Knowledge-Based Systems 263 (2023) 110261

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Cloud-Cluster: An uncertainty clustering algorithm based on cloud model

Yue Liu [a,b,*], Zitu Liu [a], Shuang Li [a], Yike Guo [c], Qun Liu [d], Guoyin Wang [d]

[a] School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China
[b] Shanghai Engineering Research Center of Intelligent Computing System, Shanghai 200444, China
[c] Department of Computing, Imperial College, London SW7 2AZ, UK
[d] Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

ARTICLE INFO

Article history:
Received 31 August 2022
Received in revised form 17 December 2022
Accepted 2 January 2023
Available online 5 January 2023

Keywords:
Clustering
Cloud model
Uncertainty
Cloud transformation algorithm

ABSTRACT

As a cornerstone of the world, uncertainty embodies the nature of data and knowledge. Existing uncertainty theory-based clustering algorithms learn fuzziness, i.e., the uncertainty of clustering objects belonging to different clusters. However, these algorithms do not refer to the fuzziness of objects themselves, i.e., the randomness of data. Here, we propose a clustering algorithm named Cloud-Cluster, which simultaneously characterizes the fuzziness and randomness of objects to reserve uncertain information, and to describe clusters into concepts. It embeds random uncertainty of concepts to extend the data distribution range for better data partitions and gradually constructs accurate concepts by an improved backward cloud transformation algorithm (MBCT-SR-Ex). Moreover, to ensure that the concept clustering process gradually converges, Cloud-Cluster introduces the Cluster Concept Drift Degree to evaluate the uncertainty of concepts during the clustering process. Experiments on UCI and OpenML clustering datasets show that Cloud-Cluster improves the average clustering accuracy by over 14% compared to K-Means and uncertainty theory-based clustering algorithms. Extensive experimental results on the evaluation of uncertainty show that Cloud-Cluster can handle the uncertainty of datasets in the clustering process well, in addition to exhibiting robustness with unclear clusters.

1. Introduction

Uncertainty, as the cornerstone of the objective world, embodies the nature of data and knowledge [1,2]. Data mining of uncertain information has become recently an active area of research [3–6]. Clustering analysis, as a fundamental data mining method, plays an important role in many fields [7–10]. However, traditional clustering methods mostly address data in a certain way without uncertain information [11–19]. For example, the K-Means algorithm strictly divides each dataset into clusters, and the cluster boundaries are clear. Therefore, to incorporate uncertain information to help humans describe the objective world, research on uncertainty clustering has become an important challenge.

Several existing clustering methods have been proposed to address uncertainty. They incorporate traditional uncertainty theories, such as fuzzy sets [20], rough sets [21], and probability theory [22], into clustering models to mainly describe the fuzziness of data. In fact, the uncertainty in reality is often complicated. Uncertainty includes incompleteness, inconsistency, fuzziness, randomness, etc. Randomness and fuzziness, as two of the most basic uncertainties, have a strong correlation [23]. Randomness means that the definition of an event is deterministic, such as the probability that an event will occur, but whether or not an event will occur is uncertain. Fuzziness describes event ambiguity which measures the degree to which an event occurs, not whether it occurs [24]. For example, from the results of the shooting game, hitting or missing the target has randomness, and there is fuzziness in measuring shooting ability by hitting the target [23]. Uncertainty in fuzzy sets and rough sets mainly describes the fuzzy state of events, while uncertainty in probability theory mainly uses a distribution to describe the fuzzy probability of events occurring. There are obvious differences between them. Probability theory focuses on the connotation of the object (the property of an object), while fuzzy sets and rough sets focus on the extension of the object (exact range of a certain property). Fuzziness in these algorithms refers to the uncertainty of clustering objects belonging to different clusters, in which probability theory even describes the connotation of the data. However, they do not refer to the fuzziness of cluster objects themselves, i.e., the randomness of data. Data are represented by information, and some information with fuzziness may affect the judgment of data division. Without considering randomness, the

* Corresponding author.
E-mail address: yueliu@shu.edu.cn (Y. Liu).

https://doi.org/10.1016/j.knosys.2023.110261
0950-7051/© 2023 Elsevier B.V. All rights reserved.

division probability of data may have a deterministic division result. Thus, due to information with fuzziness, the division of data requires the introduction of randomness. To improve the learning of uncertain information in data, it is necessary to simultaneously characterize the fuzziness and randomness of objects, reserve the uncertain information, and optimize the clustering model.

However, currently, clustering methods focus on uncertainty such as fuzziness, while randomness is less considered. Therefore, as an important human behavior, clustering analysis needs to reflect random uncertainty while analyzing the data, which means that the probabilities of data partitioning in clustering should take into account the role of randomness. Cloud model theory proposed by Li [25,26] adds hyper entropy to Gaussian distribution to represent randomness, which reflects the uncertainty in qualitative concepts and reveals the link between fuzziness and randomness of objects. Thus, this work focuses on uncertainty in clustering analysis and constructs a novel clustering algorithm based on the probabilistic framework in cloud model theory, incorporating uncertainty to represent the concepts of data to address more comprehensive uncertain information and thus gain complete knowledge from data.

Previous works on cloud models have mainly focused on uncertainty in the fields of image segmentation [27–29], performance prediction [30–33], risk evaluation [34–36], etc. These methods form the distribution curves of data and fit the curves through multiple cloud models to extract fine-grained concepts. However, the concepts cannot be simply extracted from the distribution curve in high-dimensional data. This makes clustering studies on cloud model few for multidimensional data. Thus, to build a clustering method based on the cloud model, the first challenge is how to obtain multiple concepts from multidimensional data based on cloud model for clustering while embedding random uncertainty. In addition, the model needs to converge to obtain the optimal clustering result, i.e., the stability of concepts is important. Xu et al. [37] proposed the drift degree to measure the fluctuation degree to evaluate the stability of concepts. However, there are many concepts of datasets in clustering tasks. There is still a lack of methods to evaluate the fluctuation of multiple concepts of the dataset in the clustering process. To solve these challenges, this paper proposes a novel clustering method based on cloud model theory named Cloud-Cluster to combine random uncertainty to reserve uncertain information and make a more comprehensive clustering, which is mainly composed of Concept-Based Refinement Aggregation and Concept Uncertainty Evaluation. The contributions of our work are highlighted as follows.

- To group data with random uncertainty embedded, Concept-Based Refinement Aggregation Method is proposed. It uses a cloud model to represent data as multiple concepts, while adding randomness of concepts to extend the range of the data distribution and to obtain probabilities of data division of the cluster data.
- To evaluate the uncertainty of concepts and the convergence of the algorithm, Concept Uncertainty Evaluation Method is proposed. It measures the fluctuation of multiple concepts by Cluster Concept Drift Degree, showing the uncertainty of concepts in the whole clustering process.
- Many internal and external cluster validity indices are used in experiments to demonstrate that Cloud-Cluster has good performance and learns the internal information of datasets more accurately. Extensive quantitative experimental results also verify the effectiveness and efficiency of our approaches in clustering.

The remainder of this paper is organized as follows: Section 2 introduces the related work of Cloud-Cluster from three aspects: clustering algorithms based on fuzzy set theory, clustering algorithms based on rough set theory, and algorithms based on cloud model. Section 3 presents the proposed Cloud-Cluster. Extensive empirical experiments for Cloud-Cluster are analyzed in Section 4. Finally, Section 5 concludes this work.

2. Related works

We discuss existing work that aims at either clustering algorithms or cloud models. The clustering algorithm and cloud model are inherently related, since improvements to learning clustering algorithms require uncertain evaluation metrics that are sensitive to subtle details. This section is divided into three parts: (1) clustering algorithms based on fuzzy set theory; (2) clustering algorithms based on rough set theory; and (3) algorithms based on cloud models.

2.1. Clustering algorithms based on fuzzy set theory

Fuzzy set theory effectively alleviates the weakness of hard-divided objects [20]. Fuzzy C-Means (FCM) [38] is one of the most commonly used fuzzy clustering approaches, and has been used in a wide variety of applications due to its ability to handle fuzziness [39–42]. It still has some limitations such as sensitivity to noise due to Euclidean distance measurement and the ignorance of neighborhood information. To overcome these shortcomings, weighted Euclidean distance [43], Mahalanobis distance [44], distance based on membership and typical values [45], metric-based distance [46], and distance function [47,48] have been proposed. Meng et al. [49] incorporated the derivative information into the distance measure to capture the characteristics among functional data. For FCM, the weighted factor m is an important parameter related to the algorithm performance. To avoid the subjectivity of manual setting, Jing [50] et al. combined a fuzzy matrix with Jacobian matrix to obtain an appropriate range of m. Ren [51] et al. proposed an improved particle swarm optimization (PSO) method to measure membership by calculating the adaptive weighted index m. The above algorithms find better data partitions and depict the fuzziness by improving the similarity criterion, cluster prototypes, weight factor, etc.

2.2. Clustering algorithms based on rough set theory

Rough set theory (RST) [21,52] handles uncertain and incomplete knowledge in data by discretizing objects, reducing attributes, etc. Rough set theory cannot handle discrete data with too many attribute combinations. Thus, researchers generally tend to incorporate RST into clustering algorithms [53]. K-Means integrated with rough set [54] reduces the role of unimportant attributes and makes full use of the uncertain information of the boundary to cluster. A density-based clustering algorithm with rough set [55] partitions data into clusters according to core/noncore attributes, the density of clusters, and the uncertain boundary of clusters. For FCM based on rough set [56], the fuzzy boundaries are redefined by the upper and lower approximation concept, which obtains more appropriate uncertain boundaries and effectively promotes clustering performance. Existing clustering methods combined with rough sets effectively approximate the boundary of clusters and reduce complex data, which makes clustering more efficient. Fuzzy set and rough set theory treat the fuzziness in uncertainty but ignore the randomness.

2.3. Algorithms based on cloud model

A Gaussian mixed model based on probability theory is used to cluster data, which mainly uses the distribution to describe the fuzzy probability of data division. However, it does not refer to the fuzziness of objects themselves. Cloud model theory proposed by Li [25] reflects the uncertainty of concept and reveals the relationship between the fuzziness and randomness of objects. The cloud model uses three numerical characteristics, expectation Ex, entropy En and hyper entropy He, to express a concept. Ex

is the most representative sample of a concept; *En* is used to determine the granularity scale of the concept, and *He* is used to depict the uncertainty of the granularity of the concept. Thus, a qualitative concept C can be expressed by numerical characteristics (Ex, En, He), and the extension of the concept C can be represented by data points in the concept.

Previous studies employ cloud models and reduce the image into one dimension to extract concepts in images, which effectively avoids the problem of hard segmentation [27]. For images with complex structures, uncertain problems are solved by combining the cloud model and Gaussian mixture model for multigranularity image segmentation after dimensionality reduction [57]. In fact, knowledge represented by data is multidimensional. Deng et al. [58] proposed a two-dimensional cloud model and combined time features to learn more information from data, while other works were still addressing low dimensions. Thus, it is important to study the multi-concept extraction methods in multiple dimensions. Based on the above, this study proposes a clustering method to extract multiple concepts from data, considering random and fuzzy uncertainty of data to make a more comprehensive clustering.

Moreover, the cloud model forms a transformation model between qualitative concepts and quantitative data through backward and forward cloud transformation algorithms (BCT and FCT) and parameterizes the concept into triples. To obtain better concepts, Wang et al. [59] proposed a multistep sampling approach based on previous BCTs [60–62] to learn the uncertainty of parameters to improve the accuracy of concept. However, the concept centers still do not describe the uncertainty of the whole concept. To build a better concept space, obtaining more stable concepts is essential. In this study, we consider the uncertainty of concept centers to build more stable concepts.

3. Cloud model-based clustering algorithm

Cloud-Cluster embeds clustering with random uncertainty and analyzes the changes in concepts in clustering. The framework of Cloud-Cluster is shown in Fig. 1. There are three parts: (1) initial concept generation, which initializes a confused concept space; (2) concept-based refinement aggregation, which repeatedly divides data under concepts with random uncertainty embedded and transforms data into new concepts to find the optimal concept space; and (3) concept uncertainty evaluation, which evaluates the uncertainty of concepts in the iterative process. The goal of Cloud-Cluster is that objects within concepts have a higher similarity, and objects between concepts have a lower similarity.

3.1. Initial concept generation

Let $X = \{x_i | i = 1, 2, \ldots, n\}$ be a dataset in a L-dimensional space D^L, and construct an initial concept space $S_C^{(0)}$ including c concepts where each concept C_k is described as triples $(Ex_k^{(0)}, En_k^{(0)}, He_k^{(0)})$. Expectation $Ex_k^{(0)}$ represents the basic deterministic measure of the initial concept C_k. Entropy $En_k^{(0)}$ represents the uncertainty measure of the initial concept, which is determined by the randomness and fuzziness of the concept. Hyper entropy $He_k^{(0)}$ represents the measure of uncertainty of the entropy $En_k^{(0)}$. Unlike extracting concepts from complex data in knowledge discovery, concept effectively integrate randomness and fuzziness. Furthermore, concepts can make objects in the same group more similar to each other than objects in other clusters and describe the overall quantitative nature of concepts through three numerical characteristics (Ex, En, He). Cloud-Cluster groups the dataset X under c concepts by embedding random uncertainty to them for a stable concept space S_C to discover more reasonable clusters $\pi_1, \pi_2, \ldots, \pi_c$. Among them, due to uncertainty, $C_{k'} \bigcap_{k' \neq k} C_k$ is not necessarily equal to zero.

In the real world, the initial perception of data is confusing. Thus, the concept space is randomly initialized to mimic the initial state. One can generate c initial concept centers $Ex_1^{(0)}$, $Ex_2^{(0)}$, $\ldots$, $Ex_c^{(0)}$ randomly within dataset X. Then, c concepts are created by the traditional Backward Cloud Transformation algorithm [59] with c concept centers and dataset X. Based on these, the initial concept space $S_C^{(0)}$ is constructed.

3.2. Concept-based refinement aggregation

3.2.1. Data partition under concepts

Cloud-Cluster embeds random uncertainty to make a more comprehensive clustering of data. To divide data under more accurate concepts, Cloud-Cluster adds randomness into the data partition to extend the range of data distribution, as shown in Fig. 2.

In Fig. 2, $f(x)$ represents the probability density of x, and $E(x)$ represents the expectation of x. A membership function μ is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1. Different from the Gaussian distribution, the Cloud model measures the randomness contained in the data through a membership function. The randomness is generated from *En* and *He* in the concept space, which is represented as Eq. (1).

$$E_n' = En + He * \varepsilon \tag{1}$$

where ε is randomly generated from $N \sim (0, 1)$. Then, the fuzziness of each data point x_i from dataset X in c concepts is calculated with randomness embedded, which is defined as the cluster concept membership degree in Definition 1.

Definition 1. Cluster Concept Uncertainty Degree μ (C^2UD)

$\mu_k(x_i)$ is the membership degree of data point x_i under cluster concept C_k, where x_i has $L-$ dimension features and C_k is a concept composed of the triples (Ex_k, En_k, He_k) in concept space S_C. $\mu_k(x_i) \in [0, 1]$ under concept C_k is a steady tendency of point with randomness: $\mu(x) : \tau \to [0, 1], \forall x \in \tau$.

$$\mu_k(x_i) = Exp\left(-\sum_{l=1}^{L} \frac{(x_{il} - Ex_{kl})^2}{2\left(En_{kl}'\right)^2}\right) \tag{2}$$

where L is the dimension of data point x_i, Ex_{kl} represents the lth concept center of concept C_k, and En_k' is generated as Eq. (1) from concept C_k.

