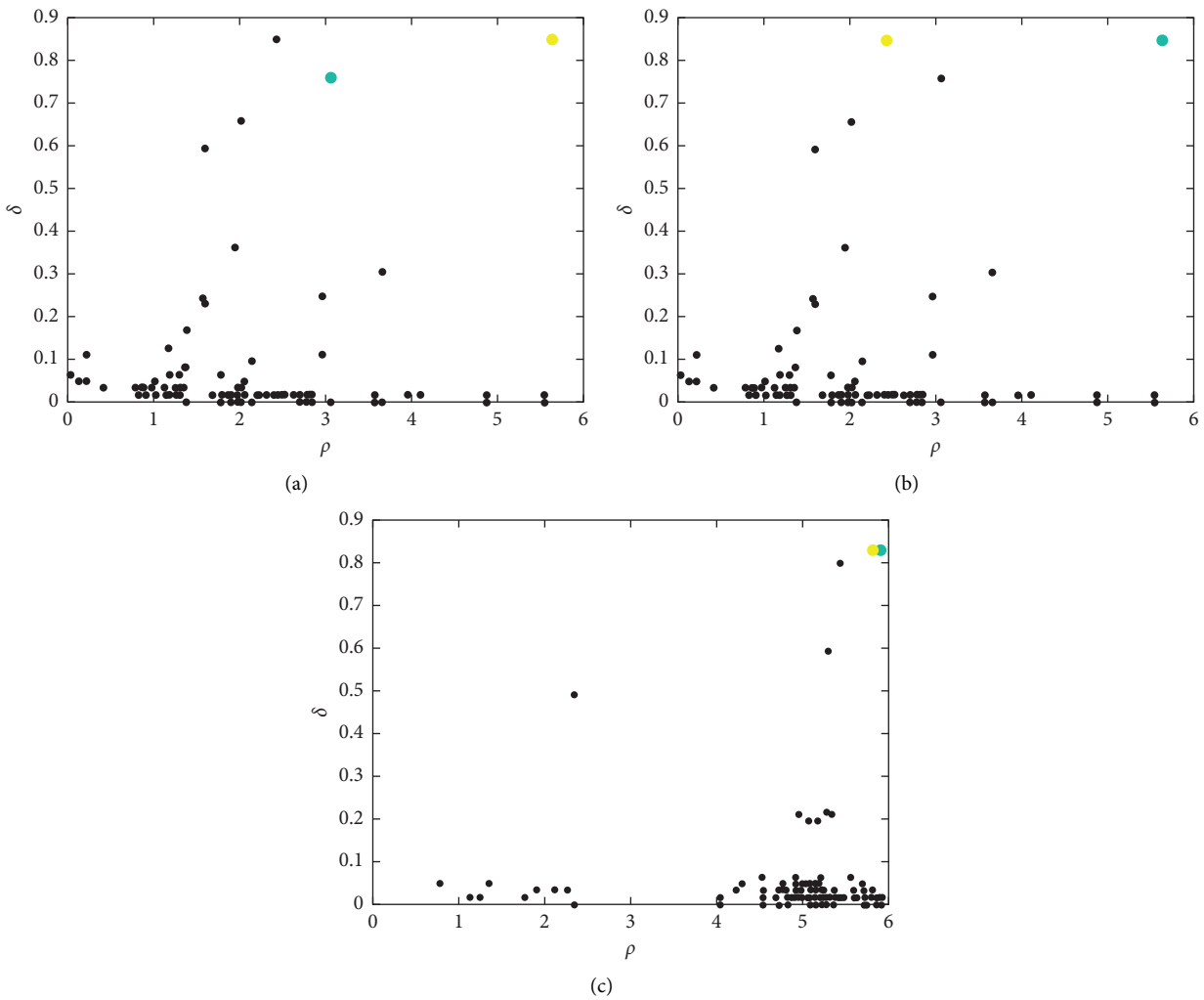

Figure 3: Decision graphs of the DPC_M and AMDPC algorithm. (a) Decision graph of DPC_M1. (b) Decision graph of DPC_M2. (c) Decision graph of AMDPC.

