Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.054775

ARTICLE

Knowledge-Driven Possibilistic Clustering with Automatic Cluster Elimination

Xianghui Hu[1], Yiming Tang[2,3], Witold Pedrycz[3,4], Jiuchuan Jiang[5,*] and Yichuan Jiang[1,*]

[1]The School of Computer Science and Engineering, Southeast University, Nanjing, 211189, China

[2]The School of Computer and Information, Hefei University of Technology, Hefei, 230601, China

[3]The Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6R 2V4, Canada

[4]The Systems Research Institute, Polish Academy of Sciences, Warsaw, 00-901, Poland

[5]The School of Information Engineering, Nanjing University of Finance and Economics, Nanjing, 210023, China

*Corresponding Authors: Jiuchuan Jiang. Email: jcjiang@nufe.edu.cn; Yichuan Jiang. Email: yjiang@seu.edu.cn

Received: 06 June 2024 Accepted: 14 August 2024 Published: 12 September 2024

ABSTRACT

Traditional Fuzzy C-Means (FCM) and Possibilistic C-Means (PCM) clustering algorithms are data-driven, and their objective function minimization process is based on the available numeric data. Recently, knowledge hints have been introduced to form knowledge-driven clustering algorithms, which reveal a data structure that considers not only the relationships between data but also the compatibility with knowledge hints. However, these algorithms cannot produce the optimal number of clusters by the clustering algorithm itself; they require the assistance of evaluation indices. Moreover, knowledge hints are usually used as part of the data structure (directly replacing some clustering centers), which severely limits the flexibility of the algorithm and can lead to knowledge misguidance. To solve this problem, this study designs a new knowledge-driven clustering algorithm called the PCM clustering with High-density Points (HP-PCM), in which domain knowledge is represented in the form of so-called high-density points. First, a new data density calculation function is proposed. The Density Knowledge Points Extraction (DKPE) method is established to filter out high-density points from the dataset to form knowledge hints. Then, these hints are incorporated into the PCM objective function so that the clustering algorithm is guided by high-density points to discover the natural data structure. Finally, the initial number of clusters is set to be greater than the true one based on the number of knowledge hints. Then, the HP-PCM algorithm automatically determines the final number of clusters during the clustering process by considering the cluster elimination mechanism. Through experimental studies, including some comparative analyses, the results highlight the effectiveness of the proposed algorithm, such as the increased success rate in clustering, the ability to determine the optimal cluster number, and the faster convergence speed.

KEYWORDS

Fuzzy C-Means (FCM); possibilistic clustering; optimal number of clusters; knowledge-driven; machine learning; fuzzy logic

1 Introduction

Clustering is an unsupervised machine learning methodology that offers a suite of algorithms to discover a structure in the given unlabeled dataset. Clustering techniques have been widely applied in various fields such as industrial chains [1], collective intelligence [2], and image processing [3,4]. Depending on whether additional knowledge hints are incorporated into the clustering process, clustering algorithms can be classified into data-driven and knowledge-driven. The classical objective function-based K-means [5] and FCM algorithms [6,7], as well as some improved algorithms [8–11] are data-driven clustering algorithms. Data-driven clustering uncovers the potential structure by only considering the relationships present in the unlabeled data, thus ignoring the impact of domain knowledge on clustering. This clustering process lacks a clear search direction, resulting in a slower completion of the clustering task [12,13]. As opposed to data-driven clustering, knowledge-oriented clustering increases the consideration of compatibility with knowledge hints so that the overall optimization process focuses on a specific search direction, thus improving the effectiveness [12–15].

Knowledge-driven clustering simulates the process of differentiating items in reality, that is, purposefully and quickly finding specific objects when provided with extra knowledge hints. For example, in the fuzzy clustering with viewpoints (V-FCM) algorithm proposed by Pedrcyz et al., the knowledge hints (called viewpoints) are extreme data (minimum, maximum, medium, etc.) [12]. Those viewpoints directly replace some of the prototypes during the optimization process as a way to achieve the navigation of a personalized search for the structure. Based on this knowledge-driven clustering framework, Tang et al. proposed possibilistic fuzzy clustering with a high-density viewpoint (DVPFCM) algorithm [13]. The DVPFCM algorithm replaces the personality viewpoints in the V-FCM algorithm with a high-density point to guide the algorithm to find centers closer to the actual data structure. Compared with V-FCM, the DVPFCM algorithm is more robust. To better deal with datasets with irregular structures, Tang et al. proposed the Knowledge-induced Multiple Kernel Fuzzy Clustering (KMKFC) in which the kernel distance is adopted [14]. However, V-FCM, DVPFCM, and KMKFC all output domain knowledge as part of the prototype and change the partition matrix U directly, which means that some clustering results are determined before the algorithm is executed. The flexibility of the algorithm is greatly limited, and the domain knowledge has too much influence.

In addition to these V-FCM-based algorithms, transfer clustering is another type of knowledge-based clustering. The self-taught clustering proposed in [16] which is a transfer clustering strategy based on the strength of mutual information. Jiang et al. [17] introduced the transfer learning idea into spectral clustering and then presented the transfer spectral clustering algorithm. Reference [18] introduced the concept of transfer learning to prototype-based fuzzy clustering and provided two clustering algorithms. The core idea of these algorithms is to extract knowledge from an auxiliary dataset and then improve the learning ability of the target dataset. However, there are not always auxiliary data to help cluster the target data in reality. Moreover, the number of clusters in knowledge-driven clustering based on the idea of V-FCM and transfer clustering is given in advance. However, in reality, the value of this parameter is difficult to obtain directly.

The research progress in knowledge-driven clustering shows that most studies focus on knowledge utilization related to the clustering center location to locate the final clustering centers more accurately or with more personality viewpoints. However, another factor that affects the performance of clustering algorithms, the number of clusters, is ignored. Existing knowledge-driven clustering algorithms are predicated on the premise that this domain knowledge is given in advance, but in reality, the number of clusters is hard to determine, and it has a significant impact on the effectiveness of clustering [19,20]. The number of clusters directly influences the granularity of the grouping and the interpretability of

the results. If the number of clusters is too high, the algorithm might overfit the data, resulting in too many small, fragmented clusters that fail to reveal meaningful patterns. Conversely, if the number of clusters is too low, the algorithm might underfit the data, merging distinct groups and obscuring important differences.

A simple and direct way to determine the number of clusters is to use a cluster validity index (CVI) to measure the clustering performance under different numbers of clusters and then select the number of groups corresponding to the best metric value as the optimal number of clusters. This is also a common approach in the current clustering field to detect the correct number of clusters [20–22]. However, this method has the issues of costly computation and excessive testing of clustering numbers (usually ranging from 2 to the square root of the sample numbers, with an increment of 1), as well as the problem of obtaining different optimal cluster numbers with different CVIs [23]. Rodriguez and Laio proposed a density peak clustering (DPC) algorithm which identifies cluster centers based on two key criteria: high local density and a large distance from other points with higher densities [24]. It does not require a predefine cluster number but allows manual selection based on the decision graph. However, its time complexity is high, at $\mathcal{O}\left(n^2\right)$ where N is the number of samples. Consequently, some methods have been proposed to reduce this time cost, such as fast clustering with local density peaks-based minimum spanning tree (Fast LDP-MST) and the Granular ball-based density peak (GB-DP) algorithm [25,26]. Fast LDP-MST designs a more effective method based on minimum spanning tree to reduce the time complexity of DPC to $\mathcal{O}(N \log(N))$. The GB-DP algorithm uses granular computing techniques to reduce the original dataset into smaller granules, and then applies the DPC algorithm to cluster these granulated datasets, significantly reducing the algorithm's time complexity. Although algorithms based on DPC do not require the number of clusters to be predefined and have reduced time complexity, the DPC algorithm is sensitive to the parameter density radius r. Additionally, DPC is a hard clustering algorithm, exhibiting binary characteristics. This binary nature can make it difficult to handle fuzzy boundaries between data points in some cases.

Facing these challenges, our work in this paper attempts to adaptively determine the optimal number of clusters during the clustering process through knowledge guidance. The knowledge extraction algorithm, DKPE, which is proposed first, obtains knowledge tidbits, i.e., high-density points, that help determine the positions of cluster centers. The initial number of clusters is fixed at twice the number of high-density points, which is obviously greater than the actual number of clusters. Finally, combined with the merging clusters schema, the HP-PCM algorithm eliminates redundant cluster centers based on the influence of high-density points. In this way, the proper number of clusters is determined.

By integrating automatic cluster elimination, our approach not only reduces computational overhead but also enhances the consistency and reliability of clustering results. This ensures that high-quality clustering solutions are achieved across various datasets and application scenarios, thereby improving the overall performance and applicability of knowledge-driven clustering algorithms. Moreover, in [12–14], knowledge tidbits directly replaced part of the family of prototypes, resulting in ineffective iterative updates of some cluster centers, as these centers are predetermined before clustering. In this way, the flexibility of the algorithm is severely limited. Therefore, in the proposed clustering algorithm, HP-PCM, adaptive influence weights for knowledge tidbits and a term to measure the matching degree between knowledge and cluster centers in the objective function are added. Our goal is to change the role of knowledge from a decision-maker to a guide, thus improving clustering performance while adaptively updating the cluster results (centers and membership degrees).

The contributions of our proposed HP-PCM algorithm can be summarized as follows:

- We propose a new knowledge hints extraction method, DKPE, to provide more useful assistance for knowledge-driven clustering. The DKPE method can extract high-density and scattered data as knowledge hints, which are utilized in the HP-PCM algorithm as guides. Its operation is not complicated and does not require additional datasets.
- The number of knowledge points is used as the basis for setting the initial number of clusters C_{ini} to ensure that C_{ini} is greater than the true value and to improve the efficiency of the algorithm in adaptively determining the final number of clusters.
- We establish an improved objective function by balancing the relationship between clustering centers and knowledge hints (high-density points) and incorporating the merging clusters schema. This allows the HP-PCM algorithm to obtain more appropriate results and select parameters with a logical explanation.

The rest of this paper is organized as follows. In Section 2, we review the PCM-type and V-FCM-based algorithms. The motivation of this study is elaborated in Section 3. In Section 4, we present our proposed algorithms and their frameworks. The experimental results on synthetic datasets and the UCI datasets are described in Section 5. Finally, Section 6 presents our conclusions.

2 Related Work

In this section, PCM-type algorithms and V-FCM-based algorithms are briefly reviewed, and their main features are discussed. Consider the dataset $X = \{x_j\}_{j=1}^{N}$ composed of N samples. The goal of these algorithms is to divide the data into C subsets to form the cluster center set $V = \{v_i\}_{i=1}^{C}$. Each sample x_j and cluster center v_i belong to $\mathbb{R}^l$ where l is the space dimensionality. The distance measure adopted in the algorithms is the Euclidean norm, denoted by $\|\cdot\|$. The iteration number of the algorithm is t, and the maximum iteration number is $tMax$.

2.1 PCM-Type Algorithms

The PCM algorithm is an improved FCM algorithm. It addresses the FCM algorithm's sensitivity to noise by removing the sum-to-one constraint on membership degrees. In PCM, the compatibility of a point with a given cluster depends solely on the distance of the point from the corresponding cluster center. This approach eliminates the need to consider the point's relationship with other clusters, thereby reducing the impact of outliers and noise on the updates to cluster centers.