In data partitioning, the cluster concept uncertainty degree can effectively evaluate the fuzziness of each data point x_i from dataset X in c concepts. Each data point has different uncertainty degrees of concepts. Then, uncertainty degrees are normalized to find a better data partition. The results of normalization are named the Concept Importance Degrees, which is defined in Definition 2.

Definition 2. Concept Importance Degree θ (CID)

$\theta_k(x_i)$ is the contribution of the data point x_i in the cluster concept C_k, where $\theta_k(x_i) \in [0, 1]$. When concepts exist in concept space S_C, $\sum_{k=1}^{c} \theta_k(x_i)$ is equal to one. The formula is given in Eq. (3).

$$\theta_k(x_i) = \frac{\mu_k(x_i)}{\sum_{j=1}^{N} \mu_k(x_j)} \tag{3}$$

where N is the number of datasets X and μ_k represents the uncertainty degree of data point x_i under concept C_k.

According to the principle of maximum membership, $\theta_k(x_i)$ of data point x_i is higher, and concept C_k is most likely to contain x_i. Under the concepts in concept space, data are divided into clusters $\pi_1, \pi_2, \ldots, \pi_c$.

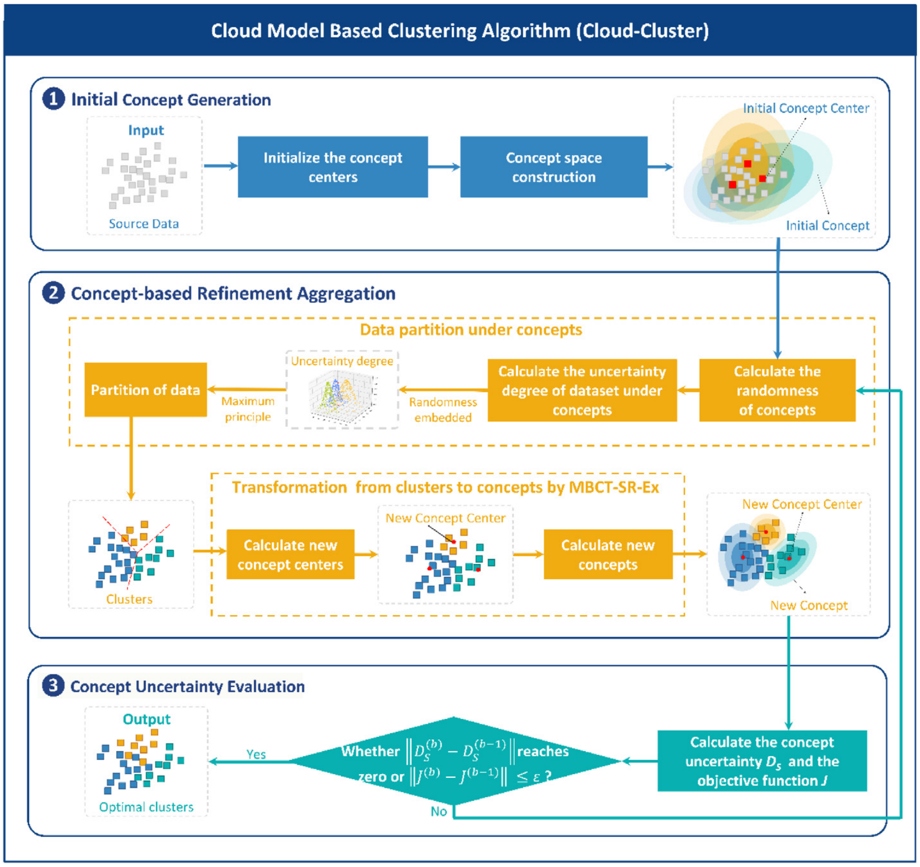

Cloud Model Based Clustering Algorithm (Cloud-Cluster)
1 Initial Concept Generation
Input
Source Data
Initialize the concept centers
Concept space construction
Initial Concept Center
Initial Concept
2 Concept-based Refinement Aggregation
Data partition under concepts
Partition of data
Maximum principle
Uncertainty degree
Randomness embedded
Calculate the uncertainty degree of dataset under concepts
Calculate the randomness of concepts
Transformation from clusters to concepts by MBCT-SR-Ex
Clusters
Calculate new concept centers
New Concept Center
Calculate new concepts
New Concept Center
New Concept
3 Concept Uncertainty Evaluation
Output
Optimal clusters
Yes
Whether $\|D_S^{(b)} - D_S^{(b-1)}\|$ reaches zero or $\|J^{(b)} - J^{(b-1)}\| \le \varepsilon$?
No
Calculate the concept uncertainty D_S and the objective function J

Fig. 1. Cloud-Cluster Framework.

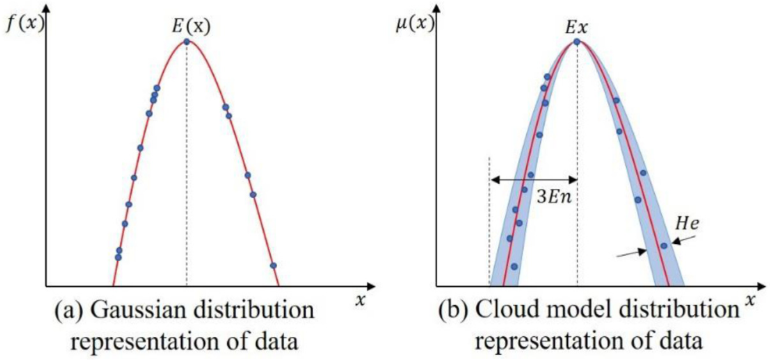


Fig. 2. Range of data distribution.

3.2.2. Transformation from data to concepts

After obtaining clusters of data, Cloud-Cluster needs to transform clusters into new concepts. The cloud model theory uses the forward cloud transformation (FCT) [59] and the backward cloud transformation (BCT) [59] to implement the two-way concept cognitive transformations between the data and quantitative description. FCT is used to implement the transformation from intension to extension of a concept while BCT is a method to transform data into concepts, where the concept is described as a triple (Ex, En, He). A suitable BCT can greatly reduce the calculation errors of parameters to obtain a precise concept. To transform each cluster into a more stable concept, an improved multistep backward cloud transformation algorithm (MBCT-SR-Ex) is proposed. It employs the concept uncertainty degree of data points as an important parameter in computing Ex on the basis of the traditional multistep backward cloud transformation algorithm (MBCT-SR) [59]. Different from MBCT-SR, MBCT-SR-Ex calculates more accurate concept centers by adding different calculation methods. Furthermore, Ex of MBCT-SR-Ex is a weighted average of the weights of clustering to concepts in each iteration.

In MBCT-SR-Ex, new clusters and C^2UD are used as parameters to calculate more accurate concept centers. Then, according to the concept importance degree (CID) of data points in cluster π_k, the concept center Ex_k can be calculated as Eq. (4).

$$Ex_k = \sum_{i=1}^{n_{\pi_k}} x_i \theta_k\left(x_i\right) \tag{4}$$

where $\theta_k(x_i)$ is the concept importance degree of point x_i under concept C_k and n_{π_k} is the data point number of cluster k.

Data points in each concept are presented in X_{π_k}. MBCT-SR-Ex samples the data from dataset X in M times, each time sampling

```
Algorithm 1:   MBCT-SR-Ex
Input: Dataset $X$ and cluster concept uncertainty degrees $U = \{\mu_i | i = 1,2,\ldots,n\}$, in which $\mu_i = \{\mu_{k_i} | k = 1,2,\ldots,c\}$
Output: Concepts $C = \{C_k | k = 1,2,\ldots,c\}$
compute CID $\theta$ using Eq. (3);
for $k \leftarrow 1$ to $c$ do
    compute concept centers $Ex_k$ using Eq. (4);
    select $R$ data points as $D_i$ for $M$ times;
    for each $D_i$ do
        calculate the mean and variance of each $D_i$ using Eq. (5) and Eq. (6);
    end for
    compute concept estimators $En_k$ and $He_k$ using Eq. (7) and Eq. (8);
end for
```

R data points. The mean and variance of each group are calculated as in Eqs. (5) and (6), respectively.

$$Ex_{km} = \frac{1}{R} \sum_{r=1}^{R} x_r \theta_k \left(x_r \right) \tag{5}$$

$$Y_{km}^2 = \frac{1}{R-1} \sum_{r=1}^{R} \left(x_r - Ex_{km} \right)^2 \tag{6}$$

where x_r is the r^{th} data point in the m^{th} group and the number of data in each group is R. Ex_{k_m} is the concept center of the m^{th} group under Ex_k. According to Eq. (6), the concept parameters En_k and He_k are calculated as in Eqs. (7) and (8), respectively.

$$En_k = \sqrt{\frac{1}{2}\sqrt{4\left(\widehat{EY}_k^2\right)^2 - 2\widehat{DY}_k^2}} \tag{7}$$

$$He_k = \sqrt{\widehat{EY}_k^2 - En_k^2} \tag{8}$$

where $\widehat{EY}_k^2$ and $\widehat{DY}_k^2$ are the mean and variance of Y_k^2, respectively, and Y_k^2 includes $\{Y_{k_1}^2, Y_{k_2}^2, \ldots, Y_{k_M}^2\}$. The pseudocode of MBCT-SR-Ex is shown in Algorithm 1. The step of computing CID has a complexity of $O(1)$. The step of computing concept center Ex_k and selecting data points requires $O(n)$ operations. The step of computing concepts En_k and He_k requires $O(1)$ operations. Thus, the time complexity of Algorithm 1 is $O(n)$.

Ex_k, En_k, and He_k of cluster k are calculated by MBCT-SR-Ex to create concept C_k. En_k reflects the discreteness of points in concept C_k. He_k reflects the aggregation degree of points in concept C_k. When obtaining a new concept center $Ex_k^{(b)}$, MBCT-SR-Ex generates a new concept $C_k^{(b)} = (Ex_k^{(b)}, En_k^{(b)}, He_k^{(b)})$ from cluster k, in which b represents the iteration number. According to new concepts, the new concept space $S_C^{(b)}$ is constructed.

In Cloud-Cluster, Concept-Based Refinement Aggregation simultaneously characterizes the fuzziness and randomness of objects to reserve the uncertain information making clustering with more refined concepts, where the function can be composed of dataset X and concepts $C = \{C_1, C_2, \ldots, C_c\}$ as in Eq. (9).

$$J_{Cloud-Cluster} = \sum_{i=1}^{n} \log \sum_{k=1}^{c} P\left(x_i | C_k\right) \tag{9}$$

where $P(x_i|C_k)$ represents the probability that x_i belongs to the concept C_k, and is defined as Eq. (10).

$$P\left(x_i | C_k\right) = \frac{P\left(x_i, C_k\right)}{P\left(C_k\right)} \tag{10}$$

where $P(C_k)$ represents the probability that the concept center belongs to concept C_k. $P(x_i, C_k)$ represents the probability that a point in the concept space belongs to a concept C_k.

3.3. Concept uncertainty evaluation

3.3.1. Concept uncertainty calculation

Cloud-Cluster clusters data into concept with random uncertainty embedded. The initial construction of concept space simulates confuses data information for the objective world. In the iterative process, Cloud-Cluster transforms new concepts from clusters to construct a new concept space. The process of learning a new concept space simulates the process of knowledge learning. In Cloud-Cluster, it is important to reflect whether the concepts have uncertainty, ambiguity, diversity, and instability in the clustering process.

To evaluate the uncertainty, the overall distribution of the concept of each cluster is drawn from the internal and external envelope curves. The KL-divergence is employed to define the Cluster Concept Drift Degree, which is a measure of the external curve of the concept distribution and is defined in Definition 3.

Definition 3. Cluster Concept Drift Degree $D_J(C^2D^2)$

Consider a dataset X with c concepts, where the parameters of concept C_k are (Ex_k, En_k, He_k). D_J is the KL-divergence between the extended curve concept distribution of the current and previous cluster. The formula is given in Eq. (11).

$$\begin{aligned}\sum_{k=1}^{c} D_J(C_k^{(b)}||C_k^{(b-1)}) &= \sum_{k=1}^{c}(D_{KL}(C_k^{(b)}||C_k^{(b-1)}) + D_{KL}(C_k^{(b-1)}||C_k^{(b)})) \\ &= \frac{1}{2}\sum_{k=1}^{c}\left(Ex_k^{(b)} - Ex_k^{(b-1)}\right)^2\left(\frac{1}{\sigma_k^{(b)2}} + \frac{1}{\sigma_k^{(b-1)2}}\right) \\ &\quad + \frac{1}{2}\sum_{k=1}^{c}\left(\frac{\sigma_k^{(b-1)2}}{\sigma_k^{(b)2}} + \frac{\sigma_k^{(b)2}}{\sigma_k^{(b-1)2}}\right) - L\end{aligned} \tag{11}$$

where $C_k^{(b)}$ is the bth concept distribution and $C_k^{(b-1)}$ is the $(b-1)$th concept distribution of concept C_k, where $\sigma^{(b)}$ is equal to $En^{(b)} + 3He^{(b)}$ and $\sigma^{(b-1)}$ is equal to $En^{(b-1)} + 3He^{(b-1)}$. L is the dimension of the concept. According to Eq. (10), the larger the change between concepts before and after is, the greater the fluctuation of the drift degree becomes. When $\left\|J^{(b)} - J^{(b-1)}\right\| < \varepsilon_1$, the clustering process is stable. If the drift degree between the new and previous concepts fluctuates greatly, it explains why the uncertainty of the dataset is strong.

3.3.2. Convergence of cluster concept drift degree

Concepts are the main unit of Cloud-Cluster. Each concept C_k contains triples (Ex_k, En_k, He_k). During the clustering process, the purpose is to find the optimal concepts. Thus, we leverage the probability distribution to find the optimal concept estimators. To simplify the calculation, log-likelihood is introduced and the whole formula is given in Eq. (12).

$$\theta = argmax \sum_{i=1}^{n} \log P(x_i; \theta) \tag{12}$$

where $P\left(x_i;\theta\right) = \sum_{k=1}^{c} P(x_i;\theta_k)$ and $\theta = \{\theta_k | k = 1, 2, \ldots, c\}$. According to the convergence of EM algorithm in Eq. (13), whether the concepts in Cloud-Cluster can be finally stable needs to be proven, which means that C^2D^2 should converge to the minimum.

$$\begin{aligned} L\left(\theta^{(b+1)}\right) &= \sum_{i=1}^{n} \log \sum_{i} Q_i^{(b)}(z_i) \frac{P\left(x_i.z_i;\theta^{(b+1)}\right)}{Q_i^{(b)}(z_i)} \\ &\geq \sum_{i=1}^{n} \log \sum_{i} Q_i^{(b)}\left(z_i\right) \frac{P\left(x_i.z_i;\theta^{(b)}\right)}{Q_i^{(b)}\left(z_i\right)} \end{aligned} \tag{13}$$

where z_i is the latent variable and $Q_i(z)$ is the approximate posterior distribution of latent variable z for sample i. For Eq. (13), $L(\theta)$ is monotonically increasing under iteration and has an upper bound. In Cloud-Cluster, according to the properties of KL divergence, we know that $D_J(C^{(b)}||C^{(b+1)}) \geq 0$. Thus, the details of $D_J(C^{(b)}||C^{(b+1)})$ are decomposed into Eq. (14).

$$\begin{aligned} \sum_{i=1}^{c} D_J\left(C^{(b)}||C^{(b+1)}\right) = &\sum_{i=1}^{n} P\left(x;\theta^{(b)}\right) log \frac{P\left(x;\theta^{(b)}\right)}{P\left(x;\theta^{(b+1)}\right)} \\ &+ \sum_{i=1}^{n} P\left(x;\theta^{(b+1)}\right) log \frac{P\left(x;\theta^{(b+1)}\right)}{P\left(x;\theta^{(b)}\right)} \end{aligned} \tag{14}$$

Since $\sum_{i=1}^{n} logP(x_i;\theta^{(b+1)}) \geq \sum_{i=1}^{n} logP(x_i;\theta^{(b)}) \geq \sum_{i=1}^{n} logP(x_i;\theta^{(b-1)})$, Eq. (14) converges when $P(x;\theta^{(b)}) \approx P(x;\theta^{(b+1)})$.