Table 9: Clustering results of the Heart dataset with different K values.

P	1%	2%	3%	4%	5%	6%	7%
K	3	6	9	11	14	17	19
ACC	0.7556	0.7926	0.7926	0.7963	0.7963	0.7963	0.7963
P	8%	9%	10%	12%	15%	20%	
K	22	25	27	33	41	54	
ACC	0.8148	0.8111	0.8185	0.8185	0.8185	0.8111	

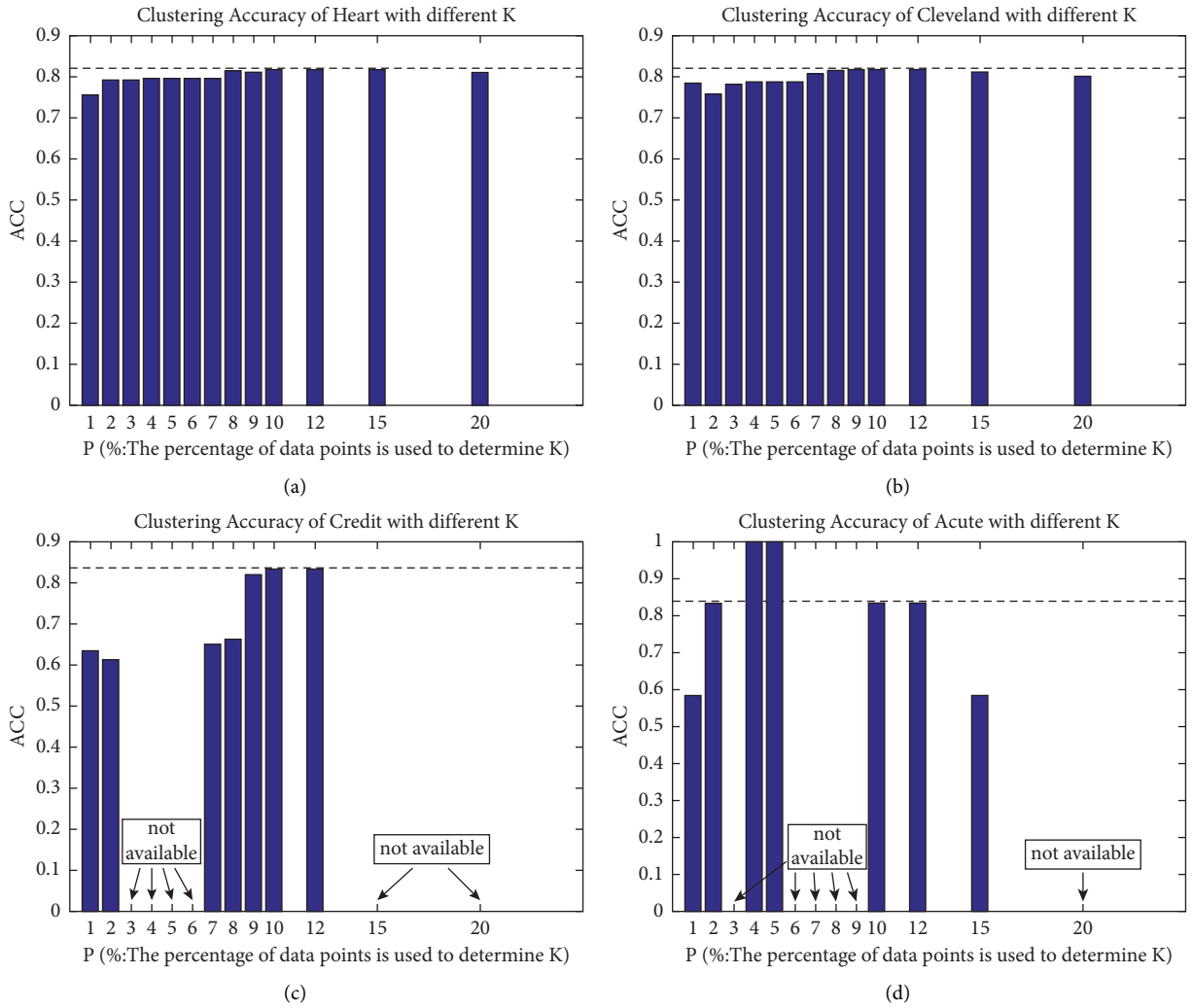

Figure 4: Clustering accuracy of datasets according to different K values. (a) Clustering accuracy of Heart with different K. (b) Clustering accuracy of Cleveland with different K. (c) Clustering accuracy of Credit with different K. (d) Clustering accuracy of Acute with different K.

4.3. Parameter-Adjustment Experiment. The AMDPC algorithm uses the improved Gaussian kernel function of the KNN information to calculate the local density of each data point. To understand how the parameter K affects the effectiveness of the algorithm, we conducted a series of experiments and found that the best effect was obtained when K was approximately 10% of the data instances in the dataset.

Taking the Heart dataset as an example, we had 270 data points in total. We took K as 1–20% of the data points to calculate the clustering accuracy of the AMDPC algorithm. The results are presented in Table 9 ; optimal results are indicated in bold).

As shown in Table 9, when K was 10% of the data points ($K = 27$), the clustering accuracy reached the best value. As shown in Figure 4, in the Heart and Cleveland datasets, K took 10% of the data points to achieve the best effect. In the Credit and Acute datasets, some values of K would have led to the incorrect clustering number, so the value of clustering accuracy (ACC = 0) was marked "not available" in the graph. For Acute dataset, $K = 4$ or 5 was the best, and 10% was also