PCM does not impose the sum-to-one limit, but a serious problem of the consistency of cluster centers may arise [27,28]. Improper γ values may lead PCM to fail to identify a sparse cluster located very close to a denser cluster, or it may even lead to a cluster center consistency problem [10,28]. To alleviate the shortcomings of PCM, scholars have proposed some improvements. The methods in [29–32] overcome the PCM algorithm cluster center consistency problem to a certain extent. Nevertheless, they cannot eliminate redundant classes if the number of initial clusters exceeds the actual value. Therefore, based on the idea of merging clusters [33], the PCM-type algorithms proposed in [10–11,34,35] have the cluster elimination abilities during their execution. Especially for the adaptive possibilistic clustering (APCM) and sparse-aware adaptive possibilistic clustering (SAPCM) algorithms proposed in [10,11], their update method of γ only considers the data points most relevant to the current center to enhance the flexibility and clustering ability of the algorithm.

However, the initial number of clusters in the abovementioned algorithms is set manually and lacks a reasonable explanation. More importantly, this setting directly affects the clustering performance, particularly in determining the correct number of final clusters [10,11].

2.2 V-FCM-Based Algorithms

The V-FCM algorithm proposed by Pedrycz et al. introduced the concept of "viewpoints" based on the FCM algorithm [12]. A viewpoint is the domain knowledge provided by a user, and it involves navigation for a personalized search of the structure. Viewpoints can be typical situations in data, such as the average, maximum, and minimum. Therefore, the V-FCM algorithm makes the clustering process more accessible with the help of the viewpoints, and the clustering results are more interpretable.

The viewpoints are defined by the two matrixes, denoted as B and F:

$$\begin{aligned} b_{ik} &= \begin{cases} 1, & \text{if the } k\text{th feature of the } i\text{th propotype is from the viewpoint} \\ 0, & \text{otherwise} \end{cases} \\ f_{ik} &= \begin{cases} y, & b_{ik} = 1 \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{1}$$

where y represents a specific value of the viewpoint. The V-FCM algorithm introduces knowledge hints, referred to as "viewpoints", which result in a different updating equation for cluster centers compared to the FCM algorithm. As seen from (2), during the iteration process, the viewpoint f_{ik} remains fixed at the initial position of the prototype, and the parts not replaced by viewpoints need to be updated. This simplification of the clustering process allows for direct determination of clustering results, but it also reduces the flexibility of the V-FCM algorithm. Compared to the FCM algorithm, V-FCM can generate more reasonable and explainable clustering results, especially in extreme cases.

$$g_{ik} = \begin{cases} \dfrac{\sum_{j=1}^{N} u_{ij}^{m} x_{jk}}{\sum_{j=1}^{N} u_{ij}^{m}}, & b_{ik} = 0 \\ f_{ik}, & b_{ik} = 1 \end{cases} \tag{2}$$

The DVPFCM and KMKF algorithms, which are improved versions of the V-FCM algorithms, improve the algorithm in terms of the knowledge type and distance formula, respectively, to obtain more accurate clustering results and increase its applicability. However, they all ignore the quality of knowledge and the unknown number of clusters, which can affect clustering performance.

3 Motivation

After describing the main characteristics of the PCM-type algorithms and the V-FCM-based algorithms, and based on the challenges mentioned in Section 1, the tasks of HP-PCM can be summarized as follows:

1. to set the proper initial number of clusters (>true number);
2. to adaptively obtain the optimal number of clusters;
3. to balance the relationships of knowledge hints and cluster centers.

The first task is the basis for the second one. Most existing studies on the determination of the optimal cluster number set the initial value through human intervention, which may cause subjective

bias. In this study, the first task is performed by proposing a knowledge extraction method, DKPE. This method provides P data points with a relatively high density and scattering distribution. P represents the approximate number of clusters of a dataset and can be used as a reference for setting C_{ini} to be twice of P.

For the second task, we adopt the idea of merging clusters in APCM, but with an automatically determined suitable initial number of clusters. During each iteration, γ_is are updated by considering only the points that are most compatible with the corresponding v_i. This allows the HP-PCM algorithm to discard invalid clusters and obtain the optimal clusters.

The final task is accomplished by adding a term to the objective function that balances the relationship between cluster centers and high-density points identified by the DKPE method. The influence of high-density points on the algorithm adapts during the iterative process, leading to cluster centers that optimize the objective function. The overall road map of HP-PCM is summarized in Fig. 1.

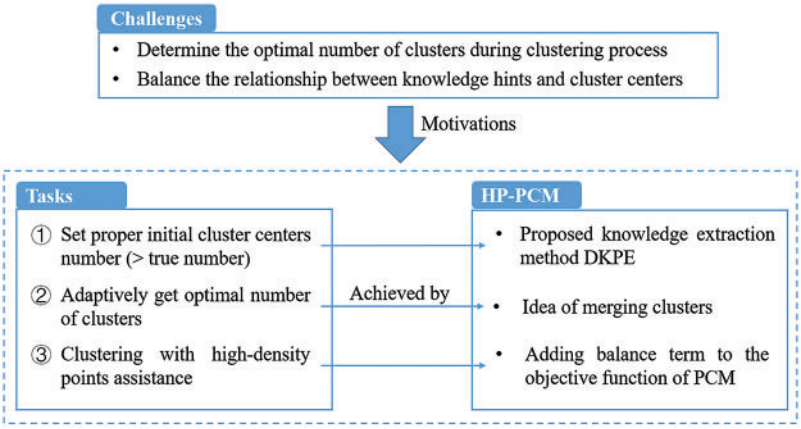

Figure 1: Road map of the proposed HP-PCM algorithm

4 Possibilistic C-Means Clustering with High-Density Points

This section establishes a knowledge extraction method, DKPE, and a novel clustering algorithm called the PCM clustering algorithm with high-density points (HP-PCM). The DKPE algorithm is designed to provide the HP-PCM algorithm with an appropriate number of initial clusters and high-density points (knowledge hints). Utilizing these knowledge hints, HP-PCM can automatically determine the optimal number of clusters during its clustering process.

4.1 Knowledge Extraction Method Based on Data Density

We propose the density knowledge points extraction (DKPE) method to select high-density points from the dataset as knowledge hints for the proposed HP-PCM clustering algorithm. The idea of DKPE is inspired by a clustering algorithm named clustering by a fast search and find of density peaks (DPC for short) [24].

The DPC algorithm involves calculating the local density and the relative distance between the data and data with higher local densities. This approach is straightforward to implement and ensures that the selected centers are well-dispersed. Similarly, the proposed DKPE method aims to extract high-density and scattered points as knowledge hints for the clustering algorithm. Although these two

ideas are similar, the DPC method faces challenges in evaluating its density radius r. This parameter r determines the accuracy of DPC in estimating the density of data points, which in turn affects whether the algorithm can obtain true high-density points. In [24], the value of r is selected experimentally and adopts the first 1% to 2% of the ascending data distance. Furthermore, experiments in the next section show that the DPC centers are different from r in this interval for the same dataset.

The DPC algorithm is sensitive to the radius r but can get some high-density points. How can the performance of DPC be improved so that high-density points can be easily distinguished and extracted? To answer this question, we develop a novel density knowledge extraction method. We redefine the calculation of the local density of a sample x_j as follows:

$$\rho_j = \sum_{x \in KNN(x_j)} d\left(x_j, x_k\right), d\left(x_j, x_k\right) = \| x_j - x_k \| . \tag{3}$$

where $KNN(x_j)$ represents the K-nearest neighbors of x_j, $K = \left\lfloor \sqrt{N} \right\rfloor$. The local density of a data point x_j is the sum of its distances from the nearest K data points. The larger the value of ρ_j is, the farther the point x_j deviates from other data points. In other words, the more scattered the data distribution around x_j, the smaller its local density. ρ_j is negatively correlated with the density concept in [24]. Eq. (3) does not introduce other parameters but focuses on the value of K. It is assumed that the clustering problem is reasonable (i.e., $N \gg C$) and a suitable upper bound of clusters number could be set to $\sqrt{N}$ in [36]. That is, the crowded areas of a dataset do not exceed$\sqrt{N}$. For the limit idea, assuming N points of a dataset are evenly distributed in $\sqrt{N}$ crowded areas, there are $\sqrt{N}$ data points around each area. Additionally, those works using the concept of KNN also set K as $\sqrt{N}$ to make their algorithms have better performance [37–39]. That is, $K = \left\lfloor \sqrt{N} \right\rfloor$ is also a commonly used rule. The local density of a data point in the center of a dense area is the highest, so the clustering center point close to the real data structure can be best screened by calculating the distance sum of the K other data points closest to the data point.

The search cost of KNN for each data point in the dataset is $\mathcal{O}(N)$, so the time complexity of DKPE is $\mathcal{O}(N^2)$, which is huge. We introduce the KD-tree [40] into the DKPE algorithm to reduce the time complexity of DKPE to be $\mathcal{O}(N \log N)$.

If the clustering centers are determined by the local density values, it is possible to select some centers in the same dense area because their densities are estimated by the same K number of neighbors. In that case, their densities are also the same. There is a requirement to calculate another parameter $\delta_j (j \in \{1, \ldots, N\})$. Its formula is expressed as:

$$\delta_j = \min \left\{ d\left(x_k, x_j\right) | \rho_k < \rho_j, k \in \{1, 2, \ldots, N\} \right\} . \tag{4}$$

For the point with the smallest local density, $\delta_j = \max \left\{ d\left(x_k, x_j\right) | j \neq k, k \in \{1, \ldots, N\} \right\}$ $(j \in \{1, \ldots, N\})$.

The data points selected by the DKPE algorithm have smaller ρ and larger δ values. So, they conform to the natural clustering centers characteristics. The DKPE algorithm calculates the local density of a data point based on the K nearest neighbors. Data points with a higher density can be generated semi-automatically, and are unaffected by the density radius r. The specific execution is shown in Algorithm 1.

Algorithm 1: $[G, C_{ini}] = \mathrm{DKPE}(X)$

1: Set $K = \left\lceil \sqrt{N} \right\rceil$;
2: **for** $j = 1$ to N **do**
3: Use KD-tree to search $KNN(x_j)$;
4: Calculate ρ_j using (3);
5: Calculate δ_j using (4);
6: **end**
7: Draw decision figure $\rho - \delta$;
8: Manually select high-density points $G = [g_k]_{k=1,\ldots,P}$ and $C_{ini} = 2P$;
9: Return G, C_{ini}

4.2 Superiority of the Parameter K

Here, we illustrate the advantage of using the K-nearest neighbors to measure the data local density compared to measuring it with radius r in DPC. The used testing data is a 2-D synthetic dataset that is composed of three Gaussian clusters with centers $v_1 = [0.0578, 15.0039]$, $v_2 = [-5.5063, 12.5743]$, and $v_3 = [4.7078, 13.5072]$, and covariance matrices $\sum_1 = \begin{bmatrix} 1.9745 & -0.0825 \\ -0.0825 & 0.5087 \end{bmatrix}$, $\sum_2 = \begin{bmatrix} 0.0656 & -0.0014 \\ -0.0014 & 0.0711 \end{bmatrix}$, $\sum_3 = \begin{bmatrix} 0.0540 & 0 \\ 0 & 0.0503 \end{bmatrix}$. The three clusters are composed of 1000, 50, and 100 data points, respectively, making the sample size be $N = 1150$. The data points belonging to the first, second, and third clusters are numbered 1 to 1000, 1001 to 1050, and 1051 to 1150, respectively. The data are shown in Fig. 2a.