When the objective function is minimized, the whole process is stable and the algorithm converges. The mapping between clusters and concepts obtained by cloud transformation forms a clustering process from data to concepts. The overall description of Cloud-Cluster is shown in Algorithm 2. Cloud-Cluster as a soft algorithm consists of three main parts: calculating the concept centers, finding the data uncertainty degree, and generating new concepts. We observe that the steps of calculating concept (Lines 3–5) are not at all time-consuming and can be carried out in $O(1)$ operations. The steps of calculating data uncertainty degree and new concepts (Lines 6–8) are decided by the time complexity of Algorithm MBCT-SR-Ex which is $O(cn)$. There are b iterations from the start to convergence. Thus, the time complexity and space cost are $O(ncb)$ and $O(n)$ respectively.

4. Experiment

In this section, we evaluate the clustering performance of Cloud-Cluster and discuss the performance impact of the improved backward cloud transformation algorithm on Cloud-Cluster. In addition, we analyze the trends of uncertainty of concepts in the clustering process for different datasets and give a quantitative representation of the concepts for some datasets.

4.1. Experimental dataset

To verify the validity of Cloud-Cluster, we used 29 datasets from clustering tasks in UCI and OpenML repositories with no more than 4500 points and 70 features. These datasets are divided into two parts: data with clear and unclear concepts, which are shown as Fig. 3. More information on these datasets is given in Tables 1 and 2. Furthermore, to verify the performance of Cloud-Cluster on high-dimensional datasets, we used the 5 public high-dimensional datasets listed in Table 3. The preprocessing tasks are applied to all datasets: removing missing values and one-hot encoding for categorical features. To fully show the real information of the features in datasets, all datasets are not standardized in our experiment.

Table 1
Summary of clustering datasets with unclear concepts.

No.	Dataset name	Classes	Data points	Dimensions
1	abalone	3	4177	8
2	analcatdata_authorship	4	841	70
3	Australian	2	690	14
4	autoPrice	2	159	15
5	Banknote-authentication	2	1372	4
6	bodyfat	2	252	14
7	chatfield_4	2	235	11
8	cleveland	2	297	13
9	dermatology	2	358	34
10	glass	6	214	9
11	iris	3	150	4
12	lowbwt	2	189	9
13	new-thyroid	3	215	5
14	Pima	2	768	8
15	prnn_synth	2	250	2
16	seeds	3	210	7
17	Statlog-heart	2	270	13
18	strikes	2	625	6
19	transplant	2	131	3
20	vertebra-column	3	310	6
21	wdbc	2	569	30
22	wine	3	178	13

Table 2
Summary of clustering datasets with clear concepts.

No.	Dataset name	Classes	Data points	Dimensions
1	Aggregation	7	788	2
2	Chainlink	2	1000	3
3	Compound	6	399	2
4	Flame	2	240	2
5	Pathbased	3	300	2
6	Spiral	3	312	2
7	Target	6	770	2

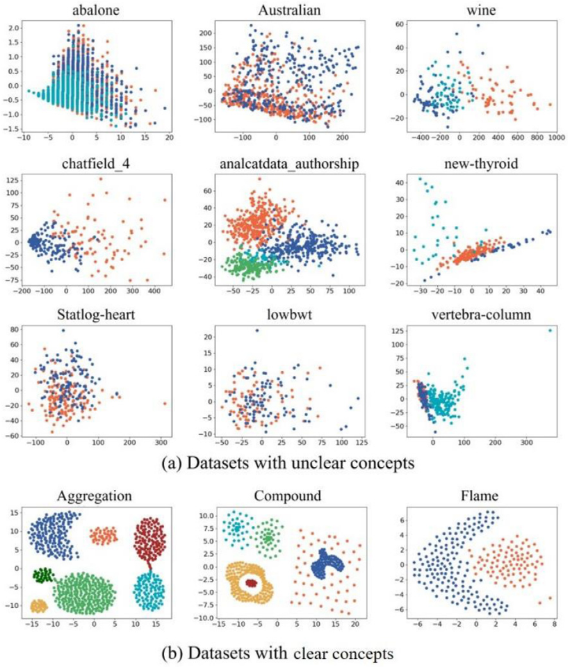
Table 3
Summary of high-dimensional datasets.

No.	Dataset name	Classes	Data points	Dimensions
1	FCC solute diffusion	5	218	111
2	Perovskite conductivity	5	7230	112
3	Crystal compound formation energy	5	4000	144
4	Fashion MNIST	10	70000	784
5	MNIST	10	70000	784

4.2. Experimental setup

To evaluate the clustering performance of Cloud-Cluster, in this basic experiment, we compare Cloud-Cluster with K-Means [63] and uncertainty theory-based algorithms Fuzzy C-Means [38], Gaussian Mixture Model [22], Rough K-Means [53], Possibilistic Fuzzy C-Means [45], Density peaks clustering [16], Kernel Fuzzy C-Means [18], Identifying Density peaks clustering [17] and Density Peaks Clustering Based on k Nearest Neighbors [19]. These nine algorithms are hard and soft clustering algorithms tuning the best number of clusters under SC, CHI, and DBI metrics. KM, FCM, GMM, RKM, PFCM, DPC, KFCM, IDPC, DPC-KNN are short for

```
Algorithm 2: Cloud-Cluster
Input: Data points $X = \{x_1, x_2, \dots, x_n\}$
Output: Optimal clustering results $\pi_1, \pi_2, \dots, \pi_c$
initialize concept space $S_C^{(0)}$, including $c$ concepts and $b = 0$;
while $\|J^{(b)} - J^{(b-1)}\| \geq \varepsilon_1$ or $\|D_J^{(b)}\| \geq \varepsilon_2$ do
    calculate Cluster Concept Uncertainty Degree $\mu(x)$ using Eq. (2);
    calculate Concept Importance Degree $\theta(x)$ using Eq. (3);
    divide data into clusters $\{\pi_1, \pi_2, \dots, \pi_c\}$ according to $\theta(x)$;
    for $k \leftarrow 0$ to $c$ do
        create new concept $C_k$ to construct concept space $S_C^{(b)}$ which is represented as $Ex_k^{(b)}$, $En_k^{(b)}$, $He_k^{(b)}$ by MBCT-SR-Ex;
    end
    $b = b + 1$
    calculate the objective function $J^{(b)}$ using Eq. (9);
    calculate the sum of Cluster Concept Drift Degree $D_J$ of cluster concept $C_k^{(b)}$ and $C_k^{(b-1)}$ #compute as Eq. (11);
end
```



(a) Datasets with unclear concepts

(b) Datasets with clear concepts

Fig. 3. Visualization results of two types of datasets using PCA for dimensionality reduction.

K-Means, Fuzzy C-Means, Gaussian Mixture, Rough K-Means, Possibilistic Fuzzy C-Means, Density peaks clustering, Kernel Fuzzy C-Means, Identifying Density peaks clustering and Density Peaks Clustering Based on k Nearest Neighbors, respectively. As a soft algorithm, the output of Cloud-Cluster is a soft partition. Cloud-Cluster describes the concept through three numerical characteristics Ex_k, En_k, and He_k, where En_k as a measurement of randomness reflects the dispersive extent of the clustering that represents a given qualitative concept. In the interaction, the randomness of Cloud-Cluster is reflected from the concept uncertainty evaluation by the cluster concept uncertainty degree and Cluster Concept Drift Degree. In the experiment, parameters are randomly initialized for all algorithms. The parameter p of DPC or DPC-KNN is 0.5%, 1%, or 2%. In DPC, the parameter d_c is selected from 0.1%, 0.2%, 0.5%, 1% and 2%.

For clustering, good algorithms discover more high-quality data partitions. Clustering validity Indexes (CVIs) are used to measure the quality of clustering results. We use internal CVIs which include Silhouette Coefficient (SC), Calinski–Harabaz Index (CHI), and Davies–Bouldin Index (DBI) to measure the quality of clustering partitions. Then, external CVIs which include Cluster Accuracy (ACC), Fowlkes and Mallows Index (FMI), Normalized Mutual Information (NMI), and Adjusted Rand index (ARI) are used to quantify the quality of a clustering. Our programs were implemented in Python. The experiments were conducted on a computer with an Intel Core i7-9700F 3.00 GHz CPU. All programs ran in memory.

For each dataset, the number of clusters C is unknown which is determined based on the affiliated Silhouette Coefficient (SC). For all algorithms, the number of clusters was searched between 2 and 10. All algorithms were run 10 times to choose the best cluster number calculating the ACC, FMI, NMI, ARI for each algorithm.

The internal indices mean that no data labels are needed, and only the clustering effect is considered. Thus, this paper evaluates the algorithm performance of learning internal information of datasets. The results of learning the internal information of the data corresponding to the clustering results are compared with those corresponding to the original labels, which is defined as internal information disparity (IID) as shown in Eq. (15).

$$IID = \left| G(data, label_{true}) - G(data, label_{predict}) \right| \tag{15}$$

where G is the internal CVIs, data and labels are the input for G to measure the quality of clustering results. $label_{true}$ represents the true labels for data and $label_{predict}$ represents the clustering results from algorithms. A smaller IID value indicates that the clustering results are more similar to the information within the data under the original labels.

4.3. Performance of cloud-cluster

4.3.1. Performance on datasets with unclear concepts

Table 4 shows the clustering performance under the ACC metric with the SC metric as the selection method for c. For these compared algorithms, if the performance is highest, the

Table 4
Clustering performance of algorithms under ACC metric for datasets with unclear border (SC metric as selection method for c).

No.	KM	FCM	PFCM	RKM	GMM	KFCM	DPC	DPC-KNN	IDPC	Cloud-Cluster
1	0.407±0.011	0.44±0.0	0.441±0.0	0.479±0.03	0.51±0.012	0.514±0.0	0.491±0.0	0.4512±0.0	0.4895±0.0	**0.529±0.0**
2	0.647±0.001	0.654±0.0	0.646±0.003	0.649±0.001	0.501±0.071	**0.7419±0.0**	0.7051±0.0	0.5552±0.0	0.5552±0.0	0.716±0.001
3	0.584±0.001	0.584±0.0	0.581±0.0	0.577±0.004	0.661±0.067	0.5753±0.0	0.5521±0.0	0.6086±0.0	0.6507±0.0	**0.851±0.002**
4	0.556±0.003	0.56±0.0	0.648±0.0	0.62±0.091	0.673±0.089	0.7242±0.13	0.8605±0.0	0.6477±0.0	0.6729±0.0	**0.874±0.005**
5	0.612±0.001	0.609±0.0	0.6±0.0	0.61±0.006	0.586±0.11	0.61±0.0	0.7419±0.0	0.5983±0.0	**0.8965±0.0**	0.574±0.003
6	0.734±0.0	0.726±0.0	0.714±0.0	0.714±0.0	0.518±0.015	0.7222±0.0	0.7341±0.0	**0.742±0.0**	0.5277±0.0	0.732±0.005
7	0.576±0.0	0.582±0.0	0.586±0.0	0.857±0.018	0.555±0.083	**0.8595±0.0**	0.5489±0.0	0.6±0.0	0.6085±0.0	0.823±0.005
8	0.863±0.002	0.847±0.0	**0.868±0.0**	0.588±0.002	0.815±0.004	0.5892±0.0	0.5185±0.0	0.6531±0.0	0.5959±0.0	0.86±0.004
9	0.567±0.0	0.564±0.0	0.561±0.0	0.567±0.0	0.543±0.03	0.5642±0.0	0.5502±0.0	0.5223±0.0	0.5223±0.0	**0.899±0.196**
10	0.5±0.0	0.5±0.0	0.436±0.021	0.439±0.005	0.446±0.008	0.4906±0.0	0.4345±0.0	**0.5186±0.0**	0.4719±0.0	0.473±0.002
11	0.667±0.0	0.667±0.0	0.667±0.0	0.667±0.0	0.607±0.021	**0.8933±0.0**	0.8533±0.0	0.8666±0.0	0.6666±0.0	0.667±0.0
12	0.529±0.0	0.529±0.0	0.545±0.0	0.525±0.002	0.565±0.029	0.5291±0.0	0.6084±0.0	0.5026±0.0	0.5396±0.0	**0.674±0.02**
13	0.846±0.03	0.791±0.0	0.488±0.0	0.68±0.122	0.806±0.008	0.7813±0.0	0.4372±0.0	0.7209±0.0	0.6511±0.0	**0.854±0.061**
14	0.66±0.001	0.659±0.0	0.624±0.0	0.616±0.042	0.577±0.058	0.6171±0.0	0.5169±0.0	0.5781±0.0	0.5169±0.0	**0.695±0.003**
15	0.528±0.0	0.544±0.0	0.539±0.023	0.528±0.001	0.68±0.048	0.544±0.0	0.504±0.0	0.704±0.0	0.504±0.0	**0.774±0.008**
16	0.655±0.003	0.662±0.0	0.666±0.001	**0.667±0.0**	**0.667±0.0**	0.6619±0.0	0.6476±0.0	**0.667±0.0**	**0.667±0.0**	**0.667±0.0**
17	0.59±0.002	0.593±0.0	0.604±0.0	0.584±0.021	0.547±0.047	0.6074±0.0	0.5777±0.0	0.6111±0.0	0.5518±0.0	**0.819±0.01**
18	0.526±0.001	0.534±0.0	0.534±0.0	0.515±0.01	0.607±0.005	**0.667±0.0**	0.5648±0.0	0.5248±0.0	0.648±0.0	0.524±0.006
19	0.87±0.0	0.87±0.0	0.87±0.0	0.86±0.004	0.885±0.0	0.8702±0.0	0.7786±0.0	0.6335±0.0	0.6335±0.0	**0.94±0.008**
20	0.667±0.003	0.677±0.0	0.71±0.0	0.662±0.072	0.524±0.004	0.5161±0.0	0.758±0.0	0.6225±0.0	**0.7838±0.0**	0.662±0.004
21	0.854±0.0	0.854±0.0	0.868±0.0	0.859±0.0	0.877±0.074	0.7985±0.04	0.5975±0.0	0.5852±0.0	0.8752±0.0	**0.923±0.005**
22	0.657±0.0	0.657±0.0	0.674±0.0	0.68±0.0	0.595±0.022	0.674±0.0	**0.7191±0.0**	0.6404±0.0	0.6573±0.0	0.601±0.0
Avg rank	5.5	5.41	5.27	6.272	6.091	4.455	6.273	5.91	5.72	2.955

corresponding entries are bolded, and if the performance is the result of the known clusters on the dataset, the corresponding entries are underlined. Cloud-Cluster obtains the highest ACC under SC on 12/22 datasets (bolded), while no dataset has significantly worse performance. Cloud-Cluster obtains the highest average rank on the unclear datasets. The average clustering performance of Cloud-Cluster under the ACC metric was an improvement of 14.4%, 14.4%, 16%, 15.6% and 17.3%, compared to KM and uncertainty theory-based methods FCM, PFCM, RKM and GMM, respectively. However, other compared methods can only perform well on a few datasets, and they all perform significantly worse on the other datasets. KM adopts hard division which does not consider the membership of data points to different categories, and therefore does not learn well for overlapping categories. From Table 4, KM has 2 bold entries. For example, on the *Australian* (No.3) and *autoPrice* (No. 4) datasets, our results show a huge improvement compared to K-Means. FCM and PFCM based on fuzzy sets, both consider the fuzzy uncertainty of data. PFCM even considers the situation in which some data contain both membership and typicality values on the basis of FCM. However, their demarcation of data boundaries is still unclear and artificial parameter choices exist. It can be seen from the experiment that under the ACC metric, there are no bold entries for FCM and only one for PFCM. RKM based on rough sets is another method to learn the fuzzy information of data. This algorithm looks at the range of each feature to find features that better represent the data to obtain concepts, ignoring features with fuzziness and the connotations of the data. Thus, its clustering performance is poor on datasets with unclear concepts. For GMM, this algorithm transforms the data into different normal distributions, and combines fuzzy probabilities for data clustering, in which the boundary division of each distribution is relatively clear and ignores the range of distribution representation. It can be seen from Table 4 that only the GMM of the *glass* (No. 10) dataset is marked in bold. KFCM outperforms FCM and DPC such as in the *analcatdata authorship* (No. 2) and *chatfield 4* (No. 4) datasets. DPC only requires considering the distance between all the pairs of data points. Therefore, the performance of DPC is affected by

Table 5
Clustering performance of algorithms under NMI metric for datasets with unclear border (SC metric as selection method for c).