Table 10: Running time of the K-prototype, DPC_M, and proposed AMDPC in different datasets.

Algorithms	Heart	Cleveland	Credit	Acute	Adult
K-prototype	0.0982	0.1110	0.1104	0.0347	0.6343
DPC_M	0.1131	0.0969	0.3445	0.0383	17.8739
AMDPC	0.3001	0.3381	2.0599	0.1386	34.1307

Table 11: Time-complexity comparison between AMDPC and DPC-M.

Datasets	Heart	Cleveland	Credit	Acute	Adult
Instances	270	297	653	120	3000
DPC_M	0.1131	0.0969	0.3445	0.0383	17.8739
AMDPC	0.3001	0.3381	2.0599	0.1386	34.1307
The time ratio:(AMDPC/DPC_M)	2.6534	3.4892	5.9794	3.6188	1.9095

good. Therefore, we determined that the value of K in the AMDPC algorithm is 10% of the data points in the dataset.

4.4. Computational Complexity Experiment. To verify the time complexity of the proposed algorithm, we calculated the running time of the above three algorithms. The K-prototype algorithm was run 100 times and the running times were averaged. The running times are shown in Table 10.

As shown above, the K-prototype is the most efficient algorithm. The proposed AMDPC needs more time to calculate distance and compute local density. With an increase in data volume, the time consumption of the AMDPC algorithm and the DPC_M algorithm presents a linear relationship, and the time complexity of the two algorithms is of the same order of magnitude, which is consistent with the previous theoretical analysis. As shown in Table 11, when there are 120 points in the Acute dataset, the clustering time used by AMDPC is 3.6 times that of the DPC_M algorithm. When the data amount increases to 653 points (in Credit), the clustering time used by AMDPC is about 6 times that of DPC_M algorithm. When the data amount increases to 3000 points (in Adult), the clustering time used by AMDPC is less than 2 times that of DPC_M algorithm.