Comparing Fig. 2c,d, it can be found that setting r to the first 2% of the ascending distance provides the decision graph of DPC with a clearer gap between high-density points and non-high-density points than setting r to the first 1% of the distance. This means that the evaluation of local data density is sensitive to the value of r in DPC. In addition, the high-density points in the smaller cluster may be ignored or mistakenly identified as noise, leading to their exclusion—such as the black dot within the orange boxes in Fig. 2c,d. However, the data points chosen in the orange boxes of these two figures are scattered in different clusters, whereas the data points in the red boxes are concentrated within the same cluster.

In Fig. 2b, the high-density points are clearly located away from the non-high-density points, which makes the manual selection operation easy to perform and free of doubt. To show that the proposed DKPE algorithm is less sensitive to the value of K, we calculate the average distance $\hat{d}$ between the high-density points and the reference cluster centers obtained by DKPE and DPC for different values of K and r, respectively. K is varied in the interval [13,43] by an increase of 10. A smaller $\hat{d}$ indicates that the high-density points are closer to the dense areas. The formula of parameter $\hat{d}$ is:

$$\hat{d} = \frac{\sum_{i=1}^{P} d(g_i, v_i)}{P} \tag{5}$$

The calculation results are summarized in Table 1. The $\hat{d}$ value obtained by DKPE with different K values is approximately 0.1340, while this index exceeds 3.12 in DPC. This shows that DKPE provides a more reasonable measure of local data density based on K.

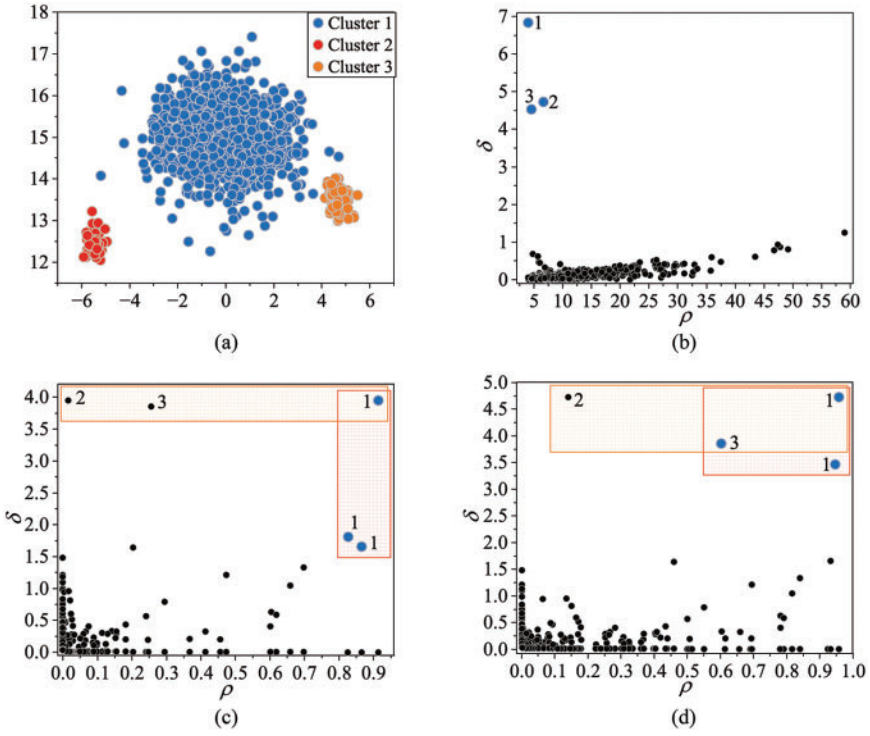

Figure 2: Synthetic dataset D1 and decision graphs of DPC and DKPE on D1. (a) Data distribution. Different colors represent different clusters. (b) DKPE decision graph with $K = \lfloor \sqrt{N} \rfloor = 33$. The numbers next to the dots are the cluster numbers to which the corresponding data point belongs. (c) Parameter r is the first 1% of the ascending data distance. (d) Parameter r is the first 2% of the ascending data distance

Table 1: Results on the dataset D1 for DKPE and DPC

Method	DKPE			
K	13	23	33	43
$\hat{d}$	0.1508	0.1525	0.1340	0.1340
ID	674, 1033, 1147	596, 1044, 1147	417, 1044, 1136	417, 1044, 1136
CN	1, 2, 3	1, 2, 3	1, 2, 3	1, 2, 3
Method	DPC			
r(%)	1.0	1.5	2.0	2.5
$\hat{d}$	3.1204	3.1204	3.2796	3.2796
ID	8, 188, 551	8, 188, 551	188, 551, 1065	188, 551, 1065
CN	1, 1, 1	1, 1, 1	1, 1, 2	1, 1, 2

Note: ID is the selected data number; CN is the cluster label of the selected data.

Changes in r can lead to variations in the number of data points surrounding the observed data point. In contrast, altering the value of K does not affect the average spacing between points. Within a certain range, the distribution of data points is relatively uniform. Therefore, DKPE can obtain high-density points more accurately and consistently.

4.3 Description of the Proposed HP-PCM Algorithm

Based on the proposed DKPE algorithm, we present the PCM clustering algorithm with high-density points (HP-PCM), and the goals of the algorithm are as follows:

1. To balance the relationship between the high-density points identified by the DKPE method and the clustering centers, thereby preventing the high-density points from exerting excessive influence;
2. To adaptively determine the optimal number of clusters with the guidance of knowledge points and the cluster merging mechanism.

Moreover, the proposed HP-PCM algorithm is improved from the PCM algorithm. Therefore, its objective function includes at least three parts: 1) the minimization of the distances between data points and clustering centers; 2) the relationship between clustering centers and high-density points; and 3) the control of u_{ij} to eliminate trivial solutions. The objective function of HP-PCM can be expressed as follows:

$$J_{\mathrm{HP-PCM}}^{(t)} = \sum_{i=1}^{C^{(t)}}\sum_{j=1}^{N} \| x_j - v_i \|^2 + \sum_{i=1}^{C^{(t)}}\sum_{i=1}^{P} \varphi_{ik} \| g_k - v_i \|^2 + \sum_{i=1}^{C^{(t)}} \eta_i \frac{\hat{\eta}}{\alpha} \sum_{j=1}^{N} (u_{ij} \ln u_{ij} - u_{ij}) + \sum_{i=1}^{C(t)} \sigma_i \sum_{k=1}^{P} \varphi_{ik} \ln \varphi_{ik}$$
$$s.t. \sum_{k=1}^{P} \varphi_{ik} = 1, 0 \le \varphi_{ik} \le 1. \tag{6}$$

Here, $C^{(t)}$ is the clusters number in the tth iteration. The term $\eta_i \frac{\hat{\eta}}{\alpha}$ is a redefinition of γ_i in the PCM algorithm, and $\hat{\eta} = \min(\eta_i)$, $(i = 1, \ldots, C^{(t)})$. α is assumed to be a predefined positive value. In the initial setting stage, $C^{(0)}$ is set to a greater value than the real number of clusters C. The FCM algorithm generates $C^{(0)}$ initial clustering centers $v_i^{(FCM)}$ and the corresponding $u_{ij}^{(FCM)}$ where $i = 1, \ldots, C^{(0)}$. Then, we initialize η_i as follows:

$$\eta_i = \frac{\sum_{j=1}^{N} u_{ij}^{(\mathrm{FCM})} \| x_j - v_i \|}{\sum_{j=1}^{N} u_{ij}^{(\mathrm{FCM})}}, i = 1, \ldots, C^{(0)}. \tag{7}$$

$G = [g_k]_{k=1,\ldots,P}$ in (6) refers to the P high-density points obtained by DKPE. The proposed HP-PCM algorithm strives to obtain scattered clustering centers in dense areas. Therefore, the second term of (6) serves as the knowledge guidance function of high-density points. φ_{ik} measures the influence of the high-density point g_k on the clustering center v_i, and there is a constraint $\sum_{k=1}^{P} \varphi_{ik} = 1$. The fourth term of Eq. (6) is to prevent a valueless solution for φ_{ik}. The specific meaning of σ_i will be explained later, and its calculation is carried out as

$$\sigma_i = \| v_i - \overline{G} \|^2, \overline{G} = \frac{\sum_{k=1}^{P} g_k}{P}. \tag{8}$$

According to the optimization that invokes the Lagrange multipliers, the updated formula of $u_{ij}^{(t)}$, $v_i^{(t+1)}$, and $\varphi_{ik}^{(t)}$ are as follows:

$$u_{ij}^{(t)} = \exp\left(\frac{-\alpha \parallel x_j - v_i^{(t-1)} \parallel^2}{\eta_i^{(t-1)}\hat{\eta}}\right), \tag{9}$$

$$v_i^{(t+1)} = \frac{\sum_{j=1}^{N} u_{ij}^{(t)} x_j + \sum_{k=1}^{P} \varphi_{ik}^{(t)} g_k}{\sum_{j=1}^{N} u_{ij}^{(t)} + \sum_{k=1}^{P} \varphi_{ik}^{(t)}}, \tag{10}$$

$$\varphi_{ik}^{(t)} = \frac{\exp\left(\frac{- \parallel g_k - v_i^{(t-1)} \parallel^2}{\sigma_i}\right)}{\sum_{i=1}^{(t-1)} \exp\left(\frac{- \parallel g_k - v_i^{(t-1)} \parallel^2}{\sigma_i}\right)}. \tag{11}$$

As seen in (11), on the one hand, if ($\parallel g_k - v_i \parallel^2$) of the term $\exp(- \parallel g_k - v_i \parallel^2 / \sigma_i)$ is too large, then the numerator $\exp(- \parallel g_k - v_i \parallel^2)$ will become too small as it approaches to zero. Then, we need to avoid this situation to ensure that as many high-density points are used for guidance during this updating step as possible. On the other hand, if the term ($\parallel g_k - v_i \parallel^2$) is too small, then the numerator $\exp(- \parallel g_k - v_i \parallel^2)$ will approach to one, if this happens, too many high-density points will be used for guidance. We also need to avoid this situation. Therefore, we need to have a suitable variable to control this term in every iteration step. As it is ($\parallel g_k - v_i \parallel^2$) that causes $\exp(- \parallel g_k - v_i \parallel^2)$ to be close to zero or one, we can use the σ_i in (8), which is the mean distance between the current clustering centers and high-density points, to control the value of ($\parallel g_k - v_i \parallel^2$). In this way, the numerator in (11) can be set to be a valid value.

Notable, the update of η_i is:

$$\eta_i^{(t+1)} = \frac{1}{n_i^{(t)}} \sum_{x_j : Z} \parallel x_j - \mu_i^{(t)} \parallel,$$
$$Z = \max_{c=1,\ldots C^{(t+1)}} u_{cj}^{(t)}, \tag{12}$$

where $n_i^{(t)}$ is the number of data points x_j that are most compatible with cluster C_i at iteration t and $\mu_i^{(t)}$ denotes the mean vector of these data points. That is, when ηs are updated, it does not consider all the data points that are weighted by their corresponding u_{ij} coefficients but only considers the data points that are most compatible with cluster C_i. This characteristic is an essential condition for the subsequent cluster elimination, since the approach allows parameter η_i to approach zero. Thus, it eliminates the corresponding cluster C_i whereas η_i would always remain positive when all data points are taken into account. Concerning the adjustment of the number of clusters $C^{(t)}$ at the tth iteration, we adopt the corresponding operation of the APCM algorithm [10]. Let *label* be an N-dimensional vector. *label*(j) stores the index of the cluster that is most compatible with x_j. After u_{ij}s are updated, invalid clusters must be removed. If the index i of cluster V_i appears at least once in the vector *label*, V_i is preserved; otherwise, it is removed. Then, $U$and V are updated accordingly. As a result, the updated number of clusters $C^{(t)}$ will be reduced (see the possible cluster elimination part in Algorithm 2).