No.	KM	FCM	PFCM	RKM	GMM	KFCM	DPC	DPC-KNN	IDPC	Cloud-Cluster
1	0.053±0.007	0.077±0.0	0.073±0.0	0.11±0.035	0.151±0.015	0.129±0.0	0.149±0.0	0.118±0.0	0.112±0.0	**0.175±0.001**
2	0.378±0.001	0.393±0.0	0.378±0.002	0.33±0.002	0.174±0.124	0.486±0.0	**0.639±0.0**	0.283±0.0	0.283±0.0	0.509±0.006
3	0.017±0.0	0.018±0.0	0.017±0.0	0.022±0.002	0.136±0.079	0.014±0.0	0.002±0.0	0.032±0.0	0.082±0.0	**0.397±0.006**
4	0.459±0.006	0.49±0.0	0.49±0.0	0.428±0.127	0.101±0.088	0.269±0.2	0.491±0.0	0.014±0.0	0.036±0.0	**0.494±0.017**
5	0.03±0.0	0.029±0.0	0.026±0.0	0.047±0.005	0.07±0.167	0.029±0.0	0.347±0.0	0.021±0.0	**0.612±0.0**	0.016±0.001
6	0.192±0.0	0.168±0.0	0.141±0.0	0.18±0.0	0.008±0.01	0.161±0.0	0.180±0.0	**0.210±0.0**	0.015±0.0	0.171±0.009
7	0.014±0.0	0.017±0.0	0.02±0.0	**0.51±0.05**	0.029±0.088	0.403 ±0.0	0.065 ±0.0	0.006 ±0.0	0.011±0.0	0.332±0.014
8	0.409±0.005	0.37±0.0	**0.423±0.0**	0.028±0.002	0.329±0.007	0.020±0.0	0.002±0.0	0.103 ±0.0	0.0263±0.0	0.402±0.009
9	0.008±0.0	0.007±0.0	0.007±0.0	0.014±0.0	0.112±0.039	0.007±0.0	0.005±0.0	0.019±0.0	0.001±0.0	**0.812±0.332**
10	0.362±0.0	**0.38±0.0**	0.3±0.072	0.186±0.011	0.292±0.047	0.359±0.0	0.256±0.0	0.345±0.0	0.259±0.0	0.323±0.009
11	0.657±0.0	0.657±0.0	0.64±0.0	0.54±0.0	0.406±0.11	0.749±0.0	**0.754±0.0**	0.696±0.0	0.478±0.0	0.734±0.0
12	0.008±0.0	0.008±0.0	0.012±0.0	−0.002±0.001	0.018±0.015	0.008±0.0	0.037±0.0	0.016 ±0.0	0.021 ±0.0	**0.09±0.021**
13	0.477±0.036	0.343±0.0	0.116±0.0	0.266±0.165	0.43±0.047	0.3363±0.0	0.1285±0.0	0.3033±0.0	0.041±0.0	**0.554±0.14**
14	0.029±0.0	0.034±0.0	0.02±0.0	0.034±0.003	0.016±0.019	0.0176±0.0	0.0008±0.0	0.0016±0.0	0.001±0.0	**0.12±0.004**
15	0.002±0.0	0.006±0.0	0.006±0.007	−0.001±0.0	0.17±0.099	0.0056±0.0	0.0004±0.0	0.2248±0.0	0.0004±0.0	**0.239±0.015**
16	0.534±0.006	0.536±0.0	0.531±0.014	0.472±0.006	0.53±0.039	0.536±0.0	0.565±0.0	0.529±0.0	0.517±0.0	**0.594±0.007**
17	0.019±0.001	0.021±0.0	0.029±0.0	0.024±0.015	0.011±0.033	0.0303±0.0	0.031±0.0	0.0479±0.0	0.0061±0.0	**0.315±0.023**
18	0.013±0.0	0.014±0.0	0.03±0.0	0.001±0.001	0.039±0.008	**0.0981±0.0**	0.0311±0.0	0.011±0.0	0.0862±0.0	0.002±0.001
19	0.551±0.0	0.551±0.0	0.551±0.0	0.514±0.011	0.521±0.015	0.5508±0.0	0.401±0.0	0.240±0.0	0.241±0.0	**0.671±0.028**
20	0.407±0.005	0.416±0.0	0.468±0.0	0.398±0.069	0.179±0.004	0.420±0.0	0.494±0.0	0.219±0.0	**0.621±0.0**	0.294±0.007
21	0.465±0.0	0.465±0.0	0.498±0.0	0.507±0.0	0.479±0.192	0.3651±0.01	0.1835±0.0	0.1728±0.0	0.5062±0.0	**0.596±0.02**
22	0.426±0.0	0.407±0.0	0.417±0.0	0.42±0.0	0.453±0.092	0.434±0.0	0.391±0.0	0.397±0.0	0.408±0.0	**0.467±0.001**
Avg rank	5.318	5.273	5.727	6.318	5.5	5.41	5.55	6.318	6.772	2.82

uncertain information in the judgment of data partitioning. IDPC has a better performance compared to DPC and DPC-KNN in unclear datasets. Specifically, Cloud-Cluster has a greater advantage over IDPC in datasets with unclear concepts. It is obvious that Cloud-Cluster has achieved gratifying results in most datasets.

Tables 5–7 show the clustering performance of the NMI, ARI, and FMI metrics with the SC metric as the selection method for c. Cloud-Cluster has stable clustering performance for these datasets under multiple metrics. Cloud-Cluster achieves the highest NMI, ARI, and FMI performances on No. 12, 11, and 10 clustering datasets, respectively, in which the average rank is first. For datasets such as *Australian* (No. 3), *autoPrice* (No. 4), *Statlog-heart* (No. 17), and *wdbc* (No. 21), our proposed method achieves the best performance. From the results, it can be also seen that in most cases Cloud-Cluster outperformed the other aforementioned methods. Specifically, Cloud-Cluster can handle the uncertainty of datasets in the clustering process fine, as well as being robust in unclear datasets of clusters. The boundaries between the categories of these datasets are not clear, which means that concepts corresponding to categories are fuzzy. Cloud-Cluster introduces randomness to expand the representation range of the category distribution to better describe the boundaries. When the KM, FCM, PFCM, RKM, GMM, KFCM, DPC, DPC-KNN, IDPC and Cloud-Cluster algorithms select the optimal number of clusters under the SC, CHI, and DBI, respectively, the average rank corresponding to multiple external indicators is shown in Fig. 4. The average rank of Cloud-Cluster is better than the other 9 compared methods and obtains 1st place.

Furthermore, to prove the effectiveness of Cloud-Cluster, we considered the internal information of datasets. Tables 8 and 9 show the *IID* values of datasets with internal validation indices SC and CHI. It is observed that Cloud-Cluster has good performance of SC and CHI close to information of raw data on most (10/22) and (10/22) datasets, respectively. It is also seen that learning the data distribution of datasets can describe more information in data from GMM and Cloud-Cluster. In addition to the SC metric, this experiment also uses the CHI and DBI metrics to select the optimal number of clusters.

Table 6
Clustering performance of algorithms under ARI metric for datasets with unclear border (SC metric as selection method for *c*).

No.	KM	FCM	PFCM	RKM	GMM	KFCM	DPC	DPC-KNN	IDPC	Cloud-Cluster
1	0.031±0.01	0.06±0.0	0.062±0.0	0.106±0.025	0.158±0.017	0.133±0.0	0.135±0.0	0.104±0.0	0.122±0.0	**0.19**±**0.001**
2	0.319±0.001	0.38±0.0	0.371±0.004	0.381±0.001	0.144±0.118	0.446±0.0	0.53±0.0	0.135±0.0	0.134±0.0	**0.517**±**0.004**
3	0.026±0.0	0.027±0.0	0.025±0.0	0.014±0.001	0.173±0.11	0.021±0.0	0.006±0.0	0.041±0.0	0.085±0.0	**0.491**±**0.006**
4	0.389±0.004	0.41±0.0	0.44±0.0	0.471±0.07	0.141±0.127	0.272±0.26	0.544±0.0	−0.012±0.0	0.083±0.0	**0.555**±**0.016**
5	0.049±0.001	0.046±0.0	0.039±0.0	0.029±0.003	0.077±0.192	0.047±0.0	0.232±0.0	0.035±0.0	**0.629**±**0.0**	0.021±0.001
6	0.216±0.0	0.202±0.0	0.18±0.0	0.142±0.003	0.001±0.003	0.194±0.0	0.216±0.0	**0.231**±**0.0**	0.002±0.0	0.213±0.009
7	0.019±0.0	0.024±0.0	0.026±0.0	0.404±0.032	0.035±0.111	**0.514**±**0.0**	−0.026±0.0	−0.003±0.0	0.004±0.0	0.415±0.014
8	0.524±0.005	0.477±0.0	**0.539**±**0.0**	0.02±0.001	0.395±0.01	0.028±0.0	−0.002±0.0	0.091±0.0	0.033±0.0	0.516±0.011
9	0.014±0.0	0.013±0.0	0.011±0.0	0.008±0.0	−0.039±0.019	0.013±0.0	0.007±0.0	−0.009±0.0	−0.001±0.0	**0.779**±**0.419**
10	0.226±0.0	0.235±0.0	0.211±0.037	**0.261**±**0.017**	0.214±0.021	0.213±0.0	0.122±0.0	0.191±0.0	0.158±0.0	0.201±0.01
11	0.54±0.0	0.54±0.0	0.531±0.0	0.657±0.0	0.366±0.068	**0.729**±**0.0**	0.663±0.0	0.676±0.0	0.424±0.0	0.568±0.0
12	−0.001±0.0	−0.001±0.0	0.003±0.0	0.007±0.001	0.017±0.017	−0.001±0.0	0.042±0.0	−0.003±0.0	0.002±0.0	**0.117**±**0.027**
13	0.548±0.062	0.441±0.0	0.044±0.0	0.256±0.087	0.472±0.052	0.422±0.0	0.011±0.0	0.297±0.0	0.016±0.0	**0.575**±**0.141**
14	0.074±0.001	0.08±0.0	0.05±0.0	0.013±0.001	0.03±0.036	0.045±0.0	−0.039±0.0	0.0099±0.0	−0.039±0.0	**0.151**±**0.005**
15	−0.001±0.0	0.004±0.0	0.004±0.01	0.002±0.0	0.135±0.065	0.004±0.0	−0.004±0.0	0.164±0.0	−0.004±0.0	**0.298**±**0.018**
16	0.466±0.004	0.468±0.0	0.456±0.009	**0.551**±**0.008**	0.452±0.028	0.468±0.0	0.476±0.0	0.453±0.0	0.489±0.0	0.504±0.005
17	0.028±0.001	0.031±0.0	0.039±0.0	0.018±0.009	0.01±0.043	0.042±0.0	0.019±0.0	0.046±0.0	−0.001±0.0	**0.406**±**0.026**
18	0.002±0.0	0.004±0.0	0.015±0.0	0.008±0.006	0.044±0.004	**0.123**±**0.0**	0.016±0.0	0.002±0.0	0.086±0.0	0.001±0.001
19	0.545±0.0	0.545±0.0	0.545±0.0	0.53±0.007	0.586±0.0	0.545±0.0	0.304±0.0	0.051±0.0	0.051±0.0	**0.771**±**0.028**
20	0.288±0.008	0.32±0.0	0.423±0.0	0.456±0.031	0.027±0.004	0.308±0.0	0.571±0.0	0.222±0.0	**0.661**±**0.0**	0.307±0.012
21	0.491±0.0	0.491±0.0	0.534±0.0	0.477±0.0	0.587±0.204	0.320±0.12	0.018±0.0	0.007±0.0	0.555±0.0	**0.713**±**0.017**
22	0.369±0.0	0.365±0.0	0.402±0.0	**0.434**±**0.0**	0.352±0.068	0.39±0.0	0.407±0.0	0.365±0.0	0.356±0.0	0.358±0.001
Avg rank	5.409	5.182	5.227	5.591	5.727	5.227	5.545	7.045	6.818	3.227

4.3.2. Performance on datasets with clear concepts

In addition to datasets with fuzzy concepts, datasets with clear concepts are considered in this subsection. Tables 10 and 11 show the performance of algorithms with the ACC and NMI metrics, respectively, for datasets with clear concepts. The performance of Cloud-Cluster on different datasets is relatively stable. These datasets have clear concepts which means that the data distributions are clear. However, Cloud-Cluster solving datasets with clear concepts also introduce the randomness into clustering process, which leads to a fuzzy representation of the distribution range of the data and does not perform optimally on these datasets. Both DPC and DPC-KNN have better clustering performances for datasets with clear concepts, since they are hard partitioned clustering algorithms.

4.3.3. Performance on high-dimensional datasets

In addition to datasets with low-dimensional datasets, high-dimensional datasets are considered in this subsection. Tables 12 and 13 show the performance of algorithms with ACC and NMI metrics for high-dimensional datasets. Cloud-Cluster has average accuracy and average rank at best on these datasets. It can be seen from the results that in all high-dimensional datasets except Perovskite conductivity and Fashion MNIST, Cloud-Cluster achieved higher accuracy. Specifically, the greater the number of features is, the better Cloud-Cluster is than GMM and DPC. The average clustering performance of Cloud-Cluster under the ACC metric was an improvement of 13.9%, 32.7%, 23.8% and 20%, compared to KM, FCM, PFCM, and RKM, respectively. When datasets have relatively high dimensions, DPC, DPC-KNN, and IDPC do a poor job of finding the cluster centers, which leads to poor performance on these datasets.