5. Conclusion

The DPC algorithm is a simple and efficient algorithm. As long as the distance-measurement problem of data points in the mixed-attribute dataset is solved, the DPC algorithm also can be used for efficient clustering of mixed-attribute data. In this paper, we study the clustering methods of mixed-attribute data, focusing on the DPC algorithm and its adaptive improvement. Accordingly, we proposed an adaptive mixed-attribute data clustering algorithm based on DPC called AMDPC that adopted a unified mixed-attribute distance-measurement method and KNN adaptive local density calculation method. We used three inflection points to calculate the cluster center set and automatically determined the clustering number, which realized adaptive clustering of mixed-attribute datasets. From the analysis of experimental results, the proposed algorithm was significantly superior to the traditional K-prototype and DPC_M algorithms. In all five datasets, the clustering accuracy of the AMDPC algorithm is significantly improved compared with that of the K-prototype algorithm, by 10.12% to 28.03%, and also slightly improved compared to the DPC_M algorithm except in the Adult dataset. In addition, AMDPC implements adaptive clustering without manual adjustment of any parameters.

The AMDPC algorithm could realize adaptive clustering of mixed-attribute data well. When we used KNN to calculate the local density of the data points, the determination of K was different from the value in the previous research paper [10, 14], and the value of K also had a significant influence on the effect of cluster. According to the experimental analysis, the effect was optimal when K was 10% of the data points, but there was still room to adjust the value of K on different datasets, which requires further research. There are still many problems in adaptive clustering of mixed-attribute data to be further studied, such as mixed-attribute data clustering on the datasets containing a huge number of objects or a huge number of attributes, or on the datasets with arbitrary shapes, different sizes, variable density, and overlapping clusters, etc.

Data Availability

Data used to support the findings of this study are available from the UCI Machine Learning Repository at https://archive.ics.uci.edu/ml/index.php, https://github.com/milaan9/Clustering-Datasets.

Conflicts of Interest

The author declares no conflicts of interest.

Acknowledgments

The author thanks LetPub (https://www.letpub.com) for its linguistic assistance during the preparation of this manuscript. This research was supported by the Qingshan Lake Science and Technology City Joint Fund of Zhejiang Provincial Natural Science Foundation of China under Grant no. LQY19F020001 and the Wenzhou Polytechnic Major Scientific Research Projects (Grant no. WZYSDCY2018002).

References

[1] Y. Chen, H. Zhang, L. Liu et al., "Research on image inpainting algorithm of improved total variation minimization method," *Journal of Ambient Intelligence and Humanized Computing*, 2021.

[2] Y. Chen, L. Liu, J. Tao et al., "The improved image inpainting algorithm via encoder and similarity constraint," *The Visual Computer*, vol. 37, no. 7, pp. 1691–1705, 2021.

[3] Y. Chen, L. Liu, V. Phonevilay et al., "Image super-resolution reconstruction based on feature map attention mechanism," *Applied Intelligence*, vol. 51, no. 7, pp. 4367–4380, 2021.

[4] S. H. Liu, L. Z. Shen, and D. C. Huang, "A three-stage framework for clustering mixed data," *WSEAS Transactions on Systems*, vol. 15, pp. 1–10, 2016.

[5] Z. Huang, "Clustering large data sets with mixed numeric and categorical values," in *Proceedings of the 1st Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pp. 21–34, Kuala Lumpur, Malaysia, May 1998.

[6] Y.-M. Cheung and H. Jia, "Categorical-and-numerical-attribute data clustering based on a unified similarity metric without knowing cluster number," *Pattern Recognition*, vol. 46, no. 8, pp. 2228–2238, 2013.

[7] C. Li and G. Biswas, "Unsupervised learning with mixed numeric and nominal data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 4, pp. 673–690, 2002.

[8] D. C. Huang and X. C. Li, "Incremental relative density-based clustering algorithm for mixture datasets," *Control and Decision*, vol. 28, no. 6, pp. 815–822, 2013.