In summary, we form the whole mechanism of HP-PCM, which is a novel PCM algorithm driven by both data and knowledge. Moreover, the HP-PCM algorithm can adaptively obtain the optimal cluster number. Its execution framework is shown in Algorithm 2, and the whole idea is shown in Fig. 3.

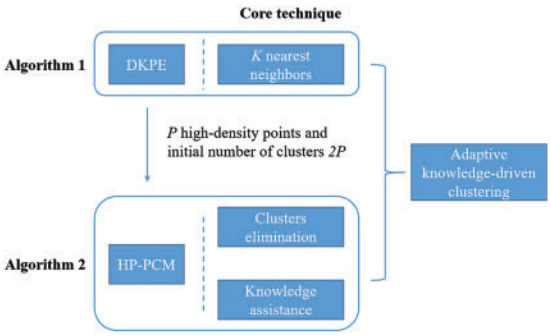

Figure 3: Overall idea of HP-PCM

Algorithm 2: $[V, U, C_{final}] = \text{HP} - \text{PCM}(X, \alpha, \varepsilon)$

Set $t = 0$;
$[G, C_{ini}] = \text{DKPE}(X)$;
Set $C^{(0)} = C_{ini}$;
$[V^{(t)}, U^{(\text{FCM})}] = \text{FCM}(X, C^{(0)}, m = 2.0)$;
Calculate $\eta^{(t)} = [\eta_i]_{i=1,\ldots,C^{(0)}}$ using (7);
Repeat
    Calculate $\varphi^{(t)}$s by $V^{(t)}$ using (11);
    Calculate $U^{(t)}$ by $\eta^{(t)}$ and $V^{(t)}$ using (9);
    Calculate $V^{(t+1)}$ by $U^{(t)}$ using (10);
    $\Rightarrow$ Possible cluster elimination
    For each $j \in [1, N]$ do
        $u_{cj}^{(t)} = max_{i=1,\ldots,C^{(t)}} u_{ij}^{(t)}$;
        $label\,(j) = c$;
    End for
    $number = 0$; //number of removed clusters
    For each $i \in [1, C^{(t)}]$ **do**
        If $i \notin label$ then
            Remove v_i and renumber accordingly V and the columns of U;
            $number = number + 1$;
        End if
    End for
    $C^{(t+1)} = C^{(t)} - number$;
    $\Rightarrow$ Adaptation of η
    Update $\eta^{(t+1)}$ using (12);
    Set $t = t + 1$;
Until $\| V^{(t)} - V^{(t-1)} \| < \varepsilon$;
$C_{final} = C^{(t)}$;
Return $V^{(t)}$, $U^{(t-1)}$, C_{final};

High-density points are the essence of the proposed HP-PCM algorithm because they guide the clustering process to more accurately uncover the latent structure of the dataset. The proposed DKPE algorithm introduced in Section 4.1 can obtain these high-density points. Its specific execution is

shown in Algorithm 1. After obtaining high-density points, the HP-PCM algorithm is applied for dataset clustering. The initialization of C_{ini} is another key step in the HP-PCM algorithm. Its value should be larger than the actual cluster numbers. In the experiment in Section 5, we show that the number of high-density points got by the DKPE algorithm closely approximates the actual number of clusters. Therefore, we set C_{ini} to be twice the number of high-density points. See Algorithm 2 for other details.

4.4 Computational Complexity

Given the input dataset with N samples and T iterations, there are three main parts for the computational complexity of HP-PCM. The first part is the knowledge extraction method, DKPE. Its main computational cost is searching the K-nearest neighbors of each sample based on the KD-tree structure. Therefore, the time complexity of DKPE is $\mathcal{O}(N \log(N))$. The second part involves the FCM algorithm, which obtains the initial cluster centers and membership degrees. The computational complexity of FCM is $\mathcal{O}(NT_{\mathrm{FCM}}C_{ini})$, where T_{FCM} is the iteration number of FCM. The third major part of HP-PCM is the T iterations to update the cluster centers and the membership degree matrix. The computational complexity of this part is $\mathcal{O}(NTC_{ini})$. Thus, the overall cost of HP-PCM is $\mathcal{O}(N \log(N) + NT_{\mathrm{FCM}}C_{ini} + NTC_{ini})$.

5 Experiments

To verify the performance of our proposed DKPE algorithm for extracting density knowledge points and the clustering performance of the proposed HP-PCM algorithm, we conduct a series of experiments and analyze the corresponding results.

5.1 Used Datasets and Comparative Algorithms

All the information about the datasets used for the experiments, including the total number of samples and the actual number of clusters, is summarized in Table 2. The clustering results for the two-dimensional synthetic datasets visually show the effectiveness of the DKPE and HP-PCM algorithms. Datasets D6 to D13 are mainly used to explore the effects of data dimensionality, structure, size, and other elements on the clustering performance of the proposed HP-PCM algorithm. The UCI datasets (D15 to D19) are used to compare the proposed HP-PCM algorithm with 10 state-of-the-art algorithms across different types, including 1) 3 traditional fuzzy clustering algorithms: FCM, PCM, and PFCM; 2) 2 knowledge-driven fuzzy clustering algorithms: V-FCM and DVPFCM; 3) a density-based clustering algorithm: DPC; 4) 2 advanced possibilistic algorithms: APCM and C-PCM; and 5) 2 clustering algorithms based on performance indices: Silhouette [41] and fRisk4-bA [20]. Among these comparative algorithms, DPC, APCM, Silhouette, and fRisk4-bA are not only used to compare clustering performance but also to evaluate the accuracy and efficiency in determining the number of clusters. The other algorithms are primarily used to compare the performance of clustering models in terms of evaluation metrics, convergence speed, time cost, and adaptability to different datasets. All the used algorithms are implemented by MATLAB R2019b.

5.2 Evaluation Indices

To evaluate clustering results of algorithms, we employ 1) the success rate (SR), which measures the percentage of patterns that are correctly classified, 2) the Normalized Mutual Information (NMI), which is an asymmetric measure to quantify the statistical information shared between two cluster distributions [42], 3) the Adjusted Rand Index (ARI), which evaluates on a pairwise-basis, 4) the

Centroid Index (CI), which counts the number of clusters that miss a centroid, or have too many centroids, 5) the Xie-Beni index (XB), which is popular in measuring fuzzy clustering performance [43]. These five indices allow the number of clusters obtained by algorithms to be inconsistent with the reference number. They are described in detail as follows:

Table 2: Summary of the experimental datasets

No.	Name	Samples	Features	Classes
D1	A1	3000	2	20
D2	A2	5250	2	35
D3	A3	7500	2	50
D4	Aggregation	788	2	7
D5	Data6	5918	2	9
D6	S1	5000	2	15
D7	S2	5000	2	15
D8	S3	5000	2	15
D9	S4	5000	2	15
D10	Dim032	1024	32	16
D11	Unbalance	6500	2	8
D12	Birch1	100,000	2	100
D13	Birch2	100,000	2	100
D14	Iris	150	4	3
D15	Wine	178	13	3
D16	Glass	213	9	6
D17	Zoo	569	17	7
D18	Image	2100	19	7
D19	Letter	3096	16	4

1) The Success Rate (SR) measures the percentage of patterns that are correctly classified. This index is calculated as follows:

$$SR^{(+)} = \frac{\sum_{i=1}^{C} d_i}{N} \tag{13}$$

where d_i is the number of objects correctly identified in the ith cluster, and N is the number of all objects in the dataset.

2) The Normalized Mutual Information (NMI) is an asymmetric measure to quantify the statistical information shared between two cluster distributions [42]. The larger such index is, the more similar the two class distribution are and the better the clustering performance is. NMI is calculated as follows:

$$NMI^{(+)} = \frac{\sum_{i=1}^{I}\sum_{j=1}^{J} P(i,j)\log\frac{P(i,j)}{P(i)\,P(j)}}{\sqrt{(H(R)\,H(Q))}} \tag{14}$$

where R and Q are two partitions of the dataset. Assume that R, Q have I, J clusters, respectively. $P(i)$ denotes the probability of an object belonging to cluster R_i with R partition. More specifically, $P(i) = \frac{|R_i|}{N}$ where $|R_i|$ is the number of samples in cluster R_i. $P(i,j)$ denotes the probability that an object belongs to cluster R_i with R partition and cluster Q_j with Q partition. $P(i,j)$ is calculated as $P(i,j) = \frac{|R_i \cap Q_j|}{N}$. $H(R)$ is the entropy associated with all probabilities $P(i)$ in partition R, and $H(R) = -\sum_{i=1}^{I} P(i) \log P(i)$. The definition of $H(Q)$ is similar to this.

3) The Adjusted Rand Index (ARI), which evaluates on a pairwise-basis. Assuming that R and Q are two hard partitions, here we first introduce some definitions:

a is the number of data point pairs that belong to the same class with R partition and to the same cluster in Q partition simultaneously;

b is the number of data point pairs that belong to the same class in R partition and to different clusters in Q partition simultaneously;

c is the number of data point pairs that belong to different clusters in R partition and to the same cluster in Q partition simultaneously;

d is the number of data point pairs that belong to different clusters in R partition and to different clusters in Q partition simultaneously.

The ARI is defined as follows:

$$ARI^{(+)} = \frac{a - \frac{(a+b)(a+c)}{d}}{\frac{(a+b)+(a+c)}{2} - \frac{(a+b)(a+c)}{d}}. \tag{15}$$

4) The Centroid Index (CI) counts the number of clusters that miss a centroid, or have too many centroids. Given two sets of prototypes $C = \{c_1, c_2, \ldots, c_{k1}\}$ and $C = \{c_1, c_2, \ldots, c_{k2}\}$, the nearest neighbor mappings $(C \to C')$ are constructed as follows:

$$q_i \leftarrow \underset{1 \le j \le k2}{\text{argmin}} \| C_i - C' \|^2, \forall i \in [1, k1]. \tag{16}$$

Here, we use orphan $(c_j^{'})$ to denote whether it has a prototype that map to the target prototype $c_j^{'}$:

$$\text{orphan}\left(c_j^{'}\right) = \begin{cases} 1, & q_i \neq j, \forall i \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

The dissimilarity of C in respect to C' is the number of orphan prototypes:

$$CI_1(C, C') = \sum_{j=1}^{k2} \text{orphan}\left(C_j^{'}\right). \tag{18}$$

The mapping is not symmetric, i.e., $C \to C' \neq C' \to C$, so CI is defined as:

$$CI^{(-)} = \max\{CI_1(C, C'), CI_1(C', C)\}. \tag{19}$$

5) The Xie-Beni index (XB) is popular in measuring fuzzy clustering performance [43]. The XB index focuses on compactness and separation, where compactness is a measure of proximity between data points within a cluster and separation is a measure of the distance between one cluster and another

cluster. The XB index is defined as follows:

$$XB^{(-)} = \frac{\sum_{i=1}^{C}\sum_{j=1}^{N} u_{ij}^{m} d(x_j, v_i)}{N \times \min d(v_i, v_k)}. \tag{20}$$

5.3 Test for DKPE

The DPC algorithm also has the capability to acquire high-density points, but it struggles with the challenge of determining the parameter r, as discussed earlier in Section 4.2. The quality of the obtained high-density points depends heavily on the value of r. Therefore, we propose the high-density point extraction method, DKPE, to overcome the parameter sensitivity. To compare the performance of DKPE and DPC in identifying high-density points, we use the A1–A3 datasets from [44]. The A1–A3 datasets are 2-dimensional datasets with an increasing number of clusters (A1: 20, A2: 35, A3: 50), decreasing distances between clusters, and gradually increasing total sample size (A1: 3000, A2: 5250, A3: 7500), which also indicates increasing clustering difficulty.