4.4. Statistical significance

A key concern is that method performance may change across datasets. To compare whether there exist significant differences between different algorithms, we analyze the statistical significance using the Friedman test and Nemenyi post-hoc test [64].

Table 7
Clustering performance of algorithms under FMI metric for datasets with unclear border (SC metric as selection method for c).

No.	KM	FCM	PFCM	RKM	GMM	KFCM	DPC	DPC-KNN	IDPC	Cloud-Cluster
1	0.481±0.0	0.471±0.0	0.467±0.0	0.457±0.0	0.516±0.0	0.479±0.001	0.525±0.0	0.472±0.0	0.482±0.0	**0.526±0.0**
2	**0.756±0.02**	0.640±0.003	0.633±0.0	0.744±0.005	0.318±0.006	0.616±0.0	0.672±0.0	0.541±0.002	0.536±0.0	0.679±0.01
3	0.534±0.001	0.52±0.0	0.516±0.0	0.535±0.0	0.649±0.0	0.524±0.001	0.568±0.0	0.612±0.0	0.623±0.001	**0.735±0.0**
4	0.681±0.007	0.791±0.0	0.722±0.0	0.753±0.007	0.698±0.0	0.66±0.01	**0.813±0.001**	0.737±0.0	0.644±0.0	0.782±0.0
5	0.558±0.0	0.542±0.0	0.526±0.0	0.551±0.0	0.518±0.0	0.545±0.001	0.636±0.001	0.591±0.0	**0.788±0.006**	0.518±0.0
6	0.626±0.0	0.609±0.0	0.592±0.0	0.625±0.0	0.651±0.0	0.606±0.0	0.617±0.0	0.629±0.0	**0.674±0.0**	0.6±0.0
7	0.76±0.0	0.757±0.0	**0.777±0.0**	0.733±0.0	0.71±0.0	0.76±0.0	0.683±0.001	0.712±0.0	0.723±0.0	0.765±0.0
8	0.533±0.0	0.52±0.0	0.515±0.0	0.532±0.0	0.618±0.0	0.525±0.0	0.51±0.0	0.567±0.0	0.578±0.0	**0.686±0.0**
9	0.547±0.0	0.542±0.0	0.541±0.0	0.542±0.0	0.639±0.012	0.542±0.0	0.538±0.0	0.545±0.0	0.534±0.0	**0.937±0.001**
10	0.495±0.0	0.412±0.0	0.461±0.0	0.451±0.001	0.466±0.003	0.411±0.0	0.389±0.0	0.472±0.0	0.375±0.0	**0.514±0.0**
11	0.81±0.0	0.819±0.0	**0.85±0.0**	0.706±0.002	0.71±0.002	0.815±0.0	0.78±0.0	0.78±0.0	0.679±0.002	0.749±0.0
12	0.572±0.0	0.564±0.0	0.54±0.0	0.56±0.0	0.633±0.001	0.576±0.001	0.524±0.0	**0.666±0.002**	0.6±0.0	0.587±0.0
13	0.773±0.002	0.712±0.0	0.474±0.006	0.677±0.007	**0.84±0.03**	0.704±0.0	0.45±0.0	0.70±0.007	0.6614±0.0	0.794±0.002
14	**0.63±0.0**	0.615±0.0	0.577±0.0	0.563±0.0	0.566±0.001	0.568±0.0	0.526±0.0	0.559±0.0	0.526±0.0	0.59±0.0
15	0.511±0.001	0.501±0.0	0.507±0.0	0.498±0.0	0.614±0.002	0.515±0.001	0.496±0.0	0.625±0.001	0.496±0.0	**0.65±0.0**
16	0.79±0.002	0.81±0.0	0.814±0.0	0.81±0.001	0.73±0.007	0.791±0.003	**0.815±0.0**	0.641±0.003	0.684±0.001	0.79±0.001
17	0.536±0.0	0.523±0.0	0.522±0.0	0.547±0.0	0.62±0.0	0.527±0.0	0.532±0.0	0.534±0.0	0.527±0.0	**0.68±0.0**
18	0.648±0.001	0.635±0.0	0.614±0.0	**0.703±0.0**	0.541±0.001	0.585±0.001	0.611±0.0	0.666±0.0	0.566±0.0	0.505±0.0
19	0.783±0.0	0.779±0.0	0.779±0.0	0.776±0.0	0.823±0.0	0.76±0.003	0.6785±0.001	0.597±0.0	0.608±0.003	**0.84±0.0**
20	0.619±0.001	0.572±0.0	0.591±0.0	0.533±0.0	0.659±0.01	0.572±0.0	0.723±0.001	0.619±0.0	**0.784±0.005**	0.623±0.0
21	0.801±0.0	0.791±0.0	0.804±0.0	0.77±0.0	0.816±0.009	0.699±0.005	0.613±0.004	0.593±0.0	0.798±0.001	**0.862±0.0**
22	0.575±0.0	0.573±0.0	0.58±0.0	0.599±0.0	0.568±0.006	0.599±0.0	0.604±0.0	0.637±0.0	0.596±0.0	**0.778±0.001**
Avg rank	4.318	5.727	6.5	6.136	4.818	6.227	6.227	5.364	6.45	3.273

Demšar [64] proposed a series of powerful statistical tests that operate on m datasets by the performance matrices of n algorithms. The key insight is to convert absolute performance on each dataset into algorithm rank, thus removing the effects of varying datasets. We rank the algorithms on each dataset based on their ACC, NMI, ARI, and FMI. This procedure yields a total of 22 rankings for 10 algorithms. In statistical analysis, the p-value (the smallest level of significance) is compared with the prespecified significance level $\alpha = 0.05$ to test the significance of the results. The p-value provides information about whether a statistical hypothesis test is significant or not. The smaller the p-value is, the stronger the evidence against the null hypothesis. The results are shown in Table 14 and Fig. 5.

Table 14 shows the p-value under the Friedman test in different evaluation indices, such as ACC (shown in Table 4), NMI (shown in Table 5), ARI (shown in Table 6) and FMI (shown in Table 7). All p-values are far less than $\alpha = 0.05$, which rejects the null hypothesis of equivalent performance and confirms the existence of significant differences among the performances of all the clustering algorithms. Fig. 5 shows the results of the significance test for a confidence level of $\alpha = 0.05$. Each average rank on different evaluation indices of Cloud-Cluster is approximately 3 and obtains the best ranking although the differences between the top 5 algorithms are not statistically significant. If we relax the confidence to $\alpha = 0.1$ the results are not significantly altered. A similar trend was also found by Piotr et al. [65] in their analysis on Pedestrian Detection. Compared to other baseline algorithms, Cloud-Cluster obtained the best ranking of performance with relative stability. Cloud-Cluster introduces randomness in the clustering process which leads to the range of the clustering distribution becoming fuzzy on these unclear small datasets. From an uncertainty point of view, this means that limited data and limited features weaken the ability of Cloud-Cluster to represent the fuzziness and randomness of data. Furthermore, in most cases, Cloud-Cluster outperformed the other comparison algorithms, thus suggesting that it was able to achieve good clustering results.

Table 8
IID value under SC metric between the clustering results of algorithms and internal information of raw data.

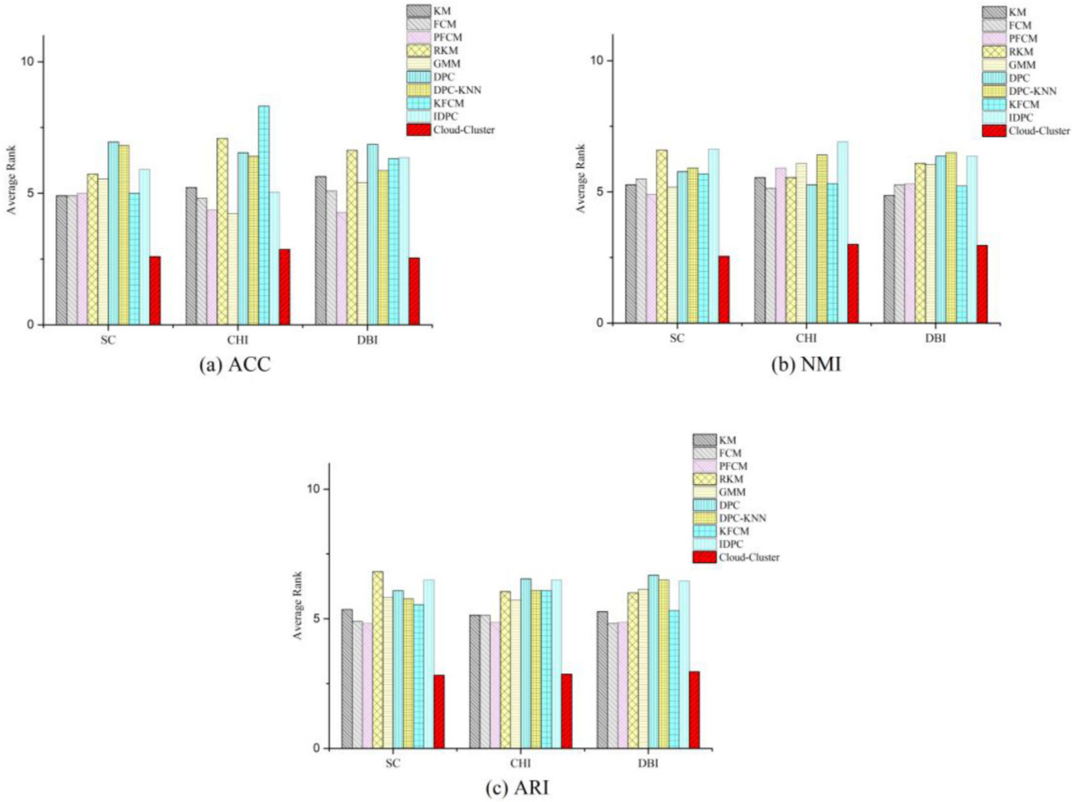
No.	KM	FCM	PFCM	RKM	GMM	KFCM	DPC	DPC-KNN	IDPC	Cloud-Cluster
1	0.54	0.496	0.5	0.458	0.223	0.4853	0.1665	0.36	0.1197	**0.206**
2	0.064	0.043	0.036	0.061	0.131	0.0287	0.1155	0.2105	0.21	**0.008**
3	0.215	0.213	0.207	0.214	0.102	0.2062	0.0993	0.0888	0.1428	**0.044**
4	0.107	0.096	0.066	0.062	0.236	**0.0003**	0.0806	0.0851	0.1908	0.035
5	0.248	0.237	0.223	0.246	0.102	0.2366	0.1195	0.259	**0.0552**	0.209
6	0.266	0.257	0.246	0.248	**0.045**	0.256	0.1745	0.2113	0.201	0.233
7	0.126	0.129	0.125	0.114	**0.016**	0.1175	0.4374	0.4666	0.3111	0.126
8	0.343	0.333	0.324	0.331	**0.011**	0.3302	0.2473	0.1088	0.3394	0.044
9	0.431	0.431	0.43	0.431	**0.008**	0.43	0.4176	0.3858	0.4099	**0.008**
10	0.673	0.673	0.592	0.642	0.534	0.3481	0.1021	0.5089	**0.0187**	0.572
11	0.178	0.178	0.174	0.178	0.109	0.046	0.016	**0.0118**	0.4066	0.183
12	0.592	0.592	0.555	0.589	0.156	0.5917	0.4347	0.6371	0.604	**0.106**
13	0.063	0.016	0.039	**0.027**	0.1	0.0085	0.37	0.1873	0.7724	0.078
14	0.465	0.451	0.421	0.479	0.197	0.4164	**0.3022**	0.0444	0.3022	0.031
15	0.357	0.354	0.351	0.357	0.169	0.3542	0.3547	0.2704	0.3547	**0.162**
16	0.109	0.103	0.091	0.099	0.071	0.0574	**0.0327**	0.0584	0.2124	0.104
17	0.331	0.319	0.313	0.309	**0.007**	0.3166	0.191	0.2045	0.7094	0.033
18	0.683	0.658	0.629	0.723	0.245	0.3874	0.6274	0.4562	0.4524	**0.003**
19	0.04	0.04	0.04	0.04	0.114	0.0408	0.0227	0.0735	0.0735	**0.012**
20	0.341	0.339	0.326	0.301	0.338	0.2349	0.0117	0.1256	**0.0038**	0.279
21	0.183	0.183	0.178	0.182	0.105	0.1729	0.3614	0.3727	0.1714	**0.016**
22	0.457	0.456	0.449	0.446	0.094	0.3658	0.3123	0.4002	0.256	**0.059**

Table 9
IID value under CHI metric between the clustering results of algorithms and internal information of raw data.

No.	KM	FCM	PFCM	RKM	GMM	KFCM	DPC	DPC-KNN	IDPC	Cloud-Cluster
1	5221.4	4934.7	5022.9	4998.5	1542.1	6868.51	1402.22	2300.58	1827.71	**1070.6**
2	133.5	124	117.9	133.1	119.8	55.04	57.69	91.05	91.05	**32.6**
3	271.4	270.1	266.7	271	**25.2**	266.69	77.16	52.63	55.13	28.4
4	65.6	64.1	48	48	66.7	9.2267	44.14	72.91	54.39	**2.7**
5	1131.6	1114.7	1071.6	1126.1	387.4	1114.26	432.03	945.42	**239.48**	988.1
6	171.1	169.7	165.1	165.8	**73.1**	169.34	102.2	111.11	73.67	152.2
7	29.7	28.9	**24.6**	133.6	202.8	131.09	189.39	203.98	204.53	186.3
8	336.1	335.2	335.1	223	218.2	222.92	155.26	**54.97**	171.1	336
9	593.3	592.6	591.7	593.3	2.7	592.55	567.14	474.36	555.73	**0.8**
10	115.3	114.4	101.1	113.5	59.1	86.05	**26.75**	70.19	9.0431	42.3
11	27	27	23.3	27	373.5	72.67	**2.92**	35.35	340.25	15.6
12	330	330	300.4	327.9	**1.3**	330.01	203.13	84.17	296.9	34
13	62.7	51.1	45.9	76.4	16.1	49.183	**3.085**	9.211	67.31	6.3
14	939.8	929.5	853.6	977.6	304	840.51	529.77	166.69	529.77	**79.7**
15	347.1	345.3	340.7	353.5	**72.5**	345.3	338.39	161.96	338.39	101.7
16	40.7	40.8	30	49.1	**1.7**	65.37	32.76	59.42	124.2	38
17	195.8	194.7	192.4	158	8.7	193.85	102.6	110.27	19.44	**6.7**
18	578.2	547	491.9	3626.9	151.5	85.24	489.39	19.92	262.5	**3.3**
19	87.1	87.1	87.1	198.4	137.2	87.05	44.52	110.9	110.9	**0.5**
20	138.8	138.5	131.7	141.9	84.1	90.53	7.388	64.2	**3.1253**	88.8
21	666.6	666.6	644.9	854.3	272	618.68	464.25	474.02	610.19	**79.5**
22	298.8	298.5	292.2	578.1	173.5	331.44	192.45	151.96	143.64	**142.9**

Table 10
Clustering performance of algorithms under ACC metric for datasets with clear border
(SC metric as selection method for *c*).