[9] J.-Y. Chen and H.-H. He, "Density-based clustering algorithm for numerical and categorical data with mixed distance measure methods," *Control Theory & Applications*, vol. 32, no. 8, pp. 993–1002, 2015.

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

[11] D. Cheng, J. Huang, S. Zhang, X. Zhang, and X. Luo, "A novel approximate spectral clustering algorithm with dense cores and density peaks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–13, 2021.

[12] D. Cheng, Q. Zhu, J. Huang, Q. Wu, and L. Yang, "Clustering with local density peaks-based minimum spanning tree," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 2, pp. 374–387, 2021.

[13] Y. Chen, S. Tang, L. Zhou et al., "Decentralized clustering by finding loose and distributed density cores," *Information Sciences*, vol. 433-434, pp. 510–526, 2018.

[14] D. Cheng, S. Zhang, and J. Huang, "Dense members of local cores-based density peaks clustering algorithm," *Knowledge-Based Systems*, vol. 193, Article ID 105454, 2020.

[15] S. Liu, B. Zhou, D. Huang, and L. Shen, "Clustering mixed data by fast search and find of density peaks," *Mathematical Problems in Engineering*, vol. 2017, Article ID 5060842, 7 pages, 2017.

[16] M. Du, S. Ding, and Y. Xue, "A novel density peaks clustering algorithm for mixed data," *Pattern Recognition Letters*, vol. 97, pp. 46–53, 2017.

[17] S. Xia, D. Peng, D. Meng et al., "A fast adaptive k-means with No bounds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, 2020.

[18] Q. Wang, H. E. Psillakis, and C. Sun, "Adaptive cooperative control with guaranteed convergence in time-varying networks of nonlinear dynamical systems," *IEEE Transactions on Cybernetics*, vol. 50, no. 12, pp. 5035–5046, 2020.

[19] Q. Wang, H. E. Psillakis, C. Sun, and F. L. Lewis, "Adaptive NN distributed control for time-varying networks of nonlinear agents with antagonistic interactions," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 6, pp. 2573–2583, 2021.

[20] X. Juanying, G. Hongchao, and X. Weixin, "K-nearest neighbors optimized clustering algorithm by fast search and nding the density peaks of a dataset," *Scientia Sinica Informationis*, vol. 46, no. 2, pp. 258–280, 2016.

[21] J. Xie, H. Gao, W. Xie, X. Liu, and P. W. Grant, "Robust clustering by detecting density peaks and assigning points based on fuzzy weighted K-nearest neighbors," *Information Sciences*, vol. 354, pp. 19–40, 2016.

[22] S. Wang, D. Wang, C. Li, Y. Li, and G. Ding, "Clustering by fast search and find of density peaks with data field," *Chinese Journal of Electronics*, vol. 25, no. 3, pp. 397–402, 2016.

[23] L. Jiang, M. Zhang, J. Zheng, J. Dai, and Z. Shang, "Optimization of clustering by fast search and find of density peaks," *Application Research of Computers*, vol. 33, no. 11, pp. 3251–3254, 2016.

[24] L. Sun, R. Liu, J. Xu, and S. Zhang, "An adaptive density peaks clustering method with Fisher linear discriminant," *IEEE Access*, vol. 7, pp. 72936–72955, 2019.

[25] A. Lotfi, P. Moradi, and H. Beigy, "Density peaks clustering based on density backbone and fuzzy neighborhood," *Pattern Recognition*, vol. 107, Article ID 107449, 2020.

[26] M. Parmar, D. Wang, X. Zhang et al., "REDPC: a residual error-based density peak clustering algorithm," *Neurocomputing*, vol. 348, pp. 82–96, 2019.

[27] M. Parmar, D. Wang, A.-H. Tan, C. Miao, J. Jiang, and Y. Zhou, "A novel density peak clustering algorithm based on squared residual error," in *Proceedings of the 2017 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC)*, pp. 43–48, Shenzhen,China, Dec. 2017.