Fig. 4 shows the experimental results of the DKPE and DPC algorithms on the A1, A2, and A3 datasets. The red dots in Fig. 4d–f mark the high-density points obtained by the proposed DKPE algorithm, and the blue dots of the high-density points distribution graphs in Fig. 4g–i mark the high-density points obtained by the DPC algorithm.

As shown in Fig. 4d–f, the DKPE algorithm highlights the high-density points, creating a clear distinction between these and non-high-density points (represented as black dots in the decision graphs). This clear separation makes it easy for us to extract the high-density points shown in red. In comparison, there is no obvious boundary between the high-density and the non-high-density points for DPC. Therefore, it is difficult for us to distinguish between high-density and non-high-density points. Focusing on the results shown in Fig. 4a–c, we can see that the number of high-density points obtained by the proposed DKPE algorithm equals the reference number. The extracted high-density points are located in dense district centers. Furthermore, the number of cluster centers obtained by the DPC algorithm is fewer than the reference number of clusters, with some selected cluster centers situated between dense regions, leading to an underestimation of the actual number of clusters. This result shows that the proposed DKPE algorithm can be better applied to datasets with multiple dense regions than the DPC algorithm.

5.4 Parameter Analyses

Next, we investigate the effect of parameter α on the proposed HP-PCM algorithm and its value selection. The datasets A1–A3, Aggregation [45] and Data6 in [46] are used as experimental objects. From Fig. 5a, we can see that Aggregation contains 7 clusters but the size of each cluster is different, so it causes the larger clusters to have more than one high-density point, and the total number of high-density points is 10, lager than the reference number of clusters, 7. That means density-based clustering algorithms may fail in obtaining the optimal number of clusters for an unbalanced dataset. Data6 consists of 918 data points and 5000 noise points. Its reference number of clusters is 9. As shown in Fig. 5b, the number of noises in Data6 is much more than the data points, and the data-intensive areas are not prominent. This data distribution feature is a great challenge for the HP-PCM algorithm to cluster.

Fig. 6 shows the influence of the parameter α on the difference between the true number of clusters and the final number of clusters obtained by HP-PCM on each dataset when it changes in the interval [0.1,2]. The difference $C_{diff} = C_{true} - C_{final}$. $C_{diff} = 0$ means HP-PCM gets the correct number of clusters. If α takes a value between 0.1 and 0.9, the HP-PCM algorithm is less likely to obtain the ideal C_{final}.

When α varies in [1.4,2], the HP-PCM algorithm cannot correctly get the optimal clusters for the A3 dataset. With further comparison, we can find that when the value of α is 1 to 1.3, the proposed HP-PCM algorithm gets all five datasets the correct optimal clusters numbers as $C_{diff} = 0$. So, we set α to 1.2 in the previous and next experiments. In addition, the difference in the number of clusters obtained by the HP-PCM algorithm is small throughout the variation, which indicates that the proposed HP-PCM algorithm has a good and stable performance in determining the optimal number of clusters.

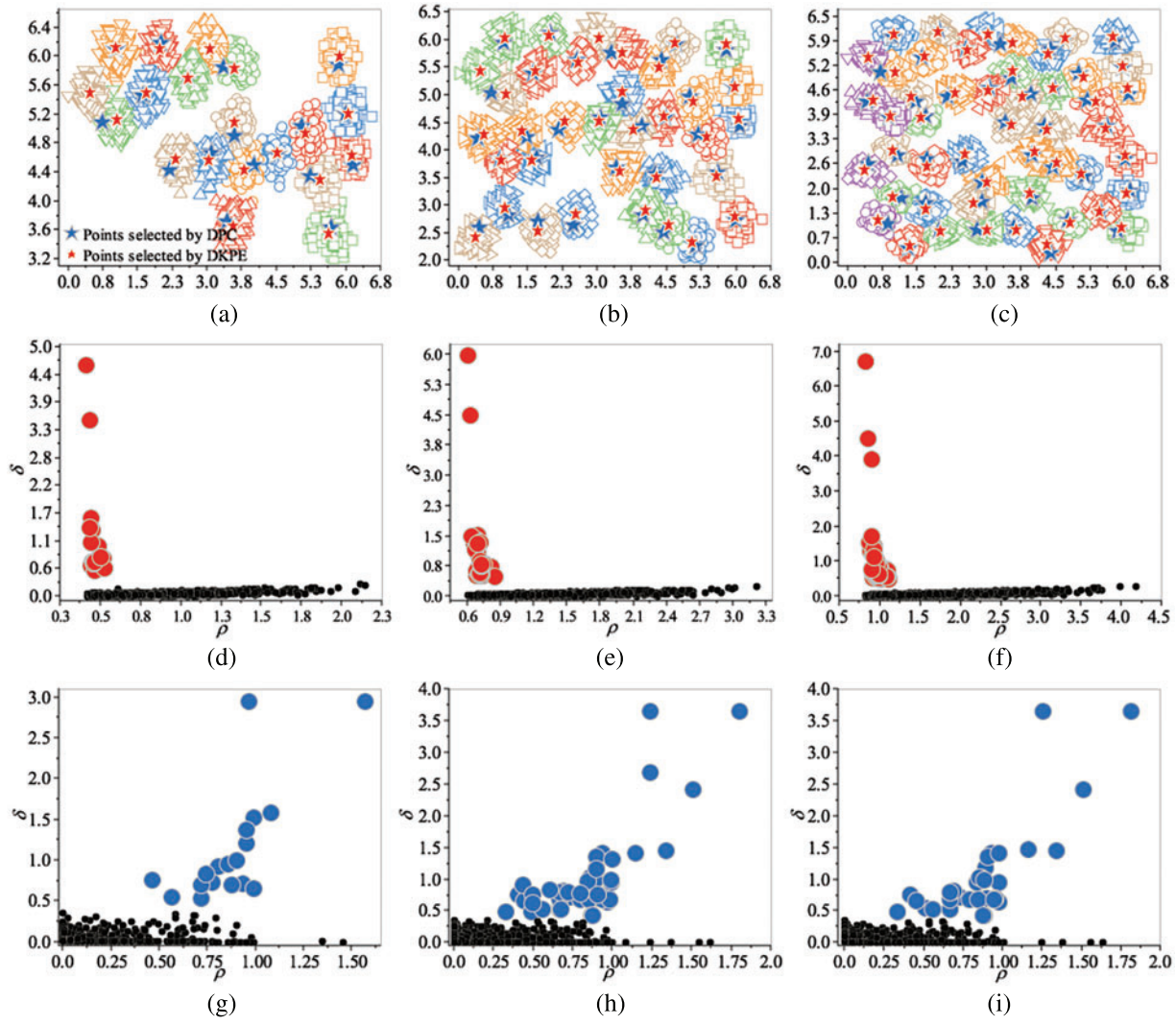

Figure 4: Results on the A1–A3 datasets. (a–c) Data distribution and high-density point distribution of A1, A2, and A3, respectively. Different colors and shapes represent different clusters. (d–f) Decision graph of the DKPE algorithm for A1, A2, and A3, respectively. Red points are selected high-density points by DKPE. (g–i) Decision graph of the DPC algorithm for A1, A2, and A3, respectively. Blue points are selected high-density points by DPC

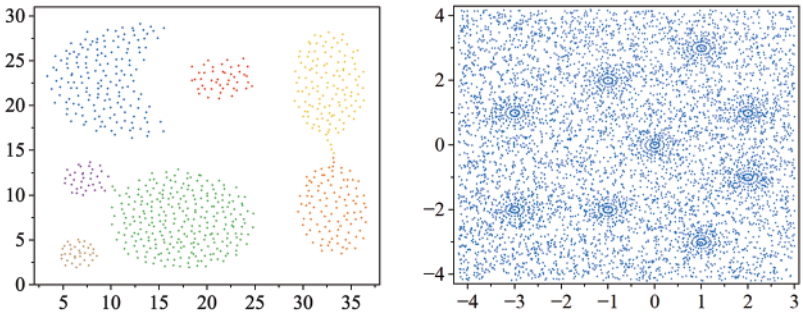

Figure 5: Distribution of Aggregation and Data6. (a) Aggregation. (b) Data6

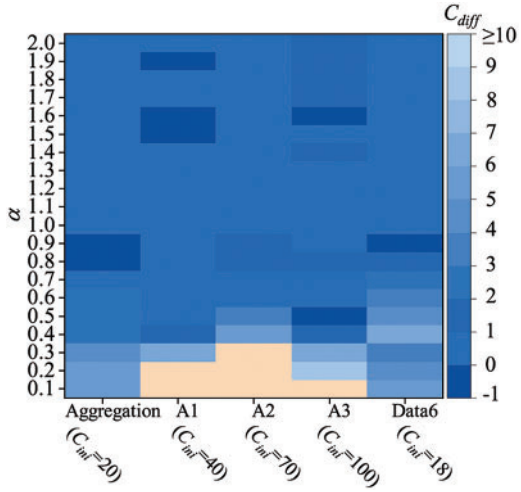

Figure 6: Graphical representation of the difference between the reference number of clusters and the final number of clusters got by HP-PCM on five datasets for various values of α

5.5 Synthetic Datasets

Here, synthetic datasets A1–A3, S1–S4 [47], Dim032 [48], Unbalance [49], Birch1 and Birch2 [50] are used to study the properties of the HP-PCM algorithm and compare with the advanced APCM algorithm and the traditional FCM algorithm. Table 3 summarizes the overall results of HP-PCM, APCM and FCM. We see that the proposed HP-PCM algorithm performs better than the APCM and FCM algorithms on all datasets. On average, the HP-PCM algorithm has 94.94% of success rate which is 2.64% and 18.26% higher than that of the APCM and the FCM algorithm respectively. As can be seen from Table 4, the HP-PCM only failed to obtain the actual clusters on the Birch2 dataset, while the APCM algorithm failed on five datasets which are the S3, S4, Unbalance, Birch1, and Birch2 datasets. We next analyze the results in more detail from two aspects: the size of samples and the number of clusters.