No.	KM	FCM	PFCM	RKM	GMM	KFCM	DPC	DPC-KNN	IDPC	Cloud-Cluster
1	0.849±0.0	0.727±0.0	0.691±0.073	0.727±0.0	0.712±0.057	0.706±0.06	**0.998±0.0**	0.908±0.0	0.411±0.0	0.727±0.0
2	0.287±0.001	0.249±0.005	0.652±0.006	0.261±0.014	0.41±0.122	0.648±0.0	0.638±0.0	**0.784±0.0**	0.747±0.0	0.455±0.002
3	0.627±0.0	0.627±0.0	0.627±0.0	0.627±0.0	0.67±0.096	0.661±0.06	0.669±0.0	**0.739±0.0**	0.466±0.0	0.627±0.0
4	0.623±0.002	0.612±0.0	0.839±0.008	0.63±0.032	0.613±0.052	0.85±0.0	0.808±0.0	**0.975±0.0**	0.658±0.0	0.785±0.076
5	**0.743±0.0**	**0.743±0.0**	0.633±0.0	0.724±0.034	0.541±0.04	**0.743±0.0**	0.693±0.0	0.68±0.0	0.626±0.0	0.609±0.02
6	0.228±0.015	0.269±0.0	0.271±0.002	0.351±0.005	0.356±0.034	0.355±0.0	0.445±0.0	0.467±0.0	**0.580±0.0**	0.375±0.011
7	0.622±0.006	**0.691±0.0**	0.506±0.007	0.648±0.043	0.676±0.034	0.506±0.005	0.5±0.0	0.524±0.0	0.585±0.0	0.53±0.029
Avg rank	6.28	6.57	6.57	6	6	4.71	4.42	2.85	6	5.57



(a) ACC

(b) NMI

(c) ARI

Fig. 4. Average rank of Cloud-Cluster and other compared methods on ACC, ARI and NMI metrics.

Table 11
Clustering performance of algorithms under NMI metric for datasets with clear border (SC metric as selection method for c).

No.	KM	FCM	PFCM	RKM	GMM	KFCM	DPC	DPC-KNN	IDPC	Cloud-Cluster
1	0.808± 0.001	0.729± 0.0	0.698± 0.105	0.679± 0.004	0.672± 0.053	0.774± 0.03	**0.995± 0.0**	0.953± 0.0	0.326± 0.0	0.749± 0.004
2	**0.501± 0.0**	0.481± 0.0	0.068± 0.005	0.225± 0.02	0.437± 0.054	0.064± 0.0	0.195± 0.0	0.395± 0.0	0.339± 0.0	0.362± 0.007
3	0.588± 0.0	0.588± 0.0	0.578± 0.0	0.437± 0.0	0.639± 0.12	0.727± 0.03	0.742± 0.0	**0.749± 0.0**	0.505± 0.0	0.503± 0.003
4	0.54± 0.0	0.517± 0.0	0.42± 0.016	0.444± 0.009	0.266± 0.133	0.487± 0.0	0.443± 0.0	**0.901± 0.0**	0.051± 0.0	0.317± 0.136
5	**0.546± 0.0**	0.534± 0.0	0.491± 0.0	0.442± 0.035	0.228± 0.102	0.534± 0.0	0.512± 0.0	0.412± 0.0	0.466± 0.0	0.406± 0.065
6	0.201± 0.017	0.24± 0.0	0.001± 0.0	−0.005± 0.0	0.045± 0.037	0.001± 0.0	0.057± 0.0	0.156± 0.0	**0.426± 0.0**	0.007± 0.003
7	0.615± 0.001	**0.691± 0.0**	0.015± 0.001	0.613± 0.095	0.66± 0.023	0.015± 0.001	0.001± 0.0	0.021± 0.0	0.045± 0.0	0.013± 0.014
Avg rank	3	3.28	6.85	7.57	6	4.28	5	3.83	7.42	7.28

4.5. Effect of MBCT-SR-Ex on concepts

This section evaluates the effectiveness of MBCT-SR-Ex, and conducts experiments from two aspects, the performance impact of MBCT-SR and MBCT-SR-Ex on Cloud-Cluster and the concept accuracy of MBCT-SR-Ex.

4.5.1. Performance on datasets with unclear concepts

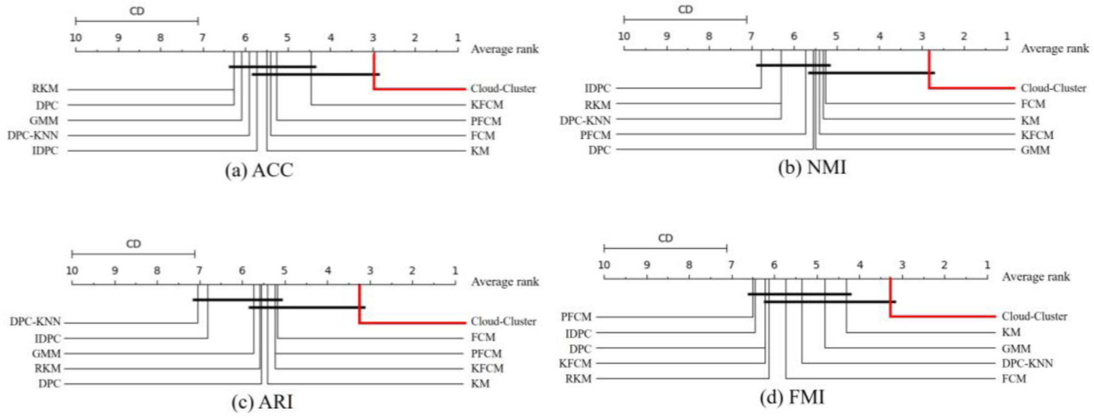
To verify the effectiveness of MBCT-SR-Ex for Cloud-Cluster, the experiments evaluate the performance of Cloud-Cluster using MBCT-SR and MBCT-SR-Ex for the 22 datasets with unclear concepts described in Section 4.1. If the performance is highest, the corresponding entries are bolded, and if the performance is significantly worse than the highest performance on this dataset (lower than 0.1), the corresponding entries are underlined. Table 15 shows that Cloud-Cluster using MBCT-SR-Ex has 16, 13, and 15 bold entries and similar performance to optimal results on other datasets. The average performances are improved by 3.4%, 6.7% and 14.7%. In particular, our proposed concept representation method, MBCT-SR-Ex, has a large performance improvement on the *dermatology* (No. 9) and *prnn_synth* (No. 15) datasets. It employs the concept uncertainty degrees of data points as an important parameter in computing *Ex* in the whole clustering process. In contrast, MBCT-SR only averages the data in the categories as *Ex*, ignoring the relationship between the data points and the concepts. This method treats edge data with overlapping concepts in the same way as central data, resulting in a degradation of algorithm performance.

Table 12
Clustering performance of algorithms under ACC metric for high-dimensional datasets (SC metric as selection method for c).

No.	KM	FCM	PFCM	RKM	GMM	KFCM	DPC	DPC-KNN	IDPC	Cloud-Cluster
1	0.392±0.017	0.316±0.0	0.324±0.0	0.328±0.012	0.31±0.02	**0.525±0.017**	0.339±0.0	0.298±0.0	0.298±0.0	0.4687±0.012
2	0.469±0.0	0.449±0.0	0.487±0.0	0.466±0.0	0.371±0.002	0.438±0.0	0.488±0.0	0.504±0.0	**0.515±0.0**	0.454±0.0
3	0.357±0.0	0.323±0.0	0.411±0.0	0.347±0.0	0.457±0.063	0.345±0.0	0.422±0.0	0.452±0.0	0.452±0.0	**0.774±0.0**
4	**0.564±0.004**	0.113±0.005	0.308±0.028	0.390±0.04	0.486±0.03	0.18±0.0	0.351±0.0	0.316±0.0	0.396±0.0	0.539±0.003
5	0.435±0.001	0.128±0.005	0.246±0.028	0.399±0.05	0.72±0.03	0.187±0.0	0.35±0.0	0.299±0.0	0.310±0.0	**0.723±0.007**
Avg rank	4	9	6.4	5.4	4.8	7.4	4.6	5.6	5	2.8

Table 13
Clustering performance of algorithms under NMI metric for high-dimensional datasets (SC metric as selection method for c).

No.	KM	FCM	PFCM	RKM	GMM	KFCM	DPC	DPC-KNN	IDPC	Cloud-Cluster
1	0.099±0.0	0.086±0.0	0.075±0.0	0.096±0.0	0.069±0.0	0.054±0.0	0.109±0.0	0.071±0.0	0.101±0.0	**0.153±0.001**
2	0.137±0.0	0.168±0.0	0.153±0.0	0.148±0.0	0.041±0.0	0.135±0.0	0.127±0.0	0.117±0.0	0.103±0.0	**0.205±0.0**
3	0.079±0.0	**0.11±0.0**	0.004±0.0	0.094±0.0	0.054±0.002	0.06±0.0	0.079±0.0	0.083±0.0	0.081±0.0	0.031±0.0
4	**0.538±0.002**	0.01±0.005	0.253±0.003	0.357±0.059	**0.538±0.0**	0.088±0.0	0.385±0.0	0.355±0.0	0.373±0.0	0.461±0.0
5	0.31±0.002	0.012±0.002	0.146±0.024	0.298±0.01	**0.687±0.02**	0.05±0.0	0.385±0.0	0.357±0.0	0.216±0.0	0.337±0.01
Avg rank	**4**	5.6	7.4	4.8	4	8.8	4	6.4	6.2	3.8



(a) ACC (b) NMI (c) ARI (d) FMI

Fig. 5. Visualization of the Nemenyi post-hoc test under different evaluation indices with unclear datasets.

Table 14
The p-values in different evaluation indices of Tables 4–7 under the Friedman test.

	ACC	NMI	ARI	FMI
p-value	0.0147	0.0018	0.0075	0.0046

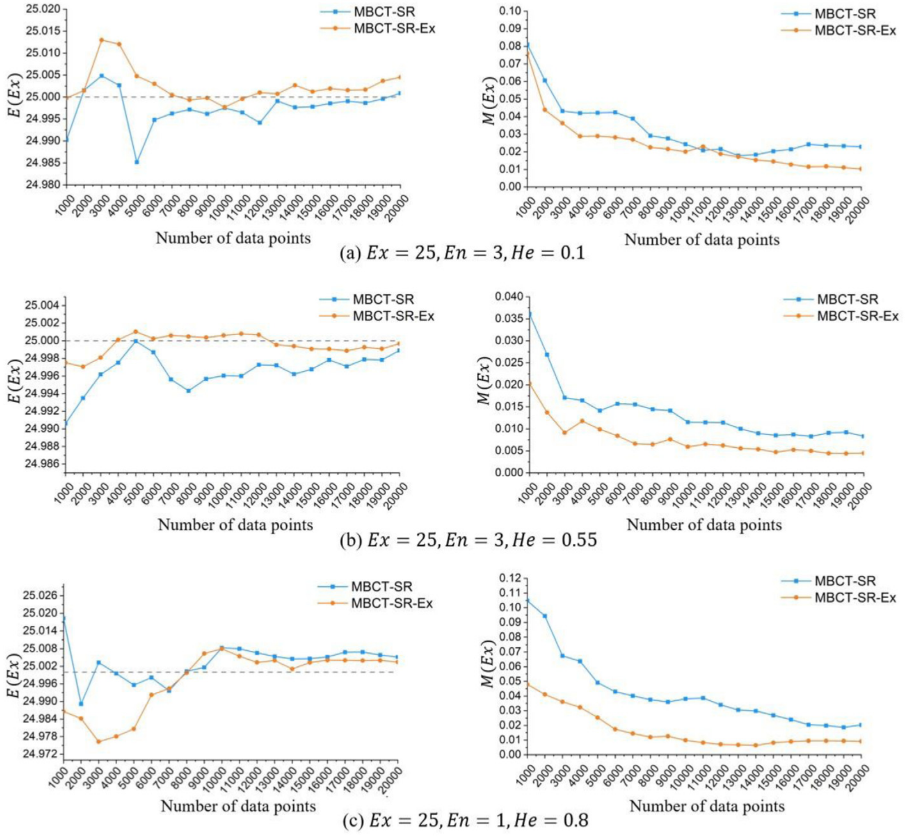
4.5.2. *Effect of data sizes on MBCT-SR-Ex*

In addition to the impact on the performance of the algorithm, we consider the impact of the data sizes for MBCT-SR-Ex on the calculation of concept accuracy. Learning concepts is a dynamic process that goes from unclear to clear with increasing information or knowledge. Aligning our approach with the setup of MBCT-SR, in our experiments the initial concepts are set to triplets: $C_1(25, 3, 0.1)$, $C_2(25, 3, 0.55)$, and $C_3(25, 1, 0.8)$. The forward cloud transformation algorithm (FCT) uses the above three situations, so as to generate data points to compare the mean $E(Ex)$ and MSE $M(Ex)$ of $\widehat{Ex}$ with increasing data. The size n of data points is $[1000, 20000]$ with a step of 1000.

From Fig. 6, the mean $E(Ex)$ is very close to 25 in MBCT-SR and MBCT-SR-Ex. Moreover, the MSE $M(Ex)$ tends to 0 faster in MBCT-SR-Ex compared to MBCT-SR. Particularly, when $He = 0.55$ and the amount of data is 4000, the Ex value calculated by MBCT-SR-Ex starts to approach 25, while the gap between the Ex of the MBCT-SR approach and standard expectations is still large. In addition, the case of a small amount of data is also considered, as shown in Fig. 7, which compares these two BCTs with the size of data points changing from $[50, 1000]$ with a step of 50. From Fig. 7, the MSE $M(Ex)$ tends to zero faster with increasing

Table 15
Clustering performance of Cloud-Cluster using MBCT-SR and MBCT-SR-Ex under ACC, NMI, ARI metrics for datasets with clear border.