[28] M. D. Parmar, W. Pang, D. Hao et al., "FREDPC: a feasible residual error-based density peak clustering algorithm with the fragment merging strategy," *IEEE Access*, vol. 7, pp. 89789–89804, 2019.

[29] 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, pp. 135–145, 2016.

[30] 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.

[31] B. Shi, L. Han, and H. Yan, "Adaptive clustering algorithm based on kNN and density," *Pattern Recognition Letters*, vol. 104, pp. 37–44, 2018.

[32] M. Xu, Y. Li, R. Li, F. Zou, and X. Gu, "EADP: an extended adaptive density peaks clustering for overlapping community detection in social networks," *Neurocomputing*, vol. 337, pp. 287–302, 2019.

[33] D. Jiang, W. Zang, R. Sun, Z. Wang, and X. Liu, "Adaptive density peaks clustering based on K-nearest neighbor and gini coefficient," *IEEE Access*, vol. 8, pp. 113900–113917, 2020.

[34] J. Sun and G. Liu, "An improvement of density peaks clustering algorithm based on KNN and gravitation," *2021 4th International Conference on Intelligent Autonomous Systems (ICoIAS)*, in *Proceedings of the 2021 4th International Conference on Intelligent Autonomous Systems (ICoIAS)*, pp. 234–239, Wuhan, China, May 2021.

[35] J.-C. Fan, P.-L. Jia, L. Ge, and Mk-Nng-Dpc, "Mk-NNG-DPC: density peaks clustering based on improved mutual

K-nearest-neighbor graph," *International Journal of Machine Learning and Cybernetics*, vol. 11, no. 6, pp. 1179–1195, 2020.

[36] C. L. Ma, H. Shan, and T. Ma, "Improved density peaks based clustering algorithm with strategy choosing cluster center automatically," *Computer Science*, vol. 43, no. 7, pp. 255–258, 2016.

[37] T. Zhao, "An improved density peak clustering algorithm and its application," Master's thesis, Jilin University, Jilin,China, 2019.

[38] T. Du, S. Qu, and Q. Wang, "A data-driven parameter adaptive clustering algorithm based on density peak," *Complexity*, vol. 2018, Article ID 5232543, 14 pages, 2018.

[39] J. C. García-García and R. García-Ródenas, "A methodology for automatic parameter-tuning and center selection in density-peak clustering methods," *Soft Computing*, vol. 25, no. 2, pp. 1543–1561, 2021.

[40] G. Wang, J. Xu, W. Deng, M. Shang, and X. Zhang, "An efficient hierarchical clustering method based on density peak," Chinese Patent No. CN105631465A, 2016.

[41] W. Zhang and J. Li, "Extended fast search clustering algorithm: widely density clusters, no density peaks," in *Proceedings of the Third International Conference on Database and Data Mining (DBDM 2015)*, pp. 1–17, Dubai, UAE, April 2015.

[42] 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.

[43] L. Ding, W. Xu, and Y. Chen, "Improved density peaks clustering based on natural neighbor expanded group," *Complexity*, vol. 2020, Article ID 8864239, 11 pages, 2020.

[44] Q. Diao, Y. Dai, Q. An, W. Li, X. Feng, and F. Pan, "Clustering by detecting density peaks and assigning points by similarity-first search based on weighted K-nearest neighbors graph," *Complexity*, vol. 2020, Article ID 1731075, 17 pages, 2020.

[45] P. Mehrmohammadi, M. Hatami, and P. Moradi, "A density peaks clustering method based on mutual kNN graph and shortest path," in *Proceedings of the 2020 28th Iranian Conference on Electrical Engineering (ICEE)*, pp. 1–6, Tabriz, Iran, Auguest 2020.

[46] F. Fang, L. Qiu, and S. Yuan, "Adaptive core fusion-based density peak clustering for complex data with arbitrary shapes and densities," *Pattern Recognition*, vol. 107, Article ID 107452, 2020.