Table 3: Results of HP-PCM, APCM and FCM on synthetic datasets

Datasets	Indices	HP-PCM	APCM	FCM
A1	SR	**0.9953**	0.9844	0.8560
	NMI	**0.9923**	0.9884	0.9469
	ARI	**0.9903**	0.9820	0.8138
A2	SR	**0.9945**	0.9722	0.7917
	NMI	**0.9920**	0.9898	0.9403
	ARI	**0.9889**	0.9823	0.8355
A3	SR	**0.9913**	0.9686	0.7373
	NMI	**0.9900**	0.9887	0.9358
	ARI	**0.9826**	0.9621	0.8032
S1	SR	**0.9938**	0.9932	0.9114
	NMI	**0.9867**	0.9853	0.9630
	ARI	**0.9868**	0.9854	0.9071
S2	SR	**0.9718**	0.9710	0.9340
	NMI	**0.9491**	0.9475	0.9334
	ARI	**0.9415**	0.9399	0.9057
S3	SR	**0.8608**	0.8488	0.7554
	NMI	**0.8012**	0.7983	0.7566
	ARI	**0.7359**	0.7327	0.6421
S4	SR	**0.8048**	0.7998	0.7114
	NMI	**0.7264**	0.7227	0.7114
	ARI	**0.6452**	0.6357	0.6186
Dim032	SR	**1.0000**	0.9919	0.9788
	NMI	**1.0000**	0.9967	0.9924
	ARI	**1.0000**	0.9890	0.9744
Unbalance	SR	**0.9836**	0.9398	0.6663
	NMI	**0.9907**	0.9343	0.9766
	ARI	**0.9995**	0.9639	0.8063
Birch1	SR	**0.9635**	0.9297	0.8021
	NMI	**0.9739**	0.9641	0.9240
	ARI	**0.9385**	0.8939	0.7886
Birch2	SR	**0.8846**	0.7746	0.6180
	NMI	**0.9716**	0.9505	0.9322
	ARI	**0.8416**	0.7878	0.7745
Average	SR	**0.9494**	0.9249	0.8028
	NMI	**0.9431**	0.9333	0.9011
	ARI	**0.9137**	0.8959	0.8063

Table 4: Initial and final number of clusters of the HP-PCM and APCM algorithms

Dataset	HP-PCM		APCM	
	C_{ini}	C_{final}	C_{ini}	C_{final}
A1	40	**20**	40	**20**
A2	70	**35**	70	**35**
A3	100	**50**	100	**50**
S1	30	**15**	30	**15**
S2	30	**15**	30	**15**
S3	30	**15**	30	14
S4	30	**15**	30	14
Dim32	32	**16**	32	**16**
Unbalance	16	**8**	16	7
Birch1	200	**100**	200	96
Birch2	200	99	200	90

The A1–A3 datasets have an increasing number of clusters, ranging from 20 to 50. From Table 3, we can see that the values of SR, NMI, and ARI decrease for all three algorithms as the number of clusters increases. Moreover, Fig. 7 demonstrates how the CI-value depends on the size of the data (Fig. 7a), and on the number of clusters (Fig. 7b) for the HP-PCM, APCM, and FCM algorithms. The value of CI fluctuates as the size of the data changes and it seems that there is no clear dependency on the size of the data for the three algorithms. On average, the value of CI for the HP-PCM algorithm is 13.92, compared to 17.54 and 18.29 for the APCM and FCM algorithms, respectively, indicating that HP-PCM performs slightly better. This improved performance of the HP-PCM algorithm can be attributed to its ability to produce a final number of clusters that is closer to, or even equal to, the actual number of clusters, whereas the APCM algorithm consistently identifies fewer clusters across all Birch2 subsets (as shown in Fig. 8a). In general, the values of CI for all algorithms increase almost linearly along with the increasing number of clusters (see Fig. 7b). This suggests that as the number of clusters grows, the difference between the cluster centers obtained by the algorithms and the ground truth centroids also increases. As illustrated in Figs. 7b and 8b, when the number of clusters is less than 24, the HP-PCM algorithm can get correct number of clusters, resulting in a CI value of zero, which matches the ground truth centroids. For the APCM algorithm, when the number of clusters is over 16, it obtains fewer clusters than the actual number, leading to a higher value of CI than that of the HP-PCM algorithm. Overall, increasing the number of clusters will degrade the performance of HP-PCM, APCM, and FCM algorithms.

5.6 UCI Datasets

Next, we compare the clustering results of the proposed HP-PCM algorithm with ten state-of-the-art algorithms: FCM, PCM, PFCM, V-FCM, Silhouette, DPC, APCM, C-PCM, DVPFCM, and fRisk4-bA. The characteristics of the adopted UCI datasets [51] are shown in Table 2. The index values for FCM, PCM, PFCM, V-FCM, APCM, C-PCM, and HP-PCM are the best values after running 20 times.

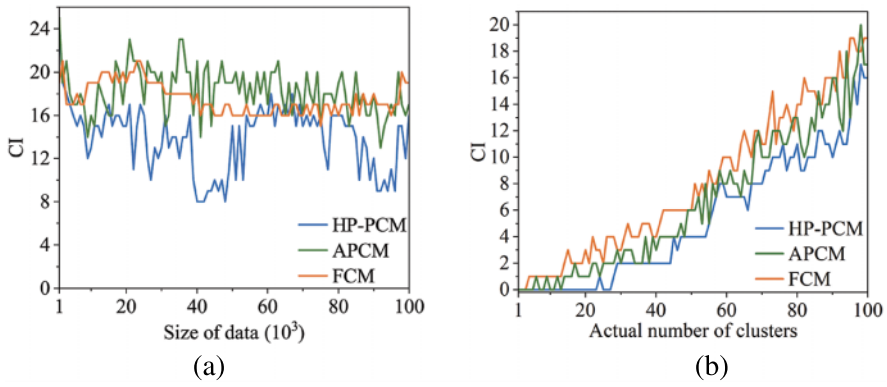

Figure 7: Dependency of the values of CI on the size of data (Birch2-random), and on the number of clusters (Birch2-sub)

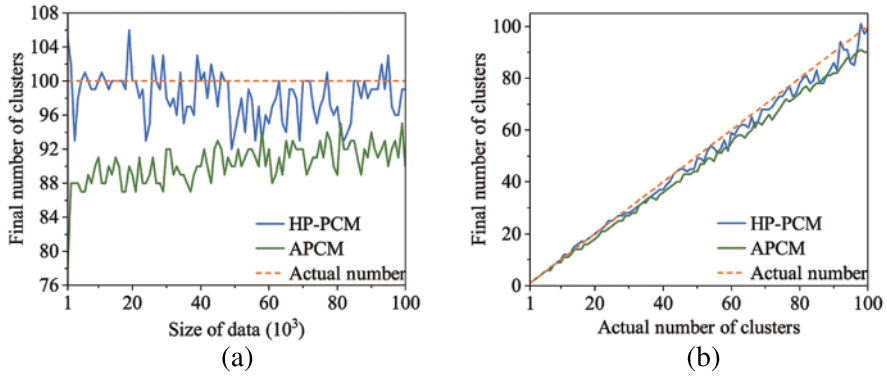

Figure 8: Dependency of the number of the final clusters on the size of data (Birch2-random), and on the number of clusters (Birch2-sub)

Tables 6–8 present the performance index values (SR, NMI, XB) obtained by 11 clustering algorithms on the selected machine learning datasets, respectively. The best results are highlighted in bold. Since DPC and Silhouette belong to hard clustering algorithms, they do not have XB values.

C_{ini} and C_{final} in Table 5 represent the initial and final cluster number of a clustering algorithm, respectively.

Tables 6–8 show that the proposed HP-PCM algorithm outperforms the other comparative algorithms across all evaluation metrics. Table 5 shows that HP-PCM can always obtain the same number of final clusters as the true number of clusters. For the four datasets, Wine, Glass, Image, and Letter, the number of high-density points identified by the DKPE method is higher than the actual number of clusters, especially for the image dataset, where the 24 high-density points are identified compared to the correct number of 4 clusters. However, our proposed HP-PCM algorithm still accurately determines the final number of clusters and ensures that the clustering results are better. This is due to the parameter φ in the HP-PCM algorithm, which regulates the influence of knowledge points, enhancing the guidance provided by true high-density points while diminishing the misleading effects of non-high-density points. As a result, guided by the high-density points, the HP-PCM algorithm can obtain more desirable clustering results.

Table 5: C_{ini} and C_{final} of each algorithm on UCI datasets

Algorithm	Iris		Glass		Image		Wine		Zoo		Letter	
	C_{ini}	C_{final}	C_{ini}	C_{final}	C_{ini}	C_{final}	C_{ini}	C_{final1}	C_{ini}	C_{final}	C_{ini}	C_{final}
HP-PCM	6	**3**	10	**6**	18	**7**	8	**3**	14	**7**	24	**4**
DVPFCM	3	**3**	6	**6**	7	**7**	3	**3**	7	**7**	4	**4**
V-FCM	3	**3**	6	**6**	7	**7**	3	**3**	7	**7**	4	**4**
fRisk4-bA	–	**3**	–	**6**	–	3	–	**3**	–	**7**	–	**4**
C-PCM	3	**3**	6	**6**	7	**7**	3	**3**	7	5	4	**4**
APCM	6	**3**	18	5	14	**7**	8	**3**	14	**7**	8	**4**
DPC	–	**3**	–	4	–	5	–	2	–	5	–	**4**
PFCM	3	**3**	6	**6**	7	**7**	3	**3**	7	**7**	4	**4**
PCM	3	2	6	3	7	4	3	2	7	4	4	**4**
FCM	3	**3**	6	**6**	7	**7**	3	**3**	7	**7**	4	**4**
Silhouette	–	2	–	5	–	5	–	2	–	5	–	**4**

Table 6: SR of each algorithm on UCI datasets

Algorithm	Iris	Glass	Image	Wine	Zoo	Letter
HP-PCM	**0.9467**	**0.5647**	**0.6652**	**0.9775**	**0.9010**	**0.6554**
DVPFCM	0.8867	0.5025	0.6081	0.9362	0.7426	0.5588
V-FCM	0.8933	0.5011	0.5971	0.8989	0.7921	0.5546
fRisk4-bA	0.8200	0.4507	0.3686	0.8933	0.7129	0.4932
C-PCM	0.8867	0.5117	0.6095	0.9663	0.6931	0.5882
APCM	0.9067	0.4664	0.6219	0.9719	0.8415	0.5930
DPC	0.9400	0.4601	0.5105	0.6348	0.7822	0.5184
PFCM	0.8733	0.4783	0.4276	0.8933	0.7128	0.5145
PCM	0.6667	0.3662	0.2810	0.8539	0.5248	0.4541
FCM	0.7333	0.4929	0.6176	0.8708	0.7921	0.4822
Silhouette	0.6667	0.4644	0.3805	0.6573	0.7822	0.5346

Table 7: NMI of each algorithm on UCI datasets

Algorithm	Iris	Glass	Image	Wine	Zoo	Letter
HP-PCM	0.8322	**0.4696**	**0.6486**	**0.9016**	**0.8895**	**0.5067**
DVPFCM	0.7615	0.3745	0.5819	0.7878	0.8122	0.3176
V-FCM	0.7496	0.3618	0.5073	0.7082	0.8022	0.3155
fRisk4-bA	0.6552	0.3236	0.4851	0.6947	0.6897	0.2360

(Continued)

Table 7 (continued)

Algorithm	Iris	Glass	Image	Wine	Zoo	Letter
C-PCM	0.7419	0.3832	0.5711	0.8759	0.7130	0.3739
APCM	0.8057	0.3586	0.6056	0.8750	0.8290	0.4228
DPC	**0.9400**	0.2813	0.5025	0.6102	0.7514	0.3716
PFCM	0.8244	0.3029	0.3857	0.6983	0.7169	0.3163
PCM	0.6667	0.0474	0.3648	0.6506	0.5719	0.2442
FCM	0.7333	0.3602	0.5163	0.6697	0.7822	0.2582
Silhouette	0.6667	0.3411	0.3798	0.4232	0.7427	0.3589

Table 8: XB of each algorithm on UCI datasets

Algorithm	Iris	Glass	Image	Wine	Zoo	Letter
HP-PCM	**0.0554**	**0.0587**	**0.2277**	**0.0036**	**0.0235**	**0.0094**
DVPFCM	0.3138	0.3209	0.5492	0.3209	0.6655	0.3209
V-FCM	0.2894	0.3990	0.3153	0.4410	0.3641	0.4535
fRisk4-bA	0.2536	1.5939	0.6927	0.5497	0.4580	0.7527
C-PCM	0.3380	0.5254	0.5607	0.4816	0.7955	0.3006
APCM	0.9214	0.0632	0.5099	0.0122	0.0870	0.2446
PFCM	0.5340	1.1242	1.4935	0.3801	54.60	1.6072
PCM	15.982	469.27	199.90	1.4064	30.10	271.67
FCM	0.5768	1.7865	1.7790	0.7862	0.3563	0.8040

Both the fRisk4-bA and Silhouette algorithms determine the optimal number of clusters based on the evaluation index values. fRisk4-bA can accurately obtain the optimal number of clusters (except for the Image dataset), but its SR and NMI values are not higher than those of the other algorithms, and its XB value is not less. This is because fRisk4-bA uses FCM to obtain the clustering results but FCM is sensitive to noises. For the Silhouette algorithm, the main reason for its poorer clustering performance is that it fails to correctly determine the optimal number of clusters.