No.	ACC		NMI		ARI	
	Cloud-Cluster (MBCT-SR)	Cloud-Cluster (Ours)	Cloud-Cluster (MBCT-SR)	Cloud-Cluster (Ours)	Cloud-Cluster (MBCT-SR)	Cloud-Cluster (Ours)
1	0.465 ± 0.075	**0.521 ± 0.032**	0.103 ± 0.078	**0.164 ± 0.034**	0.111 ± 0.084	**0.178 ± 0.037**
2	0.672 ± 0.005	**0.694 ± 0.076**	0.447 ± 0.016	**0.452 ± 0.109**	0.438 ± 0.013	**0.472 ± 0.114**
3	**0.847 ± 0.005**	0.841 ± 0.006	**0.387 ± 0.013**	0.365 ± 0.015	**0.479 ± 0.014**	0.464 ± 0.016
4	**0.874 ± 0.016**	0.866 ± 0.01	**0.49 ± 0.046**	0.478 ± 0.035	**0.558 ± 0.053**	0.532 ± 0.028
5	0.559 ± 0.003	**0.566 ± 0.003**	0.011 ± 0.001	**0.014 ± 0.001**	0.013 ± 0.001	**0.017 ± 0.002**
6	0.705 ± 0.06	**0.715 ± 0.013**	**0.145 ± 0.048**	0.143 ± 0.018	0.179 ± 0.058	**0.182 ± 0.022**
7	0.859 ± 0.005	**0.866 ± 0.004**	0.398 ± 0.013	**0.418 ± 0.012**	0.511 ± 0.015	**0.533 ± 0.013**
8	**0.821 ± 0.009**	0.806 ± 0.011	**0.33 ± 0.022**	0.29 ± 0.024	**0.41 ± 0.024**	0.373 ± 0.028
9	0.728 ± 0.156	**0.928 ± 0.116**	0.36 ± 0.216	**0.742 ± 0.177**	0.288 ± 0.299	**0.782 ± 0.247**
10	**0.446 ± 0.022**	0.435 ± 0.01	0.256 ± 0.06	**0.268 ± 0.037**	0.183 ± 0.042	**0.206 ± 0.02**
11	**0.667 ± 0.0**	**0.667 ± 0.0**	**0.734 ± 0.0**	0.71 ± 0.024	**0.568 ± 0**	0.561 ± 0.008
12	0.599 ± 0.016	**0.663 ± 0.024**	0.029 ± 0.01	**0.08 ± 0.024**	0.035 ± 0.013	**0.104 ± 0.032**
13	0.8 ± 0.011	**0.822 ± 0.018**	0.394 ± 0.047	**0.465 ± 0.06**	0.381 ± 0.046	**0.501 ± 0.071**
14	**0.697 ± 0.01**	0.69 ± 0.007	0.105 ± 0.013	**0.108 ± 0.009**	**0.153 ± 0.015**	0.143 ± 0.011
15	0.623 ± 0.082	**0.771 ± 0.01**	0.091 ± 0.096	**0.279 ± 0.019**	0.084 ± 0.087	**0.291 ± 0.021**
16	0.659 ± 0.003	**0.667 ± 0.0**	**0.564 ± 0.016**	0.539 ± 0.006	**0.487 ± 0.01**	0.462 ± 0.005
17	**0.823 ± 0.01**	0.811 ± 0.014	**0.326 ± 0.024**	0.298 ± 0.028	**0.415 ± 0.025**	0.385 ± 0.034
18	0.527 ± 0.011	**0.529 ± 0.012**	**0.003 ± 0.002**	**0.003 ± 0.002**	0.002 ± 0.002	**0.003 ± 0.002**
19	0.894 ± 0.015	**0.916 ± 0.014**	0.556 ± 0.045	**0.572 ± 0.052**	0.613 ± 0.049	**0.687 ± 0.049**
20	0.655 ± 0.009	**0.668 ± 0.027**	**0.32 ± 0.019**	0.314 ± 0.048	0.286 ± 0.024	**0.326 ± 0.052**
21	0.897 ± 0.011	**0.909 ± 0.01**	0.516 ± 0.035	**0.546 ± 0.031**	0.629 ± 0.034	**0.668 ± 0.031**
22	**0.601 ± 0.001**	**0.601 ± 0.001**	**0.477 ± 0.008**	0.476 ± 0.009	**0.373 ± 0.007**	**0.373 ± 0.006**
Avg	0.701	**0.725**	0.32	**0.351**	0.327	**0.375**



(a) $Ex = 25, En = 3, He = 0.1$

(b) $Ex = 25, En = 3, He = 0.55$

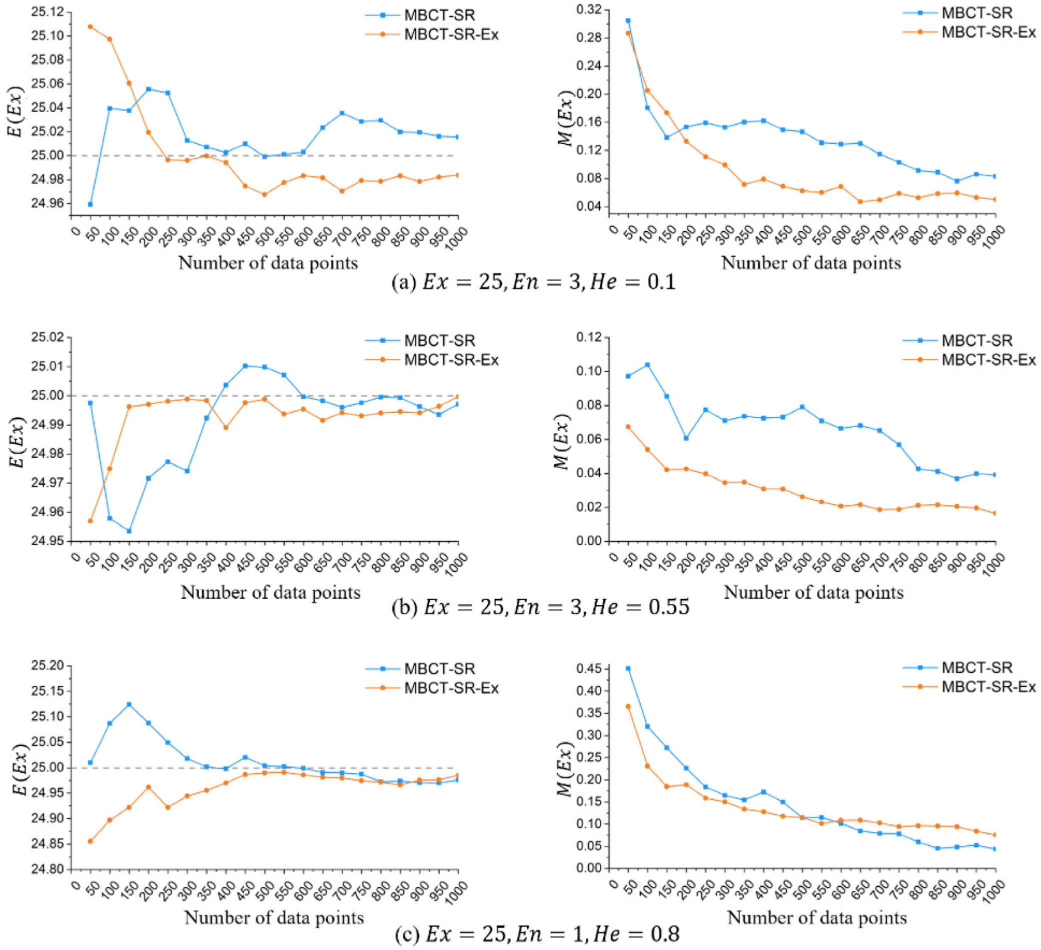
(c) $Ex = 25, En = 1, He = 0.8$

Fig. 6. The mean and MSE of *Ex* under a small amount of data generated by different concepts.

sample sizes in MBCT-SR-Ex. With $He = 0.1$ or 0.55, the *Ex* calculated by MBCT-SR-Ex gradually converges to 25 when the data number increases and its fluctuations are small from the MSE values. The method is able to reach the approximate expectation value more quickly with smaller data sizes. To obtain a more accurate representation of the concept, finding a suitable BCT can significantly reduce the computation error of the parameters. MBCT-SR-Ex improves the *Ex* computation by considering the uncertainty between each data point and the concept, obtaining a more accurate representation of the concept.

4.6. Evaluation of concepts

In this section, we research the iterative process of Cloud-Cluster to explain how the different concepts change in this



(a) $Ex = 25, En = 3, He = 0.1$

(b) $Ex = 25, En = 3, He = 0.55$

(c) $Ex = 25, En = 1, He = 0.8$

Fig. 7. The mean and MSE of *Ex* under a small amount of data generated by different concepts.

method. To verify the role of concept uncertainty in clustering, we employ concept evaluation to measure the concepts of each dataset. The concept describes a larger range of category representations, and there are both differences and similarities in categories under uncertain data information.

The results are shown in Fig. 8. The top broken line is the fluctuation of Concept Cluster Drift Degree (C^2D^2) 10 times for each dataset while the bottom is the KL-divergence. It can be seen that the drift degree of such dataset fluctuates wildly in Fig. 8(d)(e)(f), which reflects the fact that these datasets have strong unclear concepts when randomness is embedded. However, the KL-divergence cannot reflect such fluctuation. Fig. 8(a)(b)(c) shows a small fluctuation of the drift degree in the *iris*, *Flame*, and *piral* datasets, which have clear concepts. These datasets have low similarity between classes and are more conducive to concept learning and faster convergence conditions. When C^2D^2 decreases, the constructed concept is more precise.

To describe each concept, we choose the backbone element as the main feature range, giving the representation range of each feature. Fig. 9 shows the backbone range of features of each cluster in the *iris* dataset, in which the range is computed by the parameters (Ex, En, He) of each concept from $Ex - 0.67En$ to $Ex + 0.67En$. Data scores can be expressed in a human-understandable way, such as *low* or *high*. The transformation between feature scores and understandable concepts is shown in Table 16. The expression can help humans classify data in a simple way, which is shown in Table 17.

The experiment also transforms the *Wine* dataset into more concise data, as shown in Fig. 10. The features *Malic acid*, *Hue*, and *Proline* exist in the overlapping case. Meanwhile, the features *Alkalinity of ash* and *Magnesium* have fuzzy concepts, where all of the concepts overlap substantially. Thus, we find the intersection points of overlapping conceptual boundaries as boundary values to make a reasonable expression. Finally, the concepts of each class in the *wine* dataset are shown in Table 18.

Table 16
The transformation between the feature and concepts in *iris* dataset.

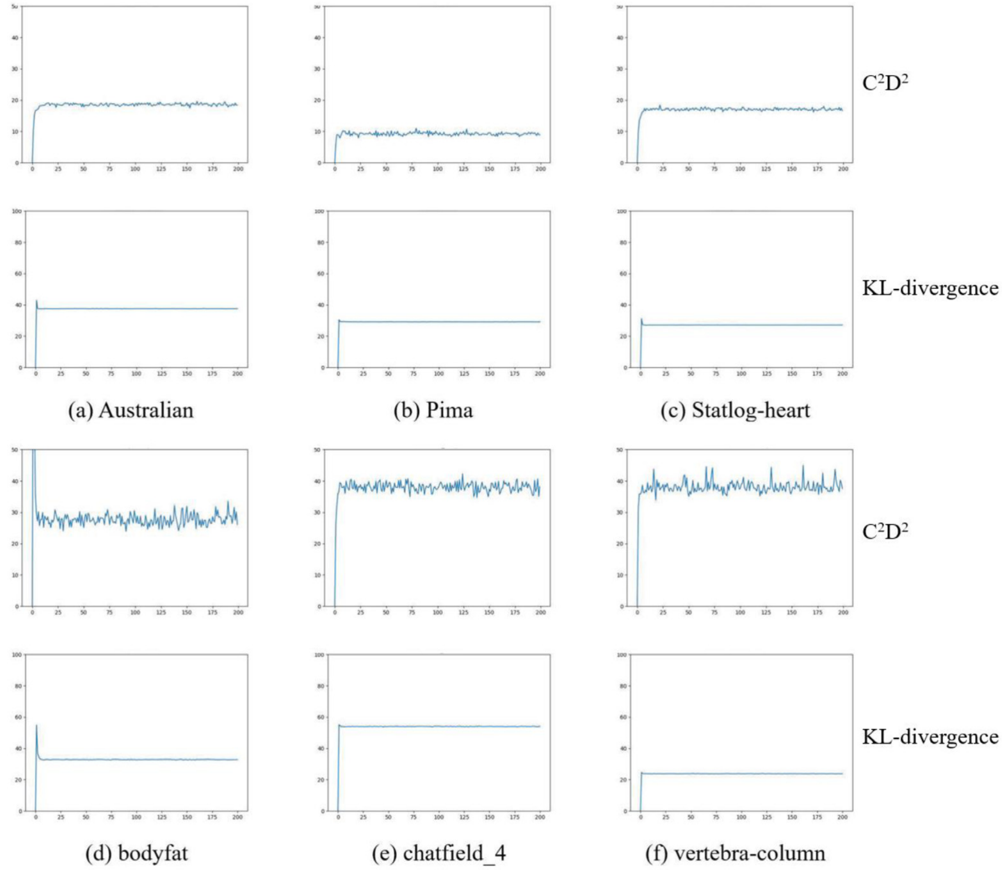
Features	Scope	Concept	Scope	Concept
Sepal length	4.437 ~ 5.575	Short	5.703 ~ 6.821	Long
Sepal width	2.573 ~ 3.155	Short	3.155 ~ 3.698	Long
Petal length	0.274 ~ 2.654	Short	3.712 ~ 6.100	Long
Petal width	0 ~ 0.755	Short	1.167 ~ 2.185	Long

Table 17
Classification in a simple way in iris dataset.

Sepal length	Sepal width	Petal length	Petal width
Short	Long	Short	Short
Long	Short	Long	Long

5. Conclusion

This paper proposes a novel clustering method named Cloud-Cluster, which combines the random uncertainty of data to reserve uncertain information to cluster data. To incorporate random and fuzzy uncertainty into the clustering method, Concept Based Refinement Aggregation is proposed in Cloud-Cluster to



(a) Australian (b) Pima (c) Statlog-heart

(d) bodyfat (e) chatfield_4 (f) vertebra-column

Fig. 8. Fluctuation of Concept Cluster Drift Degree (C^2D^2) and maximum likelihood estimate.

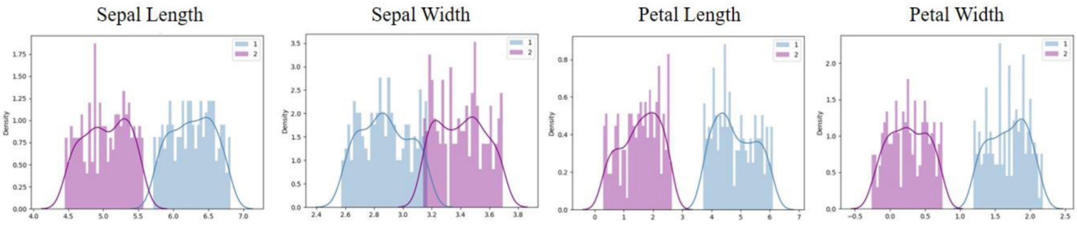


Fig. 9. The backbone range of each feature of clusters in *iris* dataset.

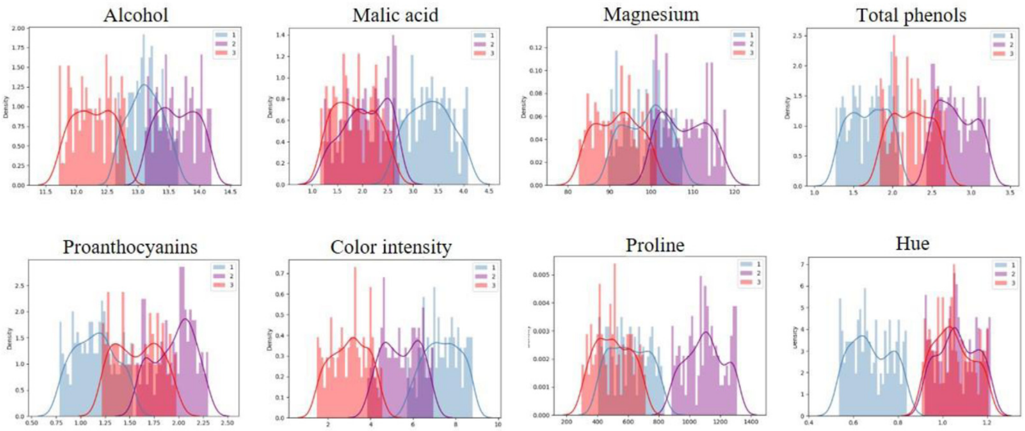


Fig. 10. The backbone range of each feature of clusters in *wine* dataset.