The final number of clusters obtained by PCM tends to be smaller than the actual number of clusters on the Iris, Wine, Zoo, and Image datasets, likely due to the influence of clustering center uniformity. The knowledge-based clustering algorithms DVPFCM and V-FCM replace one of the final cluster centers with the highest-density point, so their performance is better than that of the FCM, PCM, PFCM algorithms and even better than that of CPCM, and APCM on some datasets. However, their clustering flexibility is constrained by the highest-density point. When the obtained highest-density point is not the true high-density point, DVPFCM and V-FCM may obtain misleading clustering results. Consequently, their clustering performance is not always better than that of the CPCM and APCM algorithms.

To further illustrate the performance improvement of the proposed HP-PCM algorithm, we counted the percentage increase in SR of the HP-PCM algorithm relative to the compared algorithms, as shown in Table 9. From Table 9, we can see that HP-PCM has at most a 137.49% improvement in

the clustering success rate. On average, the HP-PCM algorithm can improve SR by 8.44% and up to 60.70% relative to the other algorithms.

Table 9: Percentage increase in SR of HP-PCM

Algorithm	Iris	Wine	Glass	Zoo	Image	Letter	Average
DVPFCM	6.77	4.19	12.37	21.33	09.39	17.29	11.89
V-FCM	5.98	8.74	12.69	13.75	11.41	18.18	11.79
fRisk4-bA	15.45	9.43	25.29	16.39	80.47	32.89	29.99
CPCM	6.77	1.16	10.35	30.00	09.14	11.42	11.47
APCM	4.41	0.58	21.07	07.07	06.96	10.52	8.44
DPC	0.71	53.99	22.73	15.19	30.30	26.43	24.89
PFCM	8.40	9.43	18.06	26.40	55.57	27.39	24.21
PCM	42.00	14.47	54.21	71.68	137.49	44.33	60.70
FCM	19.10	12.25	14.66	13.75	07.71	35.92	17.23
Silhouette	42.00	48.71	21.60	15.19	74.82	22.60	37.48

In addition, Fig. 9 counts the number of iterations required for each clustering algorithm on the six used UCI datasets. The fRisk4-bA, DPC, and Silhouette algorithms are not iterative, so they are not considered with this statistic. It can be seen from Fig. 9 that the proposed HP-PCM algorithm requires the fewest iterations, followed by the DVPFCM and V-FCM algorithms. This suggests that the introduction of knowledge hints can help speed up the algorithm convergence.

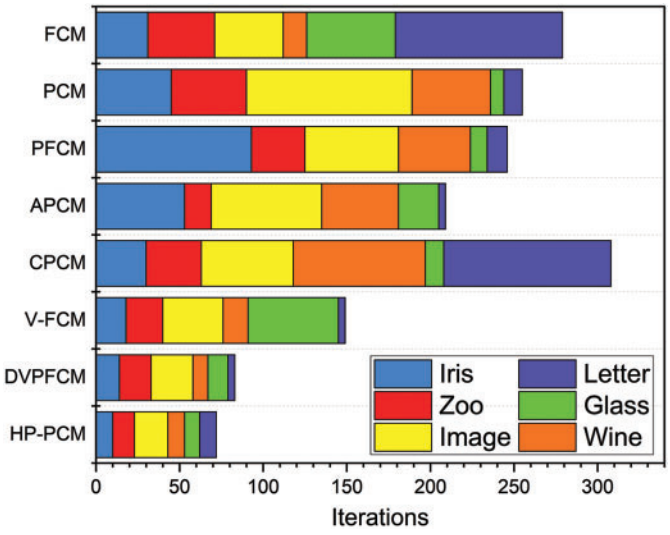

Figure 9: Average numbers of iteration for the tested algorithms

Table 10 summarizes the average running time of the tested algorithms on the used UCI datasets. It is evident that the Silhouette and fRisk4bA algorithms are the slowest among all algorithms. This slow performance highlights the significant time requirements for determining the number of clusters and analyzing dataset structure using clustering indices. In contrast, the HP-PCM algorithm shows

markedly better time efficiency compared to other algorithms designed for identifying the optimal number of clusters, including Silhouette, APCM, and fRisk4-bA.

Table 10: Average time cost of the tested algorithms (s)

Algorithm	Iris	Wine	Glass	Zoo	Image	Letter	Average rank
HP-PCM	0.0237	0.0489	0.0324	0.0130	**0.0340**	**1.4400**	4.3
DVPFCM	0.0390	0.0440	0.0110	0.0265	1.2431	13.479	6.3
V-FCM	0.0183	0.0159	0.0254	0.0119	0.2180	2.9702	3.8
fRisk4-bA	0.5316	0.6755	1.1294	0.2151	47.9498	45.451	10.0
CPCM	0.0201	0.0349	0.0237	0.0060	0.4543	6.1308	4.7
APCM	0.0853	0.2732	0.1206	0.0403	1.1450	8.0768	7.8
PFCM	0.0016	0.0008	0.0264	0.0606	0.2312	3.0029	4.8
PCM	**0.0004**	0.0006	**0.0005**	**0.0007**	0.1878	3.1391	2.2
FCM	0.0006	**0.0005**	0.0036	0.0155	0.0958	2.7559	2.3
Silhouette	0.1915	0.0835	0.0969	0.0652	1.6773	27.901	8.7

Overall, HP-PCM is the best in terms of various performance indices and the required number of iterations.

5.7 Ablation Study

Comparing the above clustering results of the proposed HP-PCM algorithm and the APCM algorithm, we can observe that high-density points help the algorithm to determine the final number of clusters more accurately and efficiently. To reveal the impact of high-density points on the adaptive determination of the number of clusters, we designed four algorithms and compared their performance. The first algorithm, denoted as Ablation 1, replaces high-density points with a randomly sampled set of points from the dataset. The second algorithm, denoted as Ablation 2, obtains high-density points from the DPC algorithm. The third algorithm, denoted as Ablation 3, is our proposed HP-PCM algorithm, which uses high-density points provided by the DKPE algorithm. The results of these three algorithms on the six UCI datasets are shown in Table 11. Table 11 demonstrates that Ablation 3, the HP-PCM algorithm, outperforms Ablation 2 and Ablation 1 in terms of SR and NMI across all UCI datasets. Furthermore, Ablation 2 outperforms Ablation 1 on five out six datasets with respect to SR and NMI. In conclusion, the ablation experimental results clearly indicate that high-density points play a crucial role in enhancing the accuracy and efficiency of clustering algorithms.

Table 11: Ablation study of the proposed algorithm on six UCI datasets

Dataset	SR			NMI		
	Ablation 1	Ablation 2	Ablation 3	Ablation 1	Ablation 2	Ablation 3
Iris	0.6667	0.9116	**0.9467**	0.6623	0.7634	**0.8322**
Wine	0.9213	0.9270	**0.9775**	0.7963	0.8011	**0.9016**
Glass	0.4554	0.4883	**0.5647**	0.3684	0.4350	**0.4696**
Zoo	0.6949	0.7797	**0.9010**	0.7011	0.7642	**0.8895**

(Continued)

Table 11 (continued)

Dataset	SR			NMI		
	Ablation 1	Ablation 2	Ablation 3	Ablation 1	Ablation 2	Ablation 3
Image	0.4358	0.3322	**0.6652**	0.4116	0.3645	**0.6486**
Letter	0.0817	0.0613	**0.6554**	0.0660	0.0468	**0.5067**

6 Conclusion

In this study, we propose a novel knowledge-driven clustering algorithm called the HP-PCM algorithm. First, the DKPE method is proposed to obtain P data points with higher density. Then, these high-density points are added to the objective function of PCM to construct a new function, which is the HP-PCM algorithm objective function. In the HP-PCM algorithm, the high-density points serve as guides, rather than deciders, facilitating the algorithm to find dense regions within the dataset. Furthermore, the proposed HP-PCM algorithm can automatically and accurately determine the final number of clusters by employing the cluster merging mechanisms. Finally, compared with 10 existing clustering algorithms on different datasets, the experimental results show that our proposed HP-PCM algorithm is better at determining the initial number of clusters, getting the success rate of clustering, and obtaining the optimal number of clusters. By comparing the number of iterations, it is shown that the knowledge-driven clustering algorithms DVPFCM, V-FCM, and our proposed HP-PCM algorithms have fewer iterations, which fully shows that the high-density points help speed up the algorithm convergence.

Our future work will focus on the extension of HP-PCM to cluster the datasets with irregular shapes. Moreover, it would be helpful to combine fuzzy clustering with fuzzy inference [52].

Acknowledgement: Special thanks to the reviewers of this paper for their valuable feedback and constructive suggestions, which greatly contributed to the refinement of this research.

Funding Statement: This work was supported by the National Key Research and Development Program of China (No. 2022YFB3304400) and the National Natural Science Foundation of China (Nos. 6230311, 62303111, 62076060, 61932007, and 62176083), and the Key Research and Development Program of Jiangsu Province of China (No. BE2022157).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Xianghui Hu, Yichuan Jiang; data collection: Yiming Tang; analysis and interpretation of results: Xianghui Hu, Witold Pedrycz, Jiuchuan Jiang; draft manuscript preparation: Xianghui Hu, Yiming Tang, Jiuchuan Jiang. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: All the synthetic datasets used in this article are sourced from "Dynamic local search for clustering with unknown number of clusters" [44], "Clustering aggregation" [45], "Generalized entropy based possibilistic fuzzy c-means for clustering noisy data and its convergence proof" [46], "Iterative shrinking method for clustering problems" [47], "Fast agglomerative clustering using a k-nearest neighbor graph" [48], "Set-matching methods for external cluster validity" [49], "BIRCH: A new data clustering algorithm and its applications" [50]; all the UCI datasets used in this article are available from: https://archive.ics.uci.edu/ (accessed on 20 May 2024).

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] F. Fontana, C. Klahn, and M. Meboldt, "Value-driven clustering of industrial additive manufacturing applications," *Int. J. Manuf. Technol. Manag.*, vol. 30, no. 2, pp. 366–390, Feb. 2019. doi: 10.1108/JMTM-06-2018-0167.

[2] H. Y. Li, B. Zhang, S. Qin, and J. L. Peng, "UAV-Clustering: Cluster head selection and update for UAV swarms searching with unknown target location," presented at the WoWMoM, Belfast, UK, Jun. 14–17, 2022, pp. 483–488, doi: 10.1109/WoWMoM54355.2022.00075.

[3] Y. M. Tang, F. J. Ren, and W. Pedrycz, "Fuzzy C-Means clustering through SSIM and patch for image segmentation," *Appl. Soft Comput.*, vol. 87, no. 4, Feb. 2020, Art. no. 105928. doi: 10.1016/j.asoc.2019.105928.

[4] F. Liu, L. C. Jiao, and X. Tang, "Task-oriented GAN for PolSAR image classification and clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2707–2719, Sep. 2019. doi: 10.1109/TNNLS.2018.2885799.