Table 18
The feature scores and concepts in *wine* dataset.

Features	Class1	Class2	Class3
Alcohol	Medium	High	Low
Malic acid	High	Low	Low
Ash	High	High	Low
Alkalinity of ash	High	Low	High
Magnesium	Medium	High	Low
Total phenols	Low	High	Medium
Flavanoids	Low	High	Medium
Nonflavanoid phenols	High	Low	Medium
Proanthocyanins	Low	High	Medium
Color intensity	High	Medium	Low
Hue	Low	High	High
OD280_OD315 of diluted wines	Low	High	High
Proline	Low	High	Low

find a better data partition by adding randomness to extend the range of data distributions. Then, the uncertainty degree is calculated, and the data are transformed into concepts to construct more stable concepts by an improved multistep cloud transformation algorithm named *MBCT-SR-Ex*. To evaluate the uncertainty of concepts in the iterative process, Concept Uncertainty Evaluation is proposed in Cloud-Cluster, which defines Cluster Concept Drift Degree (C^2D^2) to measure the fluctuation of concepts. The stability of all concepts is proven by the convergence proof of C^2D^2 in the clustering process, which implies the convergence of the algorithm. Experiments show that Cloud-Cluster has good performance and improves the average clustering accuracy by 14.4%, 14.4%, 16%, 15.6% and 17.3% compared to K-Means and uncertainty theory-based clustering algorithms such as FCM, PFCM, RKM and GMM, respectively. The proposed MBCT-SR-Ex effectively improves the algorithm performance. Moreover, the results of uncertainty evaluation show that datasets have different levels of uncertainty in the clustering process.

In the future, we will consider employing Cloud-Cluster in deep learning. For example, we will attempt to utilize Cloud-Cluster to discover the concepts hidden in the latent space of deep learning models such as VAE. We hope to research the interpretation of unsupervised learning for latent variables in VAE, with Cloud-Cluster to realize concept construction.

CRediT authorship contribution statement

Yue Liu: Conceptualization, Methodology, Validation, Formal analysis, Writing – original draft, Writing – review & editing, Supervision. **Zitu Liu:** Methodology, Software, Validation, Data Curation, Writing – original draft, Writing – review & editing. **Shuang Li:** Software, Validation, Writing – original draft, Writing – review & editing. **Yike Guo:** Methodology, Formal analysis, Writing – original draft. **Qun Liu:** Formal analysis, Writing – review & editing. **Guoyin Wang:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 52073169), the State Key Program of National Nature Science Foundation of China (Grant No. 61936001), and Training Program of the Major Research Plan of the National Natural Science Foundation of China (No. 92270124).

References

[1] X. Liu, M. Milo, N.D. Lawrence, et al., A tractable probabilistic model for affymetrix probe-level analysis across multiple chips, Bioinformatics 21 (18) (2005) 3637–3644.

[2] A. Deshpande, C. Guestrin, S.R. Madden, J.M. Hellerstein, W. Hong, Model-based approximate querying in sensor networks, VLDB J. 14 (4) (2005) 417–443.

[3] X. Du, H. Xu, F. Zhu, A data mining method for structure design with uncertainty in design variables, Comput. Struct. 244 (2021) 106457.

[4] Z. Fan, Z. Liu, S. Wang, L. Zheng, P.S. Yu, Modeling sequences as distributions with uncertainty for sequential recommendation, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021, pp. 3019–3023.

[5] S. Bobek, G.J. Nalepa, Introducing uncertainty into explainable ai methods, in: International Conference on Computational Science, Springer, 2021, pp. 444–457.

[6] R. Alizadehsani, M. Roshanzamir, S. Hussain, A. Khosravi, A. Koohestani, M.H. Zangooei, M. Abdar, A. Beykikhoshk, A. Shoeibi, A. Zare, et al., Handling of uncertainty in medical data using machine learning and probability theory techniques: A review of 30 years (1991–2020), Ann. Oper. Res. (2021) 1–42.

[7] R. Achanta, S. Susstrunk, Superpixels and polygons using simple non-iterative clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4651–4660.

[8] A. Saxena, M. Prasad, A. Gupta, N. Bharill, O.P. Patel, A. Tiwari, M.J. Er, W. Ding, C.-T. Lin, A review of clustering techniques and developments, Neurocomputing 267 (2017) 664–681.

[9] R.T. Ng, J. Han, Clarans: A method for clustering objects for spatial data mining, IEEE Trans. Knowl. Data Eng. 14 (5) (2002) 1003–1016.

[10] X. Peng, R. Li, J. Wang, H. Shang, User-guided clustering for video segmentation on coarse-grained feature extraction, IEEE Access 7 (2019) 149820–149832.

[11] Q. Wang, C. Wang, Z. Feng, et al., Review of k-means clustering algorithm, Electron. Des. Eng. 20 (7) (2012) 21–24.

[12] S. Guha, R. Rastogi, K. Shim, An eficient clustering algorithm for large databases, in: Proc. of ACM-SIGMOD Int. Conf. on Management of Data, 1998, 27(2), pp. 73–84.

[13] T. Zhang, R. Ramakrishnan, M. Livny, Birch: A new data clustering algorithm and its applications, Data Min. Knowl. Discov. 1 (2) (1997) 141–182.

[14] M. Ester, H.P. Kriegel, J. Sander, X. Xu, et al., A density-based algorithm for discovering clusters in large spatial databases with noise, in: Kdd, Vol. 96, 1996, pp. 226–231.

[15] W. Wang, J. Yang, R. Muntz, et al., Sting: A statistical information grid approach to spatial data mining, in: VLDB, Vol. 97, 1997, pp. 186–195.

[16] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (6191) (2014) 1492–1496.

[17] A. Lotfi, S.A. Seyedi, P. Moradi, An improved density peaks method for data clustering, in: 2016 6th International Conference on Computer and Knowledge Engineering, ICCKE, IEEE, 2016, pp. 263–268.

[18] H. Shen, J. Yang, S. Wang, X. Liu, Attribute weighted mercer kernel based fuzzy clustering algorithm for general non-spherical datasets, Soft Comput. 10 (11) (2006) 1061–1073.

[19] M. Du, S. Ding, H. Jia, Study on density peaks clustering based on k-nearest neighbors and principal component analysis, Knowl.-Based Syst. 99 (2016) 135–145.

[20] H.J. Zimmermann, Fuzzy Set Theory, Wiley Interdisciplinary Reviews: Computational Statistics, Vol. 2, No. 3, 2010, pp. 317–332.

[21] Z. Pawlak, Rough sets, Int. J. Comput. Inform. Sci. 11 (5) (1982) 341–356.

[22] N. Singh, A. Agrawal, R. Khan, Gaussian mixture model: a modeling technique for speaker recognition and its component, Int. J. Comput. Appl. 975 (2014) 8887.

[23] D. Li, Y. Du, Artificial Intelligence with Uncertainty, CRC Press, 2007.

[24] B. Kosko, Fuzziness vs. probability, Int. J. Gen. Syst. 17 (2–3) (1990) 211–240.

[25] D. Li, Membership clouds and membership cloud generators, Comput. Res. Dev. 32 (6) (1995) 15–20.

[26] D.y. Li, C.-Y. Liu, L. Liu, et al., Study on the universality of the normal cloud model, Eng. Sci. 6 (8) (2004) 28–34.

[27] H.-Y. Ma, G.-Y. Wang, Q.H. Zhang, N. Xu, Multi-granularity color image segmentation based on cloud model, Jisuanji Gongcheng/ Comput. Eng. 38 (20).

[28] W. Xie, D. Liu, M. Yang, S. Chen, B. Wang, Z. Wang, Y. Xia, Y. Liu, Y. Wang, C. Zhang, Segcloud: a novel cloud image segmentation model 600 using a deep convolutional neural network for ground-based all-sky-view camera observation, Atmos. Meas. Tech. 13 (4) (2020) 1953–1961.

[29] W. Li, J. Zhao, B. Xiao, Multimodal medical image fusion by cloud model theory, Signal, Image Video Process. 12 (3) (2018) 437–444.

[30] Y. Lin, K. Zhou, J. Li, Application of cloud model in rock burst prediction and performance comparison with three machine learning algorithms, IEEE Access 6 (2018) 30958–30968.

[31] S. Lou, Y. Feng, Z. Li, H. Zheng, J. Tan, An integrated decision-making method for product design scheme evaluation based on cloud model and eeg data, Adv. Eng. Inform. 43 (2020) 101028.

[32] K.p. Zhou, L. Yun, H. w. Deng, J. l. Li, C. j. Liu, Prediction of rock burst classification using cloud model with entropy weight, Trans. Nonferr. Met. Soc. China 26 (7) (2016) 1995–2002.

[33] P. Wang, X. Xu, S. Huang, C. Cai, A linguistic large group decision making method based on the cloud model, IEEE Trans. Fuzzy Syst. 26 (6) (2018) 3314–3326.

[34] S. Xie, S. Dong, Y. Chen, Y. Peng, X. Li, A novel risk evaluation method for fire and explosion accidents in oil depots using bow-tie analysis and risk matrix analysis method based on cloud model theory, Reliab. Eng. Syst. Saf. 215 (2021) 107791.

[35] Y. Xiong, D. Kong, Z. Cheng, G. Wu, Q. Zhang, The comprehensive identification of roof risk in a fully mechanized working face using the cloud model, Mathematics 9 (17) (2021) 2072.

[36] Y. Wu, H. Chu, C. Xu, Risk assessment of wind-photovoltaic-hydrogen storage projects using an improved fuzzy synthetic evaluation approach based on cloud model: A case study in China, J. Energy Storage 38 (2021) 102580.

[37] C. Xu, G. Wang, Excursive measurement and analysis of normal cloud concept, Comput. Sci. 41 (2014) 9–14.

[38] J.C. Bezdek, R. Ehrlich, W. Full, FCM: The fuzzy c-means clustering algorithm, Comput. Geosci. 10 (2–3) (1984) 191–203.

[39] E. Hanyu, Y. Cui, W. Pedrycz, et al., Design of fuzzy rule-based models with fuzzy relational factorization, Expert Syst. Appl. 206 (2022) 117904.

[40] T.M. Tuan, M.D. Sinh, T.Đ. Khang, et al., A new approach for semi-supervised fuzzy clustering with multiple fuzzifiers, Int. J. Fuzzy Syst. (2022) 1–14.

[41] L. Jiao, H. Yang, Z. Liu, et al., Interpretable fuzzy clustering using unsupervised fuzzy decision trees, Inform. Sci. 611, 540–563.

[42] S. Tongbram, B.A. Shimray, L.S. Singh, N. Dhanachandra, A novel image segmentation approach using fcm and whale optimization algorithm, J. Ambient Intell. Humaniz. Comput. (2021) 1–15.

[43] Z. Zhang, Z. Cao, Y. Li, Research based on euclid distance with weights of k means algorithm, J. Zhengzhou Univ. (Eng. Sci.) 31 (2010) 89–92.

[44] H. Xu, W. Zeng, X. Zeng, et al., An evolutionary algorithm based on Minkowski distance for many-objective optimization, IEEE Trans. Cybern. 49 (11) (2018) 3968–3979.

[45] M. Kordos, M. Blachnik, R. Scherer, Fuzzy clustering decomposition of genetic algorithm-based instance selection for regression problems, Inform. Sci. 587 (2022) 23–40.

[46] B. Chakraborty, K. Chakma, A. Mukherjee, A density-based clustering algorithm and experiments on student dataset with noises using rough set theory, in: 2016 IEEE International Conference on Engineering and Technology, ICETECH, IEEE, 2016, pp. 431–436.

[47] S. Askari, N. Montazerin, M.F. Zarandi, Generalized possibilistic fuzzy c-means with novel cluster validity indices for clustering noisy data, Appl. Soft Comput. 53 (2017) 262–283.

[48] X. Wu, H. Zhou, B. Wu, T. Zhang, A possibilistic fuzzy gath-geva clustering algorithm using the exponential distance, Expert Syst. Appl. 184 (2021) 115550.

[49] Y. Meng, J. Liang, F. Cao, Y. He, A new distance with derivative information for functional k-means clustering algorithm, Inform. Sci. 463 (2018) 166–185.

[50] L. Jing, D. Deng, J. Yu, Weighting exponent selection of fuzzy c-meansvia Jacobian matrix, in: International Conference on Knowledge Science, Engineering and Management, Springer, 2014, pp. 115–126.

[51] M. Ren, Z. Wang, J. Jiang, A self-adaptive fcm for the optimal fuzzy weighting exponent, Int. J. Comput. Intell. Appl. 18 (02) (2019) 1950008.

[52] J.F. Peters, Fuzzy sets, near sets, and rough sets for your computational intelligence toolbox, in: Foundations of Computational Intelligence, Vol. 2, Springer, Berlin, Heidelberg, 2009, pp. 3–25.

[53] B. Tripathy, A. Ghosh, Data clustering algorithms using rough sets, in: Handbook of Research on Computational Intelligence for Engineering, Science, and Business, IGI Global, 2013, pp. 297–327.

[54] P. Wu, C. Liu, Financial distress study based on pso k-means clustering algorithm and rough set theory, in: Applied Mechanics and Materials, Vol. 675, Trans Tech Publ, 2013, p. 411, 2377–2383.

[55] B. Chakraborty, K. Chakma, A. Mukherjee, A density-based clustering algorithm and experiments on student dataset with noises using rough set theory, in: 2016 IEEE International Conference on Engineering and Technology, ICETECH, IEEE, 2016, pp. 431–436.

[56] J. Zhou, Z. Lai, D. Miao, C. Gao, X. Yue, Multigranulation rough-fuzzy clustering based on shadowed sets, Inform. Sci. 507 (2020) 553–573.

[57] Y. Liu, Adaptive concept abstraction method on mullti-granularity-gaussian cloud transformation, in: Proc. Computer Engineering and Applications, Vol. 51, 2015, pp. 1–8.

[58] Y. Deng, S. Liu, W. Zhang, L. Wang, J. Wang, General multidimensional cloud model and its application on spatial clustering in Zhanjiang, Guangdong, J. Geogr. Sci. 20 (5) (2010) 787–798.

[59] G. Wang, C. Xu, Q. Zhang, X. Wang, A multi-step backward cloud generator algorithm, in: International Conference on Rough Sets and Current Trends in Computing, Springer, 2012, pp. 313–322.

[60] C. y. Liu, M. Feng, X. j. Dai, D. y. LI, A new algorithm of backward cloud, in: Acta Simulata Systematica Sinica, Vol. 11.

[61] H. Chen, B. Li, C. Liu, et al., An algorithm of backward cloud without certainty degree, J. Chinese Comput. Syst. 36 (3) (2015) 544–549.

[62] H. Wang, S. Qin, S. Liu, L. Yu, K. Wang, An improved algorithm of backward cloud based on curve fitting, CAAI Trans. Intell. Syst. 9 (5) (2014) 590–594.

[63] J.A. Hartigan, M.A. Wong, Algorithm AS 136: A k-means clustering algorithm, J. R. Stat. Soc. Ser. C (Appl. Stat.) 28 (1) (1979) 100–108.

[64] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.

[65] P. Dollar, C. Wojek, B. Schiele, et al., Pedestrian detection: An evaluation of the state of the art, IEEE Trans. Pattern Anal. Mach. Intell. 34 (4) (2011) 743–761.