[5] P. Fränti, M. Rezaei, and Q. P. Zhao, "Centroid index: Cluster level similarity measure," *Pattern Recognit.*, vol. 47, no. 9, pp. 3034–3045, Sep. 2014. doi: 10.1016/j.patcog.2014.03.017.

[6] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *J. Cybern.*, vol. 3, no. 3, pp. 32–57, Sep. 1973. doi: 10.1080/01969727308546046.

[7] J. C. Bezdek, "Objective function clustering," in *Pattern Recognition with Fuzzy Objective Function Algorithms*, 1st ed., New York, NY, USA: Springer, 2013, pp. 65–79, doi: 10.1007/978-1-4757-0450-1_3.

[8] A. Bagherinia, M. B. Behrooz, H. Mehdi, and H. Parvin, "Reliability-based fuzzy clustering ensemble," *Fuzzy Sets Syst.*, vol. 413, pp. 1–28, Jun. 2021. doi: 10.1016/j.fss.2020.03.008.

[9] H. Li and M. Wei, "Fuzzy clustering based on feature weights for multivariate time series," *Knowl.-Based Syst.*, vol. 197, no. 8, Jun. 2020, Art. no. 105907. doi: 10.1016/j.knosys.2020.105907.

[10] S. D. Xenaki, K. D. Koutroumbas, and A. A. Rontogiannis, "A novel adaptive possibilistic clustering algorithm," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 4, pp. 791–810, Aug. 2016. doi: 10.1109/TFUZZ.2015.2486806.

[11] S. D. Xenaki, K. D. Koutroumbas, and A. A. Rontogiannis, "Sparsity-aware possibilistic clustering algorithms," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 6, pp. 1611–1626, Dec. 2016. doi: 10.1109/TFUZZ.2016.2543752.

[12] W. Pedrycz, V. Loia, and S. Senatore, "Fuzzy clustering with viewpoints," *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 2, pp. 274–284, Apr. 2010. doi: 10.1109/TFUZZ.2010.2040479.

[13] Y. M. Tang, X. H. Hu, W. Pedrycz, and X. C. Song, "Possibilistic fuzzy clustering with high-density viewpoint," *Neurocomputing*, vol. 329, no. 2, pp. 407–423, Feb. 2019. doi: 10.1016/j.neucom.2018.11.007.

[14] Y. M. Tang, Z. F. Pan, X. H. Hu, W. Pedrycz, and R. H. Chen, "Knowledge-induced multiple kernel fuzzy clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 12, pp. 14838–14855, Dec. 2023. doi: 10.1109/TPAMI.2023.3298629.

[15] X. H. Hu, Y. M. Tang, W. Pedrycz, K. Di, J. C. Jiang, and Y. C. Jiang, "Fuzzy clustering with knowledge extraction and granulation," *IEEE Trans. Fuzzy Syst.*, vol. 31, no. 4, pp. 1098–1112, Apr. 2023. doi: 10.1109/TFUZZ.2022.3195033.

[16] W. Dai, Q. Yang, G. R. Xue, and Y. Yu, "Self-taught clustering," presented at the ICML, Helsinki, Finland, Jul. 5–9, 2008, pp. 200–207. doi: 10.1145/1390156.1390182.

[17] W. H. Jiang and F. L. Chung, "Transfer spectral clustering," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, Berlin, Heidelberg, 2012, vol. 7524, pp. 789–803. doi: 10.1007/978-3-642-33486-3.

[18] Z. H. Deng, Y. Z. Jiang, F. L. Chung, H. Ishibuchi, K. S. Choi, and S. T. Wang, "Transfer prototype-based fuzzy clustering," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 5, pp. 1210–1232, Oct. 2016. doi: 10.1109/TFUZZ.2015.2505330.

[19] A. E. Ezugwu *et al.*, "A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects," *Eng Appl. Artif. Intell.*, vol. 110, Apr. 2022. doi: 10.1016/j.engappai.2022.104743.

[20] S. D. Nguyen, V. S. T. Nguyen, and N. T. Pham, "Determination of the optimal number of clusters: A fuzzy-set based method," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 9, pp. 3514–3526, Sep. 2022. doi: 10.1109/TFUZZ.2021.3118113.

[21] Y. M. Tang, J. J. Huang, W. Pedrycz, B. Li, and F. J. Ren, "A fuzzy clustering validity index induced by triple center relation," *IEEE Trans. Cybern.*, vol. 53, no. 8, pp. 5024–5036, Aug. 2023. doi: 10.1109/TCYB.2023.3263215.

[22] N. Wiroonsri, "Clustering performance analysis using a new correlation-based cluster validity index," *Pattern Recogn.*, vol. 145, no. 6, Jan. 2024, Art. no. 109910. doi: 10.1016/j.patcog.2023.109910.

[23] D. Jollyta, S. Efendi, M. Zarlis, and H. Mawengkang, "Analysis of anoptimal cluster approach: A review paper," *J. Phys.: Conf. Ser.*, vol. 2421, no. 1, 2023, Art. no. 012015. doi: 10.1088/1742-6596/2421/1/012015.

[24] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014. doi: 10.1126/science.1242072.

[25] T. Qiu and Y. Li, "Fast LDP-MST: An efficient density-peak-based clustering method for large-size datasets," *IEEE Trans Knowl. Data Eng*, vol. 35, no. 5, pp. 4767–4780, May 2023. doi: 10.1109/TKDE.2022.3150403.

[26] D. Cheng, Y. Li, S. Xia, G. Wang, J. Huang and S. Zhang, "A fast granular-ball-based density peaks clustering algorithm for large-scale data," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–14, 2023. doi: 10.1109/TNNLS.2023.3300916.

[27] J. S. Zhang and Y. W. Leung, "Improved possibilistic C-means clustering algorithms," *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 2, pp. 209–217, Apr. 2004. doi: 10.1109/TFUZZ.2004.825079.

[28] M. Barni, V. Cappellini, and A. Mecocci, "Comments on a possibilistic approach to clustering," *IEEE Trans. Fuzzy Syst.*, vol. 4, no. 3, pp. 393–396, Aug. 1996. doi: 10.1109/91.531780.

[29] H. Timm, C. Borgelt, C. Döring, and R. Kruse, "An extension to possibilistic fuzzy cluster analysis," *Fuzzy Set Syst*, vol. 147, no. 1, pp. 3–16, Oct. 2004. doi: 10.1016/j.fss.2003.11.009.

[30] N. R. Pal, K. Pal, J. M. Keller, and J. C. Bezdek, "A possibilistic fuzzy c-means clustering algorithm," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 4, pp. 517–530, Aug. 2005. doi: 10.1109/TFUZZ.2004.840099.

[31] M. S. Yang and J. B. M. Benjamin, "Sparse possibilistic c-means clustering with Lasso," *Pattern Recogn.*, vol. 138, Jun. 2023, Art. no. 109348. doi: 10.1016/j.patcog.2023.109348.

[32] C. Wu and X. Zhang, "A self-learning iterative weighted possibilistic fuzzy c-means clustering via adaptive fusion," *Expert Syst. Appl.*, vol. 209, no. 1, Dec. 2022, Art. no. 118280. doi: 10.1016/j.eswa.2022.118280.

[33] M. S. Yang and K. L. Wu, "A similarity-based robust clustering method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 4, pp. 434–448, Apr. 2004. doi: 10.1109/TPAMI.2004.1265860.

[34] M. S. Yang and C. Y. Lai, "A robust automatic merging possibilistic clustering method," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 1, pp. 26–41, Feb. 2011. doi: 10.1109/TFUZZ.2010.2077640.

[35] Y. Y. Liao, K. X. Jia, and Z. S. He, "Similarity measure based robust possibilistic c-means clustering algorithms," *J. Convergence Inf. Technol.*, vol. 6, no. 12, pp. 129–138, Dec. 2011. doi: 10.4156/jcit.vol6.issue12.17.

[36] J. V. de Oliveira and W. Pedrycz, "Relational fuzzy clustering," in *Advances in Fuzzy Clustering and its Applications*, 1st ed. Chichester, England: John Wiley & Sons Ltd., 2007, pp. 38–40. doi: 10.1002/9780470061190.

[37] S. B. Zhou, Z. Y. Xu, and F. Liu, "Method for determining the optimal number of clusters based on agglomerative hierarchical clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 12, pp. 3007–3017, Dec. 2017. doi: 10.1109/TNNLS.2016.2608001.

[38] N. R. Pal and J. C. Bezdek, "On cluster validity for the fuzzy c-means model," *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 3, pp. 370–379, Aug. 1995. doi: 10.1109/91.413225.

[39] J. C. Bezdek and N. R. Pal, "Some new indexes of cluster validity," *IEEE Trans. Syst. Man Cybern B-Cybern*, vol. 28, no. 3, pp. 301–315, Jun. 1998. doi: 10.1109/3477.678624.

[40] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 9, no. 18, pp. 509–517, Sep. 1975. doi: 10.1145/361002.361007.

[41] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, 1987. doi: 10.1016/0377-0427(87)90125-7.
[42] A. Strehl and J. Ghosh, "Cluster ensembles—A knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, no. 3, pp. 583–617, 2002. doi: 10.1162/153244303321897735.
[43] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 8, pp. 841–847, Aug. 1991. doi: 10.1109/34.85677.
[44] I. Kärkkäinen and P. Fränti, "Dynamic local search for clustering with unknown number of clusters," in *Proc. Int. Conf. Pattern Recogn.*, Quebec City, QC, Canada, 2002, vol. 2, pp. 240–243. doi: 10.1109/ICPR.2002.1048283.
[45] A. Gionis, H. Mannila, and P. Tsaparas, "Clustering aggregation," in *Proc. Int. Conf. Data Eng.*, Tokyo, Japan, 2005, vol. 1, no. 1, pp. 341–352. doi: 10.1109/ICDE.2005.34.
[46] S. Askari, N. Montazerin, M. F. Zarandi, and E. Hakimi, "Generalized entropy based possibilistic fuzzy c-means for clustering noisy data and its convergence proof," *Neurocomputing*, vol. 219, no. 2–3, pp. 186–202, Jan. 2017. doi: 10.1016/j.neucom.2016.09.025.
[47] P. Fränti and I. Virmajoki, "Iterative shrinking method for clustering problems," *Pattern Recogn.*, vol. 39, no. 5, pp. 761–775, May 2006. doi: 10.1016/j.patcog.2005.09.012.
[48] P. Fränti, O. Virmajoki, and V. Hautamäki, "Fast agglomerative clustering using a k-nearest neighbor graph," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1875–1881, Nov. 2006. doi: 10.1109/TPAMI.2006.227.
[49] M. Rezaei and P. Fränti, "Set-matching methods for external cluster validity," *IEEE Trans. Knowl. Data Eng*, vol. 28, no. 8, pp. 2173–2186, Aug. 2016. doi: 10.1109/TKDE.2016.2551240.
[50] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: A new data clustering algorithm and its applications," *Data Min. Knowl. Discov.*, vol. 1, no. 2, pp. 141–182, 1997. doi: 10.1023/A:1009783824328.
[51] D. Dua and C. Graff, "UCI machine learning repository," Accessed: May 20, 2024, Jan. 2017. [Online]. Available: http://archive.ics.uci.edu/ml
[52] Y. Tang, W. Pedrycz, and F. Ren, "Granular symmetric implicational method," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 6, no. 3, pp. 710–723, Jun. 2022. doi: 10.1109/TETCI.2021.3100